SDE | 2010

XXVII Seminar in Differential Equations
Bořetice, May 31 - June 4, 2010

Vít Dolejší

Modern Numerical Methods for Solving Partial Differential Equations

CONFERENCE VENUE:

Hotel Kraví hora
Bořetice, Czech Republic
48º55´9.955"N, 16º50´46.676"E

FACULTY
OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

sde.kma.zcu.cz/2010

Proceedings of Seminar in Differential Equations[1]

Volume I

Discontinuous Galerkin method for PDE with applications in fluid dynamics[2]

Vít Dolejší
Charles University Prague,
Faculty of Mathematics and Physics

Bořetice
May 31 – June 4, 2010

[1]Seminar was supported by MŠMT ČR – ME09109.
[2]This work is a part of the research project MSM 0021620839 financed by the Ministry of Education of the Czech Republic.

University of West Bohemia

ISBN 978-80-261-0168-0

Chapter 1

Introduction

Our goal is to develop a sufficiently robust, efficient and accurate numerical scheme for the solution of the system of the compressible Navier-Stokes equations which describe a motion of viscous compressible fluids. This system can by written in the conservative form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \left(\sum_{k=1}^{d} \mathbb{I}\!\boldsymbol{K}_{s,k}(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_k} \right) \qquad \text{in } (0,T) \times \Omega, \tag{1.1}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ is a bounded domain occupied by a fluid, $T > 0$ is time to be reached, $\boldsymbol{w} : (0,T) \times \Omega \to \mathbb{R}^{d+2}$ is the state vector having components density, momentum and total energy, functions $\boldsymbol{f}_s : \mathbb{R}^{d+2} \to \mathbb{R}^{d+2}$, $s = 1, \ldots, d$ represent the inviscid fluxes and $\mathbb{I}\!\boldsymbol{K}_{s,k} : \mathbb{R}^{d+2} \to \mathbb{R}^{(d+2)\times(d+2)}$, $s, k = 1, \ldots, d$ represent the viscous terms. For more details see, e.g., [FFS03], [Wes01].

The class of *discontinuous Galerkin* (DG) methods seems to be one of the most promising candidates to construct high order accurate schemes for solving partial differential equations (1.1). In last years the DG methods were employed in many papers for the discretization of compressible fluid flow problems, see, e.g., [BCRS05], [BR97], [BR00], [BO99a], [Dol04], [Dol08], [FDK07], [FK07], [DM06], [Har06], [HH02], [HH06a], [HH06b], [KvdVdV06a], [KvdVdV06b], [LBK98], [vdVvdV02a], [vdVvdV02b] and the references cited therein.

DG technique is based on a piecewise polynomial but discontinuous approximation which provides robust and high-order accurate approximations, particularly in transport dominated regimes. Moreover, DG method can easily employ non-matching and non-uniform grids, and different polynomial approximation degrees on different elements. This allows a simple treatment with *hp*-adaptation techniques. Finally, in combination with block-type preconditioners, DG methods can be easily parallelized. For a survey about DG methods, see [Coc99] or [CKS00].

There exist several DG techniques for discretizing of linear elliptic boundary value problems (see [ABCM02]). We deal with the so-called *interior penalty Galerkin* methods, namely symmetric interior penalty Galerkin (SIPG, introduced by [Arn82]), nonsymmetric interior penalty Galerkin (NIPG, introduced by [RWG99]) and incomplete

interior penalty Galerkin (IIPG, introduced by [DSW04]) methods. IPG techniques are very popular thanks their favourable analytical properties.

The aim of these lecture notes is to give some theoretical justifications of the IPG methods applied to the Navier-Stokes equations. We concentrate on the fact that DG methods have a high order of accuracy provided that exact solution is sufficiently smooth. In virtue of an absence of theoretical results concerning an existence of the solution of (1.1), a numerical analysis of DGM applied to the Navier-Stokes equations exhibits rather difficult task. Therefore, we deal with the following simplified model problem: Find function $u : \Omega \times (0,T) \to I\!R$ such that

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \left(\sum_{k=1}^{d} D_{s,k}(u) \frac{\partial u}{\partial x_k} \right) \qquad \text{in } (0,T) \times \Omega, \tag{1.2}$$

where $f_s : I\!R^{d+2} \to I\!R,\ s = 1, \ldots, d$ represent convective fluxes and $D_{s,k} : I\!R \to I\!R,\ s,k = 1, \ldots, d$ represent (generally anisotropic) viscosity. In order to keep this notes more readable, we put

$$D_{s,k}(u) = \varepsilon \delta_{s,k}, \tag{1.3}$$

where $\delta_{s,k}$ is the Kronecker delta (i.e, $\delta_{s,k} = 1$ for $s = k$, $\delta_{s,k} = 0$ otherwise). Then the equation (1.2) reduces to

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = \Delta u \qquad \text{in } (0,T) \times \Omega, \tag{1.4}$$

where Δ is the Laplace operator.

The contents of the rest of these notes are the following. In Chapter 2 we apply IPG technique to the solution of the Poisson equation ($-\Delta u = g$) in order the explain fundamental aspects of numerical analysis of DG methods in the most simple way. We present IPG formulation and the main a priori error estimates. Furthermore, in Chapter 3 we extend IPG methods to the solution of nonstationary convection-diffusion equation (1.4) and present the corresponding error estimates. We emphasize the fact that the order of accuracy of DG schemes depends on the regularity of the exact solution. Moreover, a set of numerical experiments verifies theoretical results. Finally, Chapter 4 contains an application of IPG methods to the system of the Navier-Stokes equations (1.1). We focus on the solution of the linear algebra systems arising from DG discretization of (1.1) and discuss the use of several linear algebra solvers, preconditioning and stopping criteria.

Chapter 2

Poisson equation

2.1 Continuous problem

Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 2, 3$, with a boundary $\partial\Omega$. We denote by $\partial\Omega_D$ and $\partial\Omega_N$ parts of the boundary $\partial\Omega$, such that $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$ on which the Dirichlet and the Neumann boundary conditions are prescribed, respectively.

We consider the following Poisson model problem: find a function $u : \Omega \to \mathbb{R}$ such that

$$-\Delta u(x) = g(x), \quad x \in \Omega, \tag{2.1}$$

$$u = u_D, \quad \text{on } \partial\Omega_D, \tag{2.2}$$

$$\vec{n} \cdot \nabla u = g_N, \quad \text{on } \partial\Omega_N, \tag{2.3}$$

where $g \in L^2(\Omega)$, u_D is a trace of some suitable function, $g_N \in L^2(\partial\Omega_N)$ and $\vec{n}$ is a unit outer normal to $\partial\Omega$.

2.2 Discretization

Let $\mathcal{T}_h$ ($h > 0$) be a partition of the closure $\overline{\Omega}$ of the domain Ω into a finite number of closed d-dimensional simplexes K with mutually disjoint interiors such that

$$\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} K. \tag{2.4}$$

We call $\mathcal{T}_h$ a *triangulation* of Ω and do not require the standard conforming properties from the finite element method, introduced e.g. in [Cia79], [BS94], [Joh88], [Sch00] or [Žen90]. In two-dimensional problems ($d = 2$) we choose $K \in \mathcal{T}_h$ as triangles and in three-dimensional problems ($d = 3$) the elements $K \in \mathcal{T}_h$ are tetrahedra. As we see, we admit that in the finite element mesh the so-called *hanging nodes* (and in 3D also hanging edges) appear, see Figure 2.1, left.

In general, the discontinuous Galerkin method can handle with more general elements as quadrilaterals and convex or even nonconvex star-shaped polygons in 2D and hexahedra, pyramids and convex or nonconvex star-shaped polyhedra in 3D.

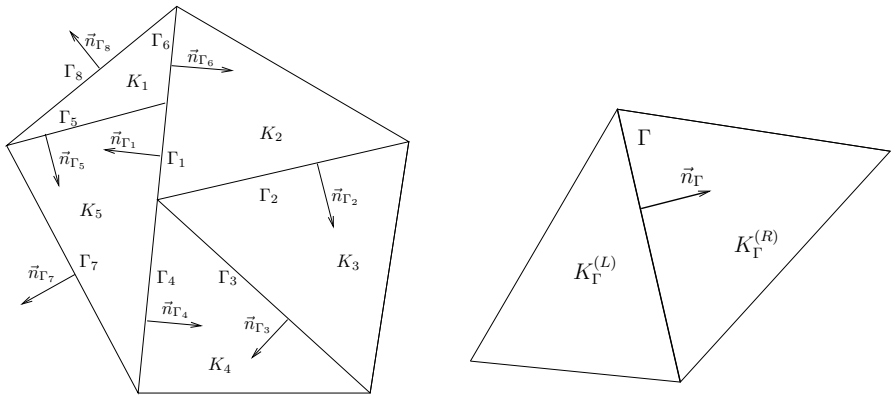


Figure 2.1: Example of elements $K_l,\ l = 1, \ldots, 5$, and faces $\Gamma_l,\ l = 1, \ldots, 8$, with the corresponding normals $\vec{n}_{\Gamma_l}$ (left) and interior face Γ, elements $K_\Gamma^{(L)}$ and $K_\Gamma^{(R)}$ and the orientation of $\vec{n}_\Gamma$ (right)

In our further considerations we shall use the following notation. By ∂K we denote the boundary of an element $K \in \mathcal{T}_h$ and set $h_K = \mathrm{diam}(K) =$ diameter of K, $h = \max_{K \in \mathcal{T}_h} h_K$. Let $K, K' \in \mathcal{T}_h$. We say that K and K' are *neighbours*, if the set $\partial K \cap \partial K'$ has positive $(d-1)$-dimensional measure. We say that $\Gamma \subset K$ is a *face* of K, if it is a maximal connected open subset either of $\partial K \cap \partial K'$, where K' is a neighbour of K, or of $\partial K \cap \partial \Omega$. By $\mathcal{F}_h$ we denote the system of all faces of all elements $K \in \mathcal{T}_h$. Further, we define the set of all inner faces by

$$\mathcal{F}_h^I = \{\Gamma \in \mathcal{F}_h;\ \Gamma \subset \Omega\}\,, \tag{2.5}$$

the set of all "Dirichlet" boundary faces by

$$\mathcal{F}_h^D = \{\Gamma \in \mathcal{F}_h;\ \Gamma \subset \partial\Omega_D\} \tag{2.6}$$

and the set of all "Neumann" boundary faces by

$$\mathcal{F}_h^N = \{\Gamma \in \mathcal{F}_h,\ \Gamma \subset \partial\Omega_N\}\,. \tag{2.7}$$

Obviously, $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^D \cup \mathcal{F}_h^N$. For a shorter notation we put

$$\mathcal{F}_h^{ID} = \mathcal{F}_h^I \cup \mathcal{F}_h^D\,, \qquad \mathcal{F}_h^{DN} = \mathcal{F}_h^D \cup \mathcal{F}_h^N\,. \tag{2.8}$$

For each $\Gamma \in \mathcal{F}_h$ we define a unit normal vector $\vec{n}_\Gamma$. We assume that for $\Gamma \in \mathcal{F}_h^{DN}$ the normal $\vec{n}_\Gamma$ has the same orientation as the outer normal to $\partial\Omega$. For each face $\Gamma \in \mathcal{F}_h^I$ the orientation of $\vec{n}_\Gamma$ is arbitrary but fixed, see Figure 2.1, left.

Over a triangulation $\mathcal{T}_h$ we define the so-called *broken Sobolev space*

$$H^k(\Omega, \mathcal{T}_h) = \{v; v|_K \in H^k(K)\ \forall\, K \in \mathcal{T}_h\} \tag{2.9}$$

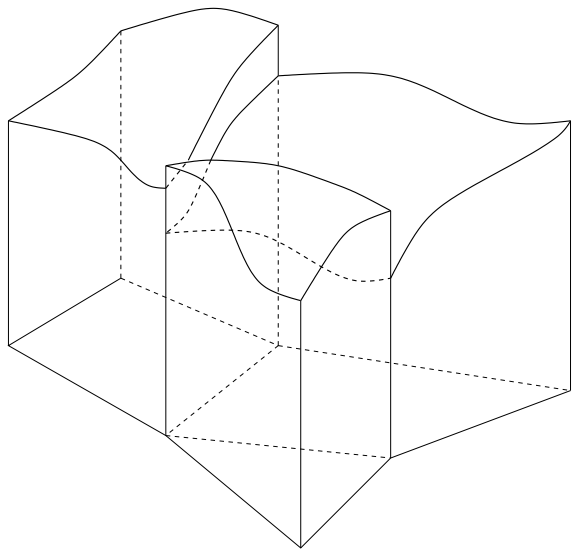


Figure 2.2: Example of a function from $S_{hp} \subset H^1(\Omega, \mathcal{T}_h)$

with the seminorm

$$|v|_{H^k(\Omega,\mathcal{T}_h)} = \left(\sum_{K \in \mathcal{T}_h} |v|^2_{H^k(K)} \right)^{1/2}, \tag{2.10}$$

where $|\cdot|_{H^k(K)}$ denotes the standard seminorm on the Sobolev space $H^k(K)$, $K \in \mathcal{T}_h$, see [Cia79], [BS94].

For each $\Gamma \in \mathcal{F}_h^I$ there exist two neighbouring elements $K_\Gamma^{(L)}, K_\Gamma^{(R)} \in \mathcal{T}_h$ such that $\Gamma \subset \partial K_\Gamma^{(L)} \cap \partial K_\Gamma^{(R)}$. We use a convention that $\vec{n}_\Gamma$ is the outer normal to the element $\partial K_\Gamma^{(L)}$ and the inner normal to the element $\partial K_\Gamma^{(R)}$, see Figure 2.1, right.

We call elements $K_\Gamma^{(L)}$, $K_\Gamma^{(R)}$ *neighbours*. For $v \in H^1(\Omega, \mathcal{T}_h)$, we introduce the following notation:

$$\begin{aligned} v|_\Gamma^{(L)} &= \text{the trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma, \\ v|_\Gamma^{(R)} &= \text{the trace of } v|_{K_\Gamma^{(R)}} \text{ on } \Gamma, \\ \langle v \rangle_\Gamma &= \frac{1}{2}\left(v|_\Gamma^{(L)} + v|_\Gamma^{(R)} \right), \\ [v]_\Gamma &= v|_\Gamma^{(L)} - v|_\Gamma^{(R)}. \end{aligned} \tag{2.11}$$

The value $[v]_\Gamma$ depends on the orientation of $\vec{n}_\Gamma$, but the value $[v]_\Gamma \vec{n}_\Gamma$ is independent of this orientation.

For $\Gamma \in \mathcal{F}_h^{DN}$ there exists element $K_\Gamma^{(L)} \in \mathcal{T}_h$ such that $\Gamma \subset K_\Gamma^{(L)} \cap \partial\Omega$. Then for

$v \in H^1(\Omega, \mathcal{T}_h)$, we introduce the following notation:

$$\begin{aligned} v|_\Gamma^{(L)} &= \text{the trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma, \\ \langle v \rangle_\Gamma &= [v]_\Gamma = v|_\Gamma^{(L)}. \end{aligned} \tag{2.12}$$

For $\Gamma \in \mathcal{F}_h^{DN}$ by $v|_\Gamma^{(R)}$ we formally denote the exterior trace of v on Γ given either by a boundary condition or by an extrapolation from the interior of Ω.

In case that $[\cdot]_\Gamma$, $\langle \cdot \rangle_\Gamma$ and $\vec{n}_\Gamma$ appear in the integrals $\int_\Gamma \dots \, \mathrm{d}S$, $\Gamma \in \mathcal{F}_h$, we omit the subscript Γ and write simply $[\cdot]$, $\langle \cdot \rangle$ and $\vec{n}$, respectively.

Moreover, we put

$$h_\Gamma = \begin{cases} \max(h_{K_\Gamma^{(L)}},\ h_{K_\Gamma^{(R)}}) & \text{for } \Gamma \in \mathcal{F}_h^I, \\ h_{K_\Gamma^{(L)}} & \text{for } \Gamma \in \mathcal{F}_h^{DN}. \end{cases} \tag{2.13}$$

We recall that $h_{K_\Gamma^{(L)}}$ and $h_{K_\Gamma^{(R)}}$ are the diameters of the elements $K_\Gamma^{(L)}$ and $K_\Gamma^{(R)}$, respectively, adjacent to an interior face Γ, and $h_{K_\Gamma^{(L)}}$ denotes the diameter of the element $K_\Gamma^{(L)}$ adjacent to a boundary face Γ.

As we already mentioned in Introduction, DG methods are based on the use of discontinuous piecewise polynomial approximations. Therefore, we define

$$S_{hp} = \{v; v|_K \in P_p(K)\ \forall K \in \mathcal{T}_h\}, \tag{2.14}$$

where $P_p(K)$ denotes the space of all polynomials on K of degree $\le p$. We call the number p the degree of polynomial approximation. See Figure 2.2 for an example of a function from S_{hp} on a fictitious grid.

2.3 Interior penalty Galerkin formulations

In this section we shall describe the interior penalty discontinuous Galerkin (IPG) technique for the solution of problem (2.1) – (2.3). The approximate solution will be sought in the space $S_{hp} \subset H^1(\Omega, \mathcal{T})$.

Let us assume that u is a sufficiently regular solution of (2.1) – (2.3). We multiply (2.1) by a function $v \in H^2(\Omega, \mathcal{T}_h)$ integrate over $K \in \mathcal{T}$ and use Green's theorem. Summing over all $K \in \mathcal{T}_h$, we obtain the identity

$$\sum_{K \in \mathcal{T}_h} \int_K \nabla u \cdot \nabla v \, \mathrm{d}x - \sum_{K \in \mathcal{T}_h} \int_{\partial K} (\vec{n}_K \cdot \nabla u)\, v \, \mathrm{d}S = \int_\Omega g\, v \, \mathrm{d}x, \tag{2.15}$$

where $\vec{n}_K$ denotes the unit outer normal to ∂K. The surface integrals over ∂K make sense due to the regularity of u. We split them according to the type of faces Γ that form the boundary of the element $K \in \mathcal{T}_h$:

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \int_{\partial K} (\vec{n}_K \cdot \nabla u)\, v \, \mathrm{d}S &= \sum_{\Gamma \in \mathcal{F}_h^D} \int_\Gamma (\vec{n}_\Gamma \cdot \nabla u)\, v \, \mathrm{d}S + \sum_{\Gamma \in \mathcal{F}_h^N} \int_\Gamma (\vec{n}_\Gamma \cdot \nabla u)\, v \, \mathrm{d}S \\ &\quad + \sum_{\Gamma \in \mathcal{F}_h^I} \int_\Gamma \vec{n}_\Gamma \cdot \left((\nabla u|_\Gamma^{(L)})\, v|_\Gamma^{(L)} - (\nabla u|_\Gamma^{(R)})\, v|_\Gamma^{(R)} \right) \mathrm{d}S. \end{aligned} \tag{2.16}$$

There is a sign "−" in the last integral, since n_Γ is the unit outer normal to $\partial K_\Gamma^{(L)}$ but the unit inner normal to $\partial K_\Gamma^{(R)}$, see Figure 2.1, right.

Since u is the regular function, we have

$$[u]_\Gamma = 0, \quad \nabla u|_\Gamma^{(L)} = \nabla u|_\Gamma^{(R)} = \langle \nabla u \rangle_\Gamma, \quad \Gamma \in \mathcal{F}_h^I. \tag{2.17}$$

Thus, using (2.12)the integrand of the last integral in (2.16) can by written in the form

$$\vec{n}_\Gamma \cdot (\nabla u)|_\Gamma^{(L)} \, v|_\Gamma^{(L)} - \vec{n}_\Gamma \cdot (\nabla u)|_\Gamma^{(R)} \, v|_\Gamma^{(R)} = \vec{n}_\Gamma \cdot \langle \nabla u \rangle_\Gamma \, [v]_\Gamma. \tag{2.18}$$

Hence, due to (2.3), (2.12) and (2.15) – (2.18), we have

$$\begin{aligned} & \sum_{K \in \mathcal{T}_h} \int_K \nabla u \cdot \nabla v \, \mathrm{d}x - \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \vec{n} \cdot \langle \nabla u \rangle \, [v] \, \mathrm{d}S \\ = & \int_\Omega g \, v \, \mathrm{d}x + \sum_{\Gamma \in \mathcal{F}_h^N} \int_\Gamma g_N \, v \, \mathrm{d}S, \qquad v \in H^2(\Omega, \mathcal{T}_h). \end{aligned} \tag{2.19}$$

Moreover, for $u, v \in H^1(\Omega, \mathcal{T}_h)$ we define the *interior penalty* bilinear form

$$J_h^\sigma(u, v) = \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sigma [u] \, [v] \, \mathrm{d}S \tag{2.20}$$

and the *boundary penalty* linear form

$$J_D^\sigma(v) = \sum_{\Gamma \in \mathcal{F}_h^D} \int_\Gamma \sigma u_D \, v \, \mathrm{d}S, \tag{2.21}$$

where $\sigma > 0$ is a penalty weight. Its choice will be discussed later.

If $u \in H^1(\Omega) \cap H^2(\Omega, \mathcal{T}_h)$ and u satisfies the Dirichlet boundary condition (2.2), then

$$\sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \vec{n} \cdot \langle \nabla v \rangle \, [u] \, \mathrm{d}S = \sum_{\Gamma \in \mathcal{F}_h^D} \int_\Gamma \vec{n} \cdot \nabla v \, u_D \, \mathrm{d}S \quad \forall v \in H^2(\Omega, \mathcal{T}_h) \tag{2.22}$$

and

$$J_h^\sigma(u, v) = J_D^\sigma(v) \qquad \forall v \in H^2(\Omega, \mathcal{T}_h), \tag{2.23}$$

since $[u]_\Gamma = 0$ for $\Gamma \in \mathcal{F}_h^I$ and $[u]_\Gamma = u|_\Gamma = u_D$ for $\Gamma \in \mathcal{F}_h^D$.

Now, we introduce three variants of the *discontinuous Galerkin weak formulation* in such a way that we sum (2.19) with (2.23) and with -1, 0 or 1-multiple of (2.22). This leads us to the following notation. For $u, v \in H^2(\Omega, \mathcal{T}_h)$ we set

$$a_h(u, v) = \sum_{K \in \mathcal{T}_h} \int_K \nabla u \cdot \nabla v \, \mathrm{d}x - \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \left(\vec{n} \cdot \langle \nabla u \rangle \, [v] + \theta \vec{n} \cdot \langle \nabla v \rangle \, [u] \right) \mathrm{d}S \tag{2.24}$$

and

$$F_h(v) = \int_\Omega g\, v \,\mathrm{d}x + \sum_{\Gamma \in \mathcal{F}_h^N} \int_\Gamma g_N\, v \,\mathrm{d}S + \theta \sum_{\Gamma \in \mathcal{F}_h^D} \int_\Gamma \vec{n} \cdot \nabla v\, u_D \,\mathrm{d}S, \tag{2.25}$$

where $\theta = -1, 0, 1$ (see below).

Moreover, for $u, v \in H^2(\Omega, \mathcal{T}_h)$ let us define the bilinear form

$$\mathcal{B}_h(u, v) = a_h(u, v) + J_h^\sigma(u, v) \tag{2.26}$$

and the linear form

$$\ell_h(v) = F_h(v) + J_D^\sigma(v). \tag{2.27}$$

Since $S_{hp} \subset H^2(\Omega, \mathcal{T}_h)$, the forms (2.26) – (2.27) make sense for $u_h, v_h \in S_{hp}$. Consequently, we define the numerical scheme.

Definition 1 *A function $u_h \in S_{hp}$ is called a* DG approximate solution *of problem (2.1) – (2.3), if it satisfies the following identity*

$$\mathcal{B}_h(u_h, v_h) = \ell_h(v_h) \quad \forall v_h \in S_{hp}, \tag{2.28}$$

where the forms $\mathcal{B}_h$ and ℓ_h are defined by (2.26) and (2.27), respectively and $\theta = 1$ (we speak about SIPG method) or $\theta = -1$ (NIPG method) and $\theta = 0$ (IIPG method).

From the construction of the forms $\mathcal{B}_h$ and ℓ_h one can see that the strong solution $u \in H^2(\Omega)$ of problem (2.1) – (2.3) satisfies the identity

$$\mathcal{B}_h(u, v) = \ell_h(v) \quad \forall v \in H^2(\Omega, \mathcal{T}_h), \tag{2.29}$$

which represents the *consistency* of the method. The expressions (2.28) and (2.29) imply the so-called *Galerkin orthogonality* of the error $e_h = u_h - u$ of the method:

$$\mathcal{B}_h(e_h, v_h) = 0 \quad \forall v_h \in S_{hp}, \tag{2.30}$$

which will be used in the analysis of error estimates.

In contrast to standard conforming finite element techniques, both Dirichlet and Neumann boundary conditions are included automatically in formulation (2.28) of the discrete problem. This is an advantage particularly in the case of nonhomogeneous Dirichlet boundary conditions, because it is not necessary to construct subsets of finite element spaces formed by functions approximating the Dirichlet boundary condition in a suitable way.

2.4 Numerical analysis

We are interested in a priori error estimates of the difference of numerical solution given by (2.28) and the exact one. Therefore, we define the following mesh-dependent norm

$$|||u||| \quad = \quad \left(|u|^2_{H^1(\Omega, \mathcal{T}_h)} + J_h^\sigma(u, u) \right)^{1/2}, \quad u \in H^1(\Omega, \mathcal{T}_h). \tag{2.31}$$

We define the *penalty weight* σ by

$$\sigma|_\Gamma = \frac{C_W}{h_\Gamma}, \quad \Gamma \in \mathcal{F}_h \tag{2.32}$$

where $C_W > 0$ is a suitable constant and $h_\Gamma,\ \Gamma \in \mathcal{F}_h$ is given by (2.13).

In order to analyse IPG methods we assume that $\{\mathcal{T}_h\}_{h\in(0,h_0)},\ h_0 > 0$, is a system of partitions of the domain Ω ($\mathcal{T}_h = \{K\}_{K\in\mathcal{T}_h}$) satisfying assumptions

(A1) The system $\{\mathcal{T}_h\}_{h\in(0,h_0)}$ is *shape regular*: there exists a positive constant C_R such that

$$\frac{h_K}{\rho_K} \le C_R \quad \forall K \in \mathcal{T}_h,\ \forall h \in (0, h_0), \tag{2.33}$$

where ρ_K is the radius of the largest d-dimensional ball inscribed into $K,\ K \in \mathcal{T}_h$.

(A2) The system $\{\mathcal{T}_h\}_{h\in(0,h_0)}$ is *locally quasi-uniform*: there exists a constant $C_H > 0$ such that

$$h_K \le C_H h_{K'} \quad \forall K, K' \in \mathcal{T}_h,\ K, K' \text{ are neighbours},\ \forall h \in (0, h_0). \tag{2.34}$$

2.4.1 Auxiliary results

The numerical analysis of IPG methods is based on the following fundamental assertions which are valid for triangulations satisfying assumptions (A1) – (A2).

- (*Multiplicative trace inequality*) There exists a constant $C_M > 0$ independent of v, h and K such that

$$\|v\|^2_{L^2(\partial K)} \le C_M \left(\|v\|_{L^2(K)}\,|v|_{H^1(K)} + h_K^{-1}\|v\|^2_{L^2(K)}\right), \quad v \in H^1(K),\ K \in \mathcal{T}_h,\ h \in (0, h_0). \tag{2.35}$$

- *(Inverse inequality)* There exists a constant $C_I > 0$ independent of v, h and K such that

$$|v|_{H^1(K)} \le C_I h_K^{-1}\|v\|_{L^2(K)}, \quad \forall v \in P_p(K),\ \forall K \in \mathcal{T}_h,\ \forall h \in (0, h_0). \tag{2.36}$$

- *(Approximation properties)* There exists a mapping $\Pi_{hp} : H^1(\Omega, \mathcal{T}_h) \to S_{hp}$ and a constant $C_A > 0$ such that

$$|\Pi_{hp}v - v|_{H^q(\Omega,\mathcal{T}_h)} \le C_A h^{\mu-q}|v|_{H^\mu(\Omega,\mathcal{T}_h)} \quad \forall v \in H^\mu(\Omega, \mathcal{T}_h)\ \forall h \in (0, h_0), \tag{2.37}$$

where $\mu = \min(p + 1, s),\ 0 \le q \le s$ and p, s are integers.

See [DF05] for the references to the proofs of (2.35) – (2.37).

2.4.2 Coercivity of bilinear form

Since the form $\mathcal{B}_h$ is bilinear, the existence and uniqueness of the approximate solution of (2.28) follows from the coercivity of the form $\mathcal{B}_h$. We say that form $\mathcal{B}_h$ is *coercive* if there exists a constant $C_C > 0$ such that

$$\mathcal{B}_h(v_h, v_h) \geq C_C |||v_h|||^2 \quad \forall v_h \in S_{hp}. \tag{2.38}$$

In order to ensure the coercivity of $\mathcal{B}_h$ it is necessary to choose constant C_W from (2.32) in a suitable way, namely

$$\begin{array}{ll} C_W > 0 \text{ (e.g., } C_W = 1) & \text{for NIPG method,} \\ C_W \geq 4C_H C_M(1 + C_I) & \text{for SIPG method,} \\ C_W \geq C_H C_M(1 + C_I) & \text{for IIPG method.} \end{array} \tag{2.39}$$

If $\theta = -1$ (NIPG method) then relations (2.20), (2.24) and (2.26) immediately yield (2.38) for any $C_W > 0$. For SIPG and IIPG techniques, see, e.g., [DF05].

Remark 1 *The coercivity of the NIPG technique for any $C_W > 0$ is a big advantage namely for the solution of the system of the Navier-Stokes equations. The lack of theory for the Navier-Stokes equations does not give any analytical relation for C_W similar to (2.39). It is possible to choose C_W "sufficiently" large but then the resulting system of linear algebra equations becomes ill-conditioned.*

Now, we are ready to formulate a priori error estimates for the IPG methods. If u and u_h denote the exact solution of problem (2.1) – (2.3) and the approximate solution obtained by method (2.29), respectively, we define the error $e_h = u_h - u$. It can be written in the form

$$e_h = \eta + \xi, \quad \text{with } \eta = \Pi_{hp} u - u,\ \xi = u_h - \Pi_{hp} u \in S_{hp}, \tag{2.40}$$

where Π_{hp} is the S_{hp}-interpolation defined by (2.37).

2.4.3 Error estimate in the H^1-seminorm

Theorem 1 *Let us assume that $s \geq 1$, $u \in H^s(\Omega)$ is the solution of problem (2.1) – (2.3), $\{\mathcal{T}_h\}_{h \in (0,h_0)}$ is a system of triangulations of the domain Ω satisfying assumptions* (A1) *and* (A2), *S_{hp} is the space of discontinuous piecewise polynomial functions (2.14) and $u_h \in S_{hp}$ is the approximate solution obtained by (2.28) with C_W chosen according to (2.39). Then*

$$|||e_h||| \leq C_1 h^{\mu - 1} |u|_{H^\mu(\Omega)}, \quad h \in (0, h_0), \tag{2.41}$$

where $e_h = u_h - u$, $\mu = \min(p + 1, s)$ and $C_1 > 0$ is a constant independent of h.

The estimate (2.41) can be derived in several steps. In order to keep the purpose of these notes we present only sketch of each step. For a detailed proof we refer, e.g., [ABCM02], [DFS05].

1. We express the error by (2.40), i.e. $e_h = u_h - u = \eta + \xi$. The error e_h satisfies the Galerkin orthogonality (2.30), which is equivalent to

$$\mathcal{B}_h(\xi, v_h) = -\mathcal{B}_h(\eta, v_h) \qquad \forall v_h \in S_{hp}. \tag{2.42}$$

   If we set $v_h := \xi \in S_{hp}$ in (2.42) and use (2.26) and the coercivity (2.38), we find that

$$C_C |||\xi|||^2 \leq |a_h(\xi, \eta)| + |J_h^\sigma(\eta, \xi)|, \tag{2.43}$$

   where C_C is the constant from (2.38).

2. Using inequalities (2.35) – (2.37) we find that

$$|a_h(\eta, \xi)| \leq C_a h^{\mu-1} |u|_{H^\mu(\Omega)} |||\xi|||, \tag{2.44}$$

   where $C_a > 0$ is a constant independent of h.

3. Furthermore, in view of the Cauchy inequality, (2.31) and (2.35) – (2.37), we have

$$\begin{aligned} |J_h^\sigma(\eta, \xi)| &\leq (J_h^\sigma(\eta, \eta))^{1/2} \, (J_h^\sigma(\xi, \xi))^{1/2} \\ &\leq C_J h^{\mu-1} |u|_{H^\mu(\Omega)} \, |||\xi|||, \end{aligned} \tag{2.45}$$

   where $C_J > 0$ is a constant independent of h.

4. Using (2.43) – (2.45), we get

$$|||\xi|||^2 \leq (C_a + C_J) C_C^{-1} h^{\mu-1} |u|_{H^\mu(\Omega)} |||\xi|||, \tag{2.46}$$

   which gives

$$|||\xi||| \leq \bar{C} h^{\mu-1} |u|_{H^\mu(\Omega)} \tag{2.47}$$

   with $\bar{C} = (C_a + C_J)/C_C$.

5. Finally, (2.31), (2.40), the triangle inequality

$$|||e_h||| \leq |||\xi||| + |||\eta|||, \tag{2.48}$$

   and (2.37) imply that

$$|||e_h||| \leq \bar{C} h^{\mu-1} |u|_{H^\mu(\Omega)} + (C_A^2 + C_J^2)^{1/2} h^{\mu-1} |u|_{H^\mu(\Omega)}. \tag{2.49}$$

   Hence, (2.41) holds with $C_1 = (C_a + C_J) C_C^{-1} + (C_A^2 + C_J^2)^{1/2}$.

2.4.4 Error estimate in the L^2-norm

The error estimate in the $L^2(\Omega)$-norm follows directly from (2.41) and the *broken Poincaré inequality*

$$\|v_h\|_{L^2(\Omega)}^2 \leq C_P |||v_h||| \quad \forall v_h \in S_{hp}, \ \forall h \in (0, h_0), \tag{2.50}$$

where $C_P > 0$ is a constant independent of v_h and h, see [Bre03]. Therefore, the error $e_h = u_h - u$ satisfies

$$\|e_h\|_{L^2(\Omega)} \leq C_1 C_P h^{\mu-1} |u|_{H^\mu(\Omega)}, \quad h \in (0, h_0). \tag{2.51}$$

Remark 2 *The error estimate (2.51), which is the order of $O(h^p)$ (for $s = p+1$), is suboptimal with respect to the approximation property of the piecewise polynomial space S_{hp} with $q = 0$, $\mu = s = p+1$, (cf. (2.37)) giving the order $O(h^{p+1})$. It is possible to prove an optimal error estimate in the $L^2(\Omega)$-norm using the Aubin-Nitsche technique, but for SIPG method only, since the symmetry of $\mathcal{B}_h$ is required. The suboptimality of the order of accuracy measured in the L^2-norm is a disadvantage of NIPG and IIPG methods. However, numerical experiments show the optimal order of accuracy measured in the L^2-norm for odd degrees of polynomial approximations.*

2.4.5 Summary of the analysis of IPG methods

Based on the presented results of numerical analysis we can conclude that

- Only SIPG method gives the optimal order of accuracy in the $L^2(\Omega)$-norm (see Remark 2). On the other hand, the constant C_W appearing in the penalty parameter σ (cf. (2.32)) should be chosen according to (2.39). However, a similar analytical relation is missing for the Navier-Stokes equations.
- NIPG technique does not give the optimal order of accuracy in the $L^2(\Omega)$-norm but the constant $C_W > 0$ can be arbitrary.
- IIPG method has both disadvantages: suboptimal order of convergence in the L^2-norm and the necessity to choose C_W according to (2.39). On the other hand, some terms disappear in the definition of a_h and F_h in (2.24) and (2.25), respectively (since $\theta = 0$). This is a big advantage of IIPG technique particularly for the system of the Navier-Stokes equations, where the choice of the stabilization terms is a complicated. Consequently, the implementation of the IIPG method is simpler.

These observations are without any modification valid for the convection-diffusion problems discussed in Chapter 3.

2.5 Numerical experiments

In this section, we shall justify the a priori error estimates (2.41) and (2.51). In the first example, we assume that the exact solution is sufficiently regular. We show that the use of a higher degree of polynomial approximation increases the rate of convergence of the method. In the second example, the exact solution has a singularity. Then the order of convergence does not increase with the increasing degree of the used polynomial approximation. The computational results are in agreement with theory and show that the accuracy of the method is determined by the degree of the polynomial approximation as well the regularity of the solution.

We consider the Poisson equation

$$-\Delta u = g \quad \text{in } \Omega, \qquad \Omega = (0,1)^2 \tag{2.52}$$

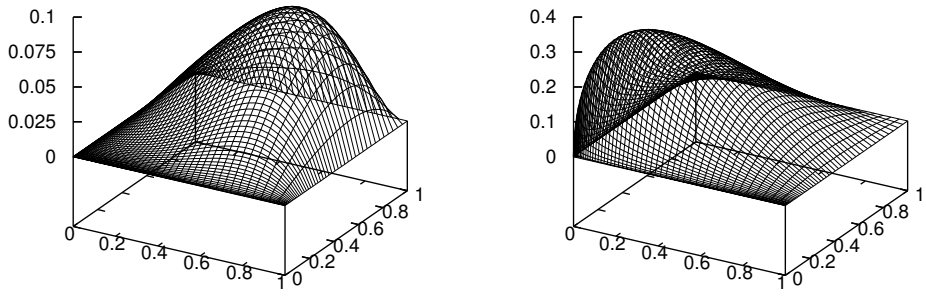


Figure 2.3: The exact solution (2.53) for $\alpha = 2$ (left) and $\alpha = -3/2$ (right)

with the homogeneous Dirichlet boundary condition on $\partial\Omega$. We define the function g in such a way that the exact solution has the form

$$u(x_1, x_2) = 2r^{\alpha} x_1 x_2 (1 - x_1)(1 - x_2) = r^{\alpha+2} \sin(2\varphi)(1 - x_1)(1 - x_2), \tag{2.53}$$

where (r, φ) $(r = (x_1^2 + x_2^2)^{1/2})$ are the polar coordinates and $\alpha \in I\!R$ is a constant. The regularity of u is depends on the value of α, namely (cf. [BS90])

$$u \in H^{\beta}(\Omega) \quad \forall \beta \in (0,\, \alpha + 3), \tag{2.54}$$

where $H^{\beta}(\Omega)$ denotes (in general) the Sobolev-Slobodetskii space of functions with "noninteger derivatives".

In the presented numerical tests we use the values $\alpha = 2$ and $\alpha = -3/2$. The value $\alpha = 2$ gives function u sufficiently regular ($\in H^{\beta}(\Omega)$ for $\beta < 5$), whereas the value $\alpha = -3/2$ gives $u \in H^{\beta}(\Omega),\ \beta < 3/2$. Figure 2.3 shows functions u for both values of α.

Numerical experiments were carried out with the use of P^1, P^2, P^3 and P^4 polynomial approximations on 7 triangular meshes having 128, 288, 512, 1152, 2048, 4608 and 8192 elements, see Figure 2.4.

Figures 2.5 - 2.8 show computational errors in the $L^2(\Omega)$-norm and in the H^1-seminorm and indicate the corresponding experimental orders of convergence (EOC) for $\alpha = 2$ and $\alpha = -3/2$ using SIPG, NIPG and IIPG methods. We observe that

- **regular exact solution (case $\alpha = 2$)**
  - SIPG method gives optimal order of convergence $O(h^{p+1})$ for $p = 1, 2, 3, 4$,
  - NIPG and IIPG methods give optimal order of convergence $O(h^{p+1})$ for $p = 1$ and $p = 3$ (even degrees) and suboptimal $O(h^p)$ for $p = 2$ and $p = 4$ (odd degrees).

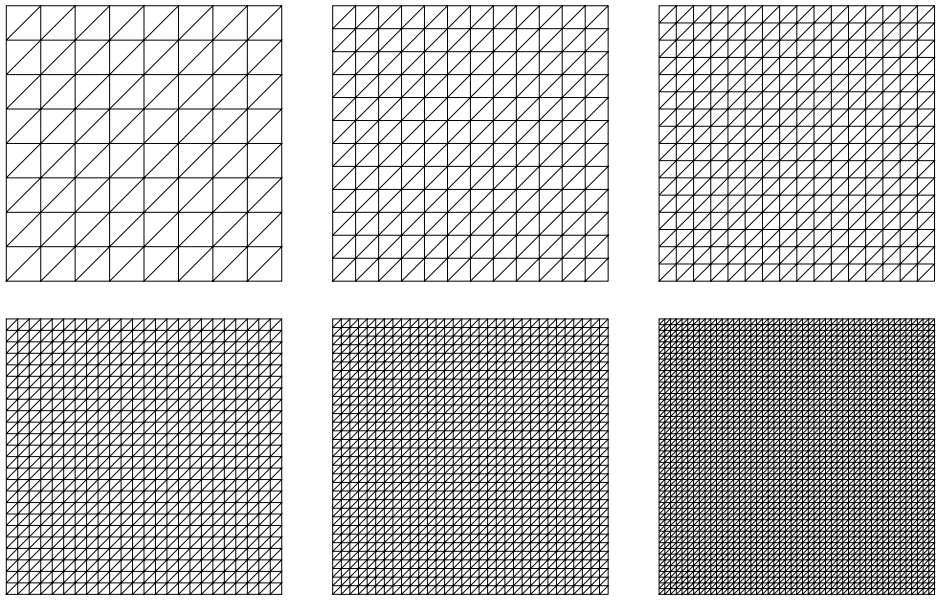


Figure 2.4: Computational grids except the finest one

- All IPG technique produce optimal order of convergence in th H^1-seminorm ($O(h^p)$).

- **singular exact solution (case $\alpha = -3/2$)**

  - The experimental order of convergence in the L^2-norm is equal to $3/2$ for $p = 1, 2, 3, 4$. Using the result from [Fei89], for any $\beta \in (1, 3/2)$ we get

    $$\|v - I_h v\|_{L^2(\Omega)} \leq C(\beta) h^\mu \|v\|_{H^\beta(\Omega)}, \quad v \in H^\beta(\Omega), \tag{2.55}$$

    where $I_h v$ is a piecewise polynomial Lagrange interpolation to v of degree $\leq p$, $\mu = \min(p + 1, \beta)$ and $C(\beta)$ is a constant independent of h and v. The exact approximation of order $O(h^{3/2})$, corresponding precisely to our experimental results, can be obtained with the use of the interpolation in the so-called Besov spaces. See [BS01], Section 3.3* and the references therein.

  - Similarly, the experimental order of convergence in the H^1-seminorm is equal to $1/2$ for $p = 1, 2, 3, 4$ which is in agreement with similar theoretical results.

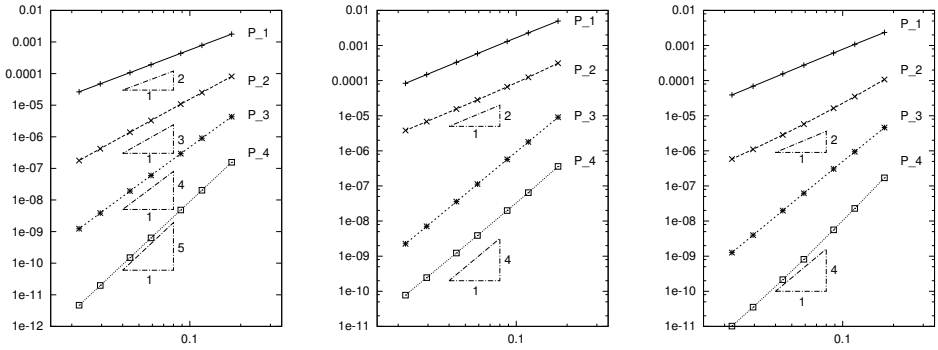


Figure 2.5: Computational error and EOC in the L^2-norm for the SIPG (left), NIPG (middle), IIPG (right) method, the regular solution ($\alpha = 2$)

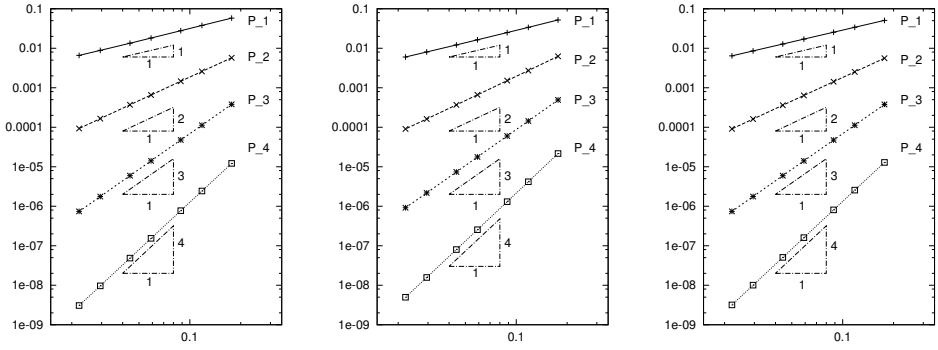


Figure 2.6: Computational error and EOC in the H^1-semi-norm for the SIPG (left), NIPG (middle), IIPG (right) method, the regular solution ($\alpha = 2$)

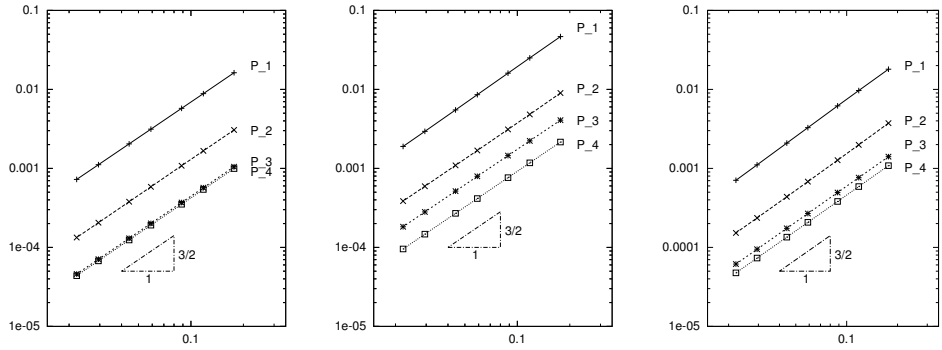


Figure 2.7: Computational error and EOC in the L^2-norm for the SIPG (left), NIPG (middle), IIPG (right) method, the solution having a singularity ($\alpha = -3/2$)

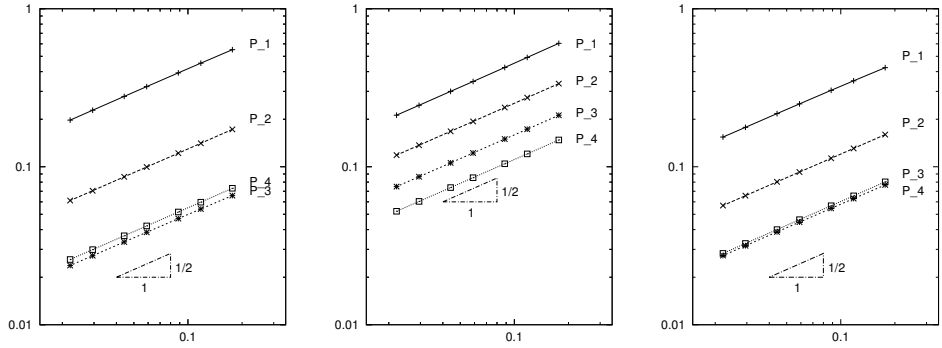


Figure 2.8: Computational error and EOC in the H^1-semi-norm for the SIPG (left), NIPG (middle), IIPG (right) method, the solution having a singularity ($\alpha = -3/2$)

Chapter 3

Convection-diffusion problem

Within this section we extend the application of the DGFEM to the nonstationary nonlinear convection-diffusion equation (1.4). In order to avoid these notes to be too technical we present the IPG formulations of (1.4) and without proof we present a priori error estimates. We underline some links to elliptic problem analyzed in Chapter 2 and present some numerical experiments.

3.1 Continuous problem

Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be a bounded polygonal (if $d = 2$) or polyhedral (if $d = 3$) domain with Lipschitz-continuous boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\partial\Omega_D \cap \partial\Omega_N = \emptyset$, and $T > 0$. We are concerned with the following nonstationary nonlinear convection-diffusion problem: find $u : Q_T = \Omega \times (0, T) \to \mathbb{R}$ such that

$$\begin{aligned}
&\text{a)} \quad \frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = \varepsilon \Delta u + g \quad \text{in } Q_T, \\
&\text{b)} \quad u\big|_{\partial\Omega_D \times (0,T)} = u_D, \\
&\text{c)} \quad \varepsilon \frac{\partial u}{\partial n}\Big|_{\partial\Omega_N \times (0,T)} = g_N, \\
&\text{d)} \quad u(x, 0) = u^0(x), \quad x \in \Omega.
\end{aligned} \tag{3.1}$$

We assume that the data satisfy the following conditions:

$$\begin{aligned}
&\text{a)} \quad f_s \in C^1(\mathbb{R}),\ s = 1, \ldots, d, \text{ are Lipschitz-continuous}, \\
&\text{b)} \quad \varepsilon > 0, \\
&\text{c)} \quad g \in C([0, T]; L^2(\Omega)), \\
&\text{d)} \quad u_D = \text{trace of some suitable function} \\
&\text{e)} \quad g_N \in C([0, T]; L^2(\partial\Omega_N)), \\
&\text{f)} \quad u^0 \in L^2(\Omega).
\end{aligned} \tag{3.2}$$

The coefficient ε represents a diffusion coefficient, $f_s(\cdot)$, $s = 1, \ldots, d$ are nonlinear convective fluxes and g is a source term.

We use the standard notation for function spaces and their norms $\|\cdot\|$ and seminorms $|\cdot|$ (see, e. g. [KJk77]): $L^q(\Omega)$, $L^q(Q_T)$ denote the Lebesgue spaces, $W^{k,q}(\Omega)$, $H^k(\Omega) = W^{k,2}(\Omega)$ are the Sobolev spaces, $W^{k,q}(0,T;X)$ is the Bochner space of functions whose weak k^{th}-time derivative is q-integrable over the interval $(0,T)$ with values in a Banach space X, $W^{k,\infty}(0,T;X)$ is the Bochner space of functions whose weak k^{th} time derivative is bounded a. e. on $(0,T)$ with values in a Banach space X, $C(0,T;X)$ $(C^1(0,T;X))$ is the space of continuous (continuously differentiable) mappings of the interval $[0,T]$ into X. By $H_0^1(\Omega)$ we denote the subspace of all functions from $H^1(\Omega)$ with zero traces on $\partial\Omega$. Moreover, we define the space $V \equiv \left\{w;\ w \in H^1(\Omega),\ w|_{\partial\Omega_D} = 0\right\}$, obviously $H_0^1(\Omega) \subset V \subset H^1(\Omega)$.

A sufficiently regular function satisfying (3.1) pointwise is called a *classical solution*. It is convenient to introduce the concept of a weak solution. To this end we use the following *notation*: $(u,w) \equiv \int_\Omega uw\,\mathrm{d}x$ for $u,\ w \in L^2(\Omega)$, $(u,w)_N \equiv \int_{\partial\Omega_N} uw\,\mathrm{d}S$ for $u,\ w \in L^2(\partial\Omega_N)$, $a(u,w) \equiv \varepsilon\int_\Omega \nabla u \cdot \nabla w\,\mathrm{d}x$ for $u,\ w \in H^1(\Omega)$ and

$$b(u,w) \equiv \int_\Omega \nabla \cdot \boldsymbol{f}(u)\, w\,\mathrm{d}x,\ u \in H^1(\Omega) \cap L^\infty(\Omega),\ w \in L^2(\Omega),$$

The assumption $u \in L^\infty(\Omega)$ in the definition of b guarantees the boundedness of function $\boldsymbol{f}(u)$ and therefore the existence of the integral. This assumption can be weakened if function $\boldsymbol{f}(u)$ satisfies some growth conditions. We do not consider this case for simplicity.

Definition 2 *We say that function u is the* weak solution *of (3.1), if the following conditions are satisfied*

$$\begin{aligned}
&a) && u - u^* \in L^2(0,T;V), \quad u \in L^\infty(Q_T), && (3.3)\\
&b) && \frac{\mathrm{d}}{\mathrm{d}t}(u(t),w) + b(u(t),w) + a(u(t),w) = (g(t),w) + (u_N,w)_N \\
& && \qquad \textit{for all } w \in V \textit{ in the sense of distributions on } (0,T),\\
&c) && u(0) = u^0 \quad \textit{in } \Omega.
\end{aligned}$$

By $u(t)$ we denote the function on Ω such that $u(t)(x) = u(x,t)$, $x \in \Omega$. With the aid of techniques from [Lio96] and [Rek82], it is possible to prove that there exists a unique weak solution in the sense of Definition 2.

3.2 Discretization

In this section, we shall introduce a DG space semi-discretization of problem (3.1). We use the notation and auxiliary results introduced in Section 2.2.

By $\mathcal{T}_h$ $(h > 0)$ we denote a triangulation of the domain Ω introduced in Section 2.2. We start from the strong solution u, multiply equation (3.1), a) by an arbitrary $v \in H^2(\Omega, \mathcal{T}_h)$, integrate over each $K \in \mathcal{T}_h$ and apply Green's theorem. We obtain the

identity

$$\int_K \frac{\partial u(t)}{\partial t}\, v\,\mathrm{d}x + \int_{\partial K} \sum_{s=1}^{d} f_s(u(t)) n_s\, v\,\mathrm{d}S - \int_K \sum_{s=1}^{d} f_s(u(t)) \frac{\partial v}{\partial x_s}\,\mathrm{d}x \qquad (3.4)$$
$$+\varepsilon \int_K \nabla u(t)\cdot\nabla v\,\mathrm{d}x - \varepsilon \int_{\partial K} (\nabla u(t)\cdot\vec{n})\, v\,\mathrm{d}S = \int_K g(t)\, v\,\mathrm{d}x.$$

Here $\vec{n} = (n_1, \ldots, n_d)$ denotes the unit outer normal to ∂K. Summing (3.4) over all $K \in \mathcal{T}_h$, using the relations

$$[u]_\Gamma = 0, \quad \langle \nabla u\rangle_\Gamma = \nabla u|_\Gamma^{(L)} = \nabla u|_\Gamma^{(R)}, \qquad \Gamma \in \mathcal{F}_h^I, \qquad (3.5)$$

valid for the strong solution u, and the boundary conditions (3.1) b) – c), we obtain the identity

$$\left(\frac{\partial u(t)}{\partial t}, v\right) + a_h(u(t), v) + \tilde{b}_h(u(t), v) + \varepsilon J_h^\sigma(u(t), v) = \ell_h(v)(t), \qquad (3.6)$$

where

$$a_h(u,v) = \varepsilon \sum_{K\in\mathcal{T}_h} \int_K \nabla u\cdot\nabla v\,\mathrm{d}x - \varepsilon \sum_{\Gamma\in\mathcal{F}_h^{ID}} \int_\Gamma \left(\langle\nabla u\rangle\cdot\vec{n}[v] + \theta\langle\nabla v\rangle\cdot\vec{n}[u]\right)\mathrm{d}S \qquad (3.7)$$

$$\tilde{b}_h(u,v) = \sum_{K\in\mathcal{T}_h} \int_{\partial K} \sum_{s=1}^{d} f_s(u(t)) n_s\, v\,\mathrm{d}S - \sum_{K\in\mathcal{T}_h} \int_K \sum_{s=1}^{d} f_s(u(t)) \frac{\partial v}{\partial x_s}\,\mathrm{d}x, \qquad (3.8)$$

$$\ell_h(v)(t) = (g(t), v) + \varepsilon\theta \sum_{\Gamma\in\mathcal{F}_h^D} \int_\Gamma u_D(t)\,(\nabla v\cdot\vec{n})\,\mathrm{d}S \qquad (3.9)$$
$$+\varepsilon \sum_{\Gamma\in\mathcal{F}_h^D} \int_\Gamma \sigma u_D(t)\, v\,\mathrm{d}S + (g_N(t), v)_N,$$

$$J_h^\sigma(u,v) = \sum_{\Gamma\in\mathcal{F}_h^{ID}} \int_\Gamma \sigma[u]\,[v]\,\mathrm{d}S, \qquad (3.10)$$

where the *penalty parameter* σ is given (similarly as in (2.32)) by

$$\sigma|_\Gamma = \frac{C_W}{h_\Gamma}, \quad \Gamma \in \mathcal{F}_h, \qquad (3.11)$$

h_Γ represents the "size" of $\Gamma \in \mathcal{F}_h$ defined in Section 2.4 and $C_W > 0$ is a suitable constant. The forms a_h, ℓ_h and J_h^σ are identical with those defined by (2.24), (2.27) and (2.20), respectively.

Similarly as in Chapter 2), the parameter θ is equal to 1 for the symmetric interior penalty Galerkin (SIPG) variant of DGM, $\theta = -1$ for the non-symmetric interior penalty Galerkin (NIPG) variant of DGM and $\theta = 0$ for the incomplete interior penalty Galerkin (IIPG) variant of the DG discretization of the diffusion term.

The identity (3.6) makes sense for functions $u, v \in H^2(\Omega, \mathcal{T}_h)$. In virtue of Section 2.3, we derive the discrete problem in such a way that $u(t)$ in (3.6) is approximated by $u_h(t) \in S_{hp}$ and v is replaced by $v_h \in S_{hp}$. Similarly as in the finite volume method, the fluxes $\int_\Gamma \sum_{s=1}^d f_s(u) n_s \, v \, \mathrm{d}S$, $\Gamma \in \mathcal{F}_h$ are approximated with the aid of the so-called *numerical flux* $H(u, u', \vec{n})$ (see, e.g., [FFS03], [Wes01]):

$$\int_\Gamma \sum_{s=1}^d f_s(u) n_s \, v \, \mathrm{d}S \approx \int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n}) \, v|_\Gamma^{(L)} \, \mathrm{d}S, \qquad \Gamma \in \mathcal{F}_h. \tag{3.12}$$

Of course, if $\Gamma \in \mathcal{F}_h^{DN}$, then it is necessary to specify the meaning of $u|_\Gamma^{(R)}$. It is possible to use the *extrapolation* from the interior of the computational domain

$$u|_\Gamma^{(R)} := u|_\Gamma^{(L)}, \qquad \Gamma \in \partial\Omega, \tag{3.13}$$

or the *Dirichlet boundary condition*

$$u|_\Gamma^{(R)} := u_D|_\Gamma, \qquad \Gamma \in \partial\Omega. \tag{3.14}$$

We shall assume that the numerical flux has the following properties:
Assumptions (H):

1. $H(u, v, \vec{n})$ is defined in $\mathbb{R}^d \times B_1$, where $B_1 = \{\vec{n} \in \mathbb{R}^d; |\vec{n}| = 1\}$, and *Lipschitz-continuous* with respect to u, v:

$$|H(u, v, \vec{n}) - H(u^*, v^*, \vec{n})| \le L_H(|u - u^*| + |v - v^*|), \qquad u, \, v, \, u^*, \, v^* \in \mathbb{R}, \ \vec{n} \in B_1. \tag{3.15}$$

2. $H(u, v, \vec{n})$ is *consistent*:

$$H(u, u, \vec{n}) = \sum_{s=1}^d f_s(u) n_s, \quad u \in \mathbb{R}, \ \vec{n} = (n_1, \dots, n_d) \in B_1. \tag{3.16}$$

3. $H(u, v, \vec{n})$ is *conservative*:

$$H(u, v, \vec{n}) = -H(v, u, -\vec{n}), \quad u, \, v \in \mathbb{R}, \ \vec{n} \in B_1. \tag{3.17}$$

Using the conservativity (3.17) of H and the notation (2.11), we find that

$$\begin{aligned} & \sum_{K \in \mathcal{T}_h} \sum_{\Gamma \in \partial K} \int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n}) \, v|_\Gamma^{(L)} \, \mathrm{d}S \qquad (3.18) \\ = & \sum_{\Gamma \in \mathcal{F}_h^I} \int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n}) \left(v|_\Gamma^{(L)} - v|_\Gamma^{(R)} \right) \mathrm{d}S \\ & + \sum_{\Gamma \in \mathcal{F}_h^{DN}} \int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n}) \, v|_\Gamma^{(L)} \, \mathrm{d}S. \\ = & \sum_{\Gamma \in \mathcal{F}_h} \int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n}) \, [v] \, \mathrm{d}S \end{aligned}$$

Then, in virtue of (3.12) and (3.18), we obtain the approximation $b_h(u,v)$ of the convection form $\tilde{b}_h(u,v)$:

$$b_h(u,v) = \sum_{\Gamma\in\mathcal{F}_h}\int_\Gamma H(u|_\Gamma^{(L)}, u|_\Gamma^{(R)}, \vec{n})\,[v]\,\mathrm{d}S \tag{3.19}$$

$$- \sum_{K\in\mathcal{T}_h}\int_K \sum_{s=1}^{d} f_s(u)\frac{\partial v}{\partial x_s}\,\mathrm{d}x, \quad u,\, v\in H^1(\Omega,\mathcal{T}_h), \quad u\in L^\infty(\Omega).$$

By (3.8), (3.19) and (3.16),

$$b_h(u,v) = \tilde{b}_h(u,v) \quad \forall u\in H^2(\Omega),\ \forall v\in H^2(\Omega,\mathcal{T}_h). \tag{3.20}$$

Since $S_{hp}\subset H^2(\Omega,\mathcal{T}_h)$, the forms (3.7) – (3.10) make sense for $u := u_h,\ v := v_h \in S_{hp}$. We define the *semidiscrete approximate solution* as a function $u_h : Q_T \to \mathbb{R}$ satisfying the conditions

a) $$u_h \in C^1([0,T]; S_{hp}), \tag{3.21}$$

b) $$\left(\frac{\partial u_h(t)}{\partial t}, v_h\right) + a_h(u_h(t),v_h) + b_h(u_h(t),v_h) + \varepsilon J_h^\sigma(u_h(t),v_h) = \ell_h(v_h)\,(t)$$
$$\forall\, v_h\in S_{hp},\ \forall\, t\in[0,T],$$

c) $$(u_h(0), v_h) = (u^0, v_h) \quad \forall\, v_h\in S_{hp},$$

where $C^1([0,T];S_{hp})$ is the space of continuously differentiable functions on an interval $[0,T]$ with values in S_{hp}. The discrete problem (3.21), a) – c) is equivalent to an initial-problem for a system of ordinary differential equations (ODE). This approach to the numerical solution of initial-boundary value problems via the space semidiscretization is called the *method of lines*. If we apply some ODE solver to problem (3.21), a) – c), we obtain the fully discrete problem, see Section 3.4.

Taking into account that the exact regular solution satisfies $[u]_\Gamma = 0,\ \Gamma\in\mathcal{F}_h^I$, $u|_{\partial\Omega_D\times(0,T)} = u_D$ and using (3.6) and (3.20), we find that u satisfies the identity

$$\left(\frac{\partial u}{\partial t}(t), v_h\right) + a_h(u(t),v_h) + b_h(u(t),v_h) + \varepsilon J_h^\sigma(u(t),v_h) = \ell_h(v_h)\,(t) \tag{3.22}$$

for all $v_h\in S_{hp}$ and almost all $t\in(0,T)$, which implies the Galerkin orthogonality property of the error.

3.3 A priori error estimates

We present the error estimates of the method of lines (3.21) under the assumption that the exact solution u satisfies the condition

$$\frac{\partial u}{\partial t}\in L^2(0,T;H^s(\Omega)), \tag{3.23}$$

where an integer $s\geq 1$.

Then the main error estimates read

Theorem 2 *Let assumptions (3.2), a) – f), numerical flux assumptions* (H) *from Section 3.2 and the mesh assumptions* (A1) – (A2) *from Section 2.4.1 be satisfied. Let u be the exact strong solution of problem (3.1) satisfying (3.23) and let u_h be the approximate solution defined by (3.21). We assume that the constant C_W is chosen according to (2.39). Then the error $e_h = u_h - u$ satisfies the estimate*

$$\begin{aligned} &\max_{t\in[0,T]} \|e_h(t)\|^2_{L^2(\Omega)} + \varepsilon \int_0^T |||e_h(\vartheta)|||^2 \,\mathrm{d}\vartheta \qquad (3.24)\\ \leq\ & C_2 h^{2\mu-2} \left(|u|^2_{H^\mu(\Omega)} + |\partial u/\partial t|^2_{H^\mu(\Omega)} \right) \end{aligned}$$

where $\mu = \min(p+1, s)$ and $C_2 > 0$ is a constant independent of h.

We mention several remarks to the error estimate (3.24).

- The proof of Theorem 2 can be found in [DFS05]. We observe that estimate (3.24) is optimal in the H^1-seminorm and suboptimal in the L^2-norm, compare with Remark 2.
- It is possible derive the optimal L^2 estimates for the SIPG technique, see [DFKS08].
- On the other hand, numerical experiments show an optimal order of accuracy in the L^2-norm for odd degrees of polynomial approximations.
- The estimate (3.24) implies (similarly as the estimates (2.41) and (2.51)) that IPG methods have the order accuracy $O(h^p)$ provided that the exact solution is sufficiently regular (i.e., $s \geq p+1$). On the other hand if the exact solution is not sufficiently accurate ($s < p+1$) then the order of accuracy is given by the regularity of the solution ($O(h^{s-1})$) and not by the degree of polynomial approximation. This implies that it makes no sense to use a high degree of polynomial approximation, e.g., in the vicinity of shock waves where the solution is discontinuous. The following numerical experiments verify this observation.

3.4 Full space-time discretization

In this section, we deal with the time discretization of the system of ODEs (3.21), a) – c). It is possible to use the Runge-Kutta methods which are very popular for their simplicity and a high order of accuracy. Their drawback is a strong restriction to the size of the time step. To avoid this disadvantage it is suitable to use an implicit time discretization. However, a full implicit scheme leads to a necessity to solve a nonlinear system of algebraic equations at each time step which is rather expensive.

Therefore, we employ a semi-implicit method based on a higher order multi-step backward difference formula (BDF), which has favourable stability properties. In order to avoid the solution of a system of nonlinear algebraic equations at each time step we employ an explicit extrapolation method for the nonlinear terms.

k	$\alpha_v,\ v=k,k-1,\ldots,0$				$\beta_v,\ v=1,\ldots,k$		
1	1	-1			1		
2	$\frac{3}{2}$	-2	$\frac{1}{2}$		2	-1	
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$	3	-3	1

Table 3.1: Values of the coefficients $\alpha_v,\ v=0,\ldots,k$ and $\beta_v,\ v=1,\ldots,k$ for $k=1,2,3$

Let $k\geq 1$ and $\alpha_v,\ v=0,\ldots,k$ and $\beta_v,\ v=1,\ldots,k$ be real constants given by the expressions

$$\alpha_k \equiv \sum_{v=1}^{k}\frac{1}{v},\quad \alpha_v \equiv (-1)^{k-v}\binom{k}{v}\frac{1}{k-v},\ v=0,\ldots,k-1, \tag{3.25}$$

$$\beta_v \equiv (-1)^{v+1}\binom{k}{v} = -\alpha_{k-v}v,\ v=1,\ldots,k. \tag{3.26}$$

Table 3.1 shows the values of $\alpha_v,\ v=0,\ldots,k$ and $\beta_v,\ v=1,\ldots,k$ for $k=1,2,3$.

Let $t_s=s\tau,\ s=0,1,\ldots,r$, be a uniform partition of the time interval $[0,T]$ with a time step $\tau=T/r$, where $r>k$ is an integer. For simplicity, we put

$$A_h(u,v)\equiv a_h(u,v)+\varepsilon J_h^{\sigma}(u,v),\quad u,v\in H^2(\Omega,\mathcal{T}_h).$$

Definition 3 *We define the* approximate solution *of problem (3.3) a) – c) obtained by a* k-step BDF-DGFE *method ($k\geq 1$) as the functions u_h^{s+k}, $t_{s+k}\in[0,T]$, satisfying the conditions*

$$\begin{aligned}
&a) \quad u_h^{s+k}\in S_h, \qquad (3.27)\\
&b) \quad \frac{1}{\tau}\left(\sum_{v=0}^{k}\alpha_v u_h^{s+v},w_h\right)+A_h(u_h^{s+k},w_h)+b_h(E^{s+k}(u_h),w_h)\\
&\qquad\qquad = \ell_h(w_h)\,(t_{s+k}) \qquad \forall\, w_h\in S_h,\ s=0,1,2,\ldots,r-k,\\
&\qquad \textit{where}\ \ E^{s+k}(u_h)=\sum_{v=1}^{k}\beta_v u_h^{s+k-v}\\
&c) \quad u_h^1,\ldots,u_h^{k-1}\in S_h \ \ \textit{are given from previous time steps},\\
&d) \quad (u_h^0,w_h)=(u^0,w_h)\quad \forall\, w_h\in S_h,
\end{aligned}$$

where $\alpha_v,\ v=0,\ldots,k$ and $\beta_v,\ v=1,\ldots,k$ are given by (3.25) and (3.26), respectively. The function u_h^s is called the approximate solution *at time t_s.*

Remark 3

i) We see that the higher order explicit extrapolation $E^{s+k}(u_h)$ *depends on $u_h^s,\ldots,u_h^{s+k-1}$ and is independent of u_h^{s+k}.*

ii) Since (3.27) , a)-d) represents a k-step formula, we have to define the solution $u_h^1, \ldots, u_h^{k-1}$ *at times* $t_1, \ldots, t_{k-1}$*. This can be done, e.g., by a one step formula or a* k^{th}*-order Runge-Kutta scheme.*

iii) The approximate solution u_h^0 *at* $t = 0$ *given by (3.27), d) is the* L^2*-projection of the function* u^0 *from the initial condition (3.1), d) into* S_h*.*

The discrete problem (3.27), a) – d) is equivalent to a system of linear algebraic equations for each $t_{s+k} \in [0, T]$.

Numerical analysis

This method was analysed in [DFH07] for $k = 1$ and in [DV08] for $k = 2, 3$. We assume that the weak solution u is sufficiently regular, namely,

$$u \in W^{1,\infty}(0, T; H^{p+1}(\Omega)) \cap W^{k,\infty}(0, T; H^1(\Omega)) \cap W^{k+1,\infty}(0, T; L^2(\Omega)), \quad (3.28)$$

where $p \in IN$ is a degree of a polynomial approximation with respect to the space coordinates and $k \in IN$ is the degree of a multi-step formula for the time discretization.

We set

$$\begin{aligned} e_h^s &\equiv u_h^s - u(t_s), \quad s = 0, 1, \ldots, r, \\ \|e\|^2_{h,\tau,L^\infty(L^2)} &\equiv \max_{s=0,\ldots,r} \|e_h^s\|^2_{L^2(\Omega)}, \\ \|e\|^2_{h,\tau,L^2(H^1)} &\equiv \tau\varepsilon \sum_{s=0}^{r} |||e_h^s|||^2. \end{aligned} \quad (3.29)$$

Then, using SIPG method to the space semi-discretization, we are able to derive the estimate

$$\|e\|^2_{h,\tau,L^\infty(L^2)} + h^2\|e\|^2_{h,\tau,L^2(H^1)} \le \tilde{C}_2 \left((h^{2p+2} + \tau^{2k})(1 + 1/\varepsilon) + \sum_{j=0}^{k-1} \|e_h^j\|^2_{L^2(\Omega)} \right). \quad (3.30)$$

3.5 Numerical examples

In this section we shall verify the theoretical error estimates (3.30) derived in the previous section. We try to investigate the dependence of the computational error on h and τ independently. Based on (3.30) we expect the error dependence in L^2-norm and in the H^1-seminorm according to the formula

$$e_{h,\tau} \approx c_h h^{p+1} + c_\tau \tau^k \quad \text{and} \quad e_{h,\tau} \approx c_h h^p + c_\tau \tau^k, \quad (3.31)$$

respectively, where h and τ are the space and time steps, p and k are the degrees of the space and time discretization, c_h and c_τ are constants and $e_{h,\tau}$ is the corresponding computational error.

In both cases we solve the 2D viscous Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x_1} + u\frac{\partial u}{\partial x_2} = \varepsilon\Delta u + g \quad \text{in } \Omega \times (0,T), \tag{3.32}$$

equation (3.1), a) with $\Omega = (0,1)^2$, $\partial\Omega = \partial\Omega_D$, $f_s(u) = u^2/2$, $s = 1,2$, the boundary condition (3.1), b) and the initial condition (3.1), d).

In the form b_h we use the numerical flux

$$H(u_1,u_2,\vec{n}) = \begin{cases} \sum_{s=1}^{2} f_s(u_1)n_s & \text{if } Df > 0 \\ \sum_{s=1}^{2} f_s(u_2)n_s & \text{if } Df \le 0 \end{cases}, \tag{3.33}$$

where $Df = \sum_{s=1}^{2} \frac{\partial f_s}{\partial u}\left((u_1+u_2)/2\right) n_s$, $\vec{n} = (n_1,n_2)$.

3.5.1 Convergence with respect to τ

In this case we put $\varepsilon = 0.01$, $T = 1$ and the functions u_D, u_0 and g are chosen in such a way that the exact solution has the form

$$u(x_1,x_2,t) = 16\,\frac{e^{10t}-1}{e^{10}-1}\,x_1(1-x_1)x_2(1-x_2). \tag{3.34}$$

The computations were carried out on a fine triangular mesh having 4219 elements with a piecewise cubic approximation in space and using 6 different time steps: 1/20, 1/40, 1/80, 1/160, 1/320, 1/640. For such data setting we expect that $c_h h^{p+1} \ll c_\tau \tau^k$. Fig. 3.1 shows the computational errors at $t = T$ and the corresponding orders of convergence with respect to τ in the L^2–norm and the H^1-seminorm for the k-step BDF scheme (3.27), a) – d) with $k = 1$, $k = 2$ and $k = 3$. The expected order of convergence $O(\tau^k)$ is observed in each case. A small decrease of the order of convergence in the H^1-seminorm for $k = 3$ and $\tau = 1/640$ is caused by the influence of the spatial discretization since in this case the statement $c_h h^{p+1} \ll c_\tau \tau^k$ is no longer valid.

3.5.2 Convergence with respect to h

In this case we put $\varepsilon = 0.1$, $T = 10$ and the functions u_D, u_0 and g are chosen in such a way that the exact solution has the form

$$u(x_1,x_2,t) = (1-e^{-10t})(x_1^2+x_2^2)x_1x_2(1-x_1)(1-x_2). \tag{3.35}$$

The computations were carried out with the 3-step BDF (3.27), a) – d) on 7 triangular meshes having 128, 288, 512, 1152, 2048, 4608 and 8192 elements. For such data setting we expect that $c_h h^{p+1} \gg c_\tau \tau^k$. Fig. 3.2 shows the computational errors at $t = T$ and the corresponding orders of convergence with respect to h in the L^2–norm and the H^1-seminorm for piecewise linear P_1, quadratic P_2 and cubic P_3 approximations. We observe the order of convergence $O(h^{p+1})$ for $p = 1,2,3$ in the L^2–norm and $O(h^p)$ in the H^1-seminorm, which perfectly corresponds with the theoretical results (3.30).

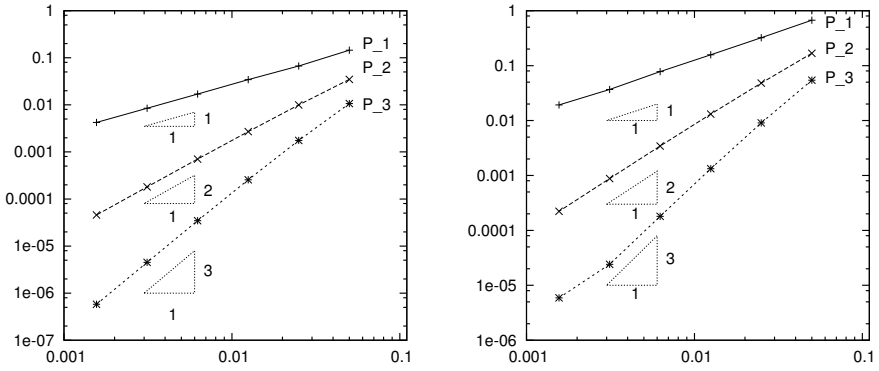


Figure 3.1: Computational errors and orders of convergence in the L^2–norm (left) and the H^1-seminorm (right) for scheme (3.27), a) – d) with $k = 1$ (full line), $k = 2$ (dashed line) and $k = 3$ (dotted line).

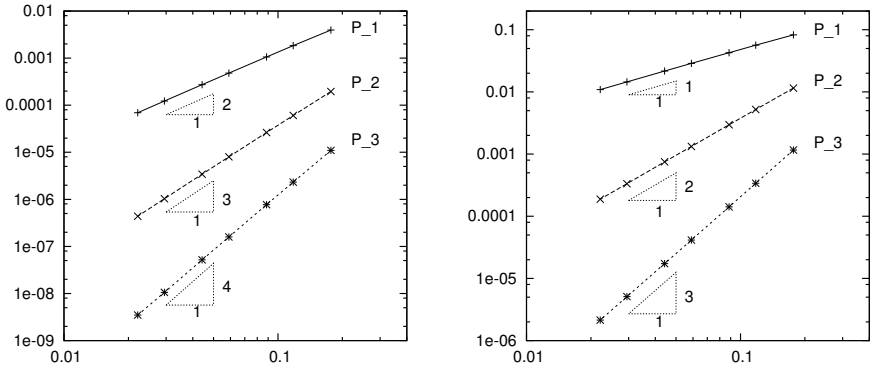


Figure 3.2: Computational errors and orders of convergence in the L^2–norm (left) and the H^1-seminorm (right) for scheme (3.27), a) – d) with P_1 (full line), P_2 (dashed line) and P_3 (dotted line) approximations.

Chapter 4

The Navier-Stokes equations

4.1 Governing equations

Within this section we extend the application of the DGFEM to the the system of the compressible Navier-Stokes equations. Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, be a bounded domain with a Lipschitz piecewise polynomial boundary and $T > 0$. We set $Q_T = \Omega \times (0, T)$ and by $\partial\Omega$ denote the boundary of Ω which consists of several disjoint parts. We distinguish inlet $\partial\Omega_i$, outlet $\partial\Omega_o$ and impermeable walls $\partial\Omega_w$, i.e. $\partial\Omega = \partial\Omega_i \cup \partial\Omega_o \cup \partial\Omega_w$. The system of the Navier-Stokes equations describing a motion of non-stationary viscous compressible flow can be written in the dimensionless form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = \sum_{s=1}^{d} \frac{\partial}{\partial x_s} \left(\sum_{k=1}^{d} \mathbb{K}_{sk}(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_k} \right) \quad \text{in } Q_T, \tag{4.1}$$

where

$$\boldsymbol{w} = \boldsymbol{w}(x, t) : Q_T \to \mathbb{R}^{d+2},\ \boldsymbol{w} = (\rho,\, \rho v_1, \ldots, \rho v_d,\, e)^{\mathrm{T}} \tag{4.2}$$

is the state vector,

$$\boldsymbol{f}_s : \mathbb{R}^{d+2} \to \mathbb{R}^{d+2},\ s = 1, \ldots, d,\ \boldsymbol{f}_s = (\rho v_s,\, \rho v_s v_1 + \delta_{s1} p,\, \ldots,\, \rho v_s v_d + \delta_d p,\, (e+p)\, v_s)^{\mathrm{T}}, \tag{4.3}$$

are the inviscid fluxes, and

$$\mathbb{K}_{sk} : \mathbb{R}^{d+2} \to \mathbb{R}^{(d+2)\times(d+2)},\ s, k = 1, \ldots, d, \tag{4.4}$$

represents the viscous terms. The forms of matrices $\mathbb{K}_{sk},\ s = 1, \ldots, d$, can be found, e.g., in [Dol08] or [FFS03, Section 4.3]. Let us note that the viscous terms of the right-hand side of (4.1) are usually written in the form

$$\sum_{s=1}^{d} \frac{\partial}{\partial x_s} \boldsymbol{R}_s(\boldsymbol{w}, \nabla \boldsymbol{w}) \tag{4.5}$$

where

$$\begin{aligned}\boldsymbol{R}_s(\boldsymbol{w},\nabla\boldsymbol{w}) &= (R_s^{(1)}(\boldsymbol{w},\nabla\boldsymbol{w}),\ldots,R_s^{(d+2)}(\boldsymbol{w},\nabla\boldsymbol{w}))^{\mathrm{T}}\\ &= \left(0,\,\tau_{s1},\,\ldots,\,\tau_{sd},\sum_{k=1}^{d}\tau_{sk}v_k+\frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial\theta}{\partial x_s}\right)^{\mathrm{T}}.\end{aligned}$$

Finally, vectors $\boldsymbol{R}_s$ and matrices $\mathbb{K}_{s,k}$ satisfy

$$\boldsymbol{R}_s(\boldsymbol{w},\nabla\boldsymbol{w})=\sum_{k=1}^{d}\mathbb{K}_{s,k}(\boldsymbol{w})\frac{\partial\boldsymbol{w}}{\partial x_k},\quad s=1,\ldots,d. \tag{4.6}$$

We consider the Newtonian type of fluid, i. e., the viscous part of the stress tensor has the form

$$\tau_{sk}=\frac{1}{\mathrm{Re}}\left[\left(\frac{\partial v_s}{\partial x_k}+\frac{\partial v_k}{\partial x_s}\right)-\frac{2}{3}\sum_{i=1}^{d}\frac{\partial v_i}{\partial x_i}\,\delta_{sk}\right], \tag{4.7}$$

$s,k=1,\ldots,d$. We use the following notation: ρ – density, p – pressure, e – total energy, $\boldsymbol{v}=(v_1,\ldots,v_d)$ – velocity, θ – temperature, γ – Poisson adiabatic constant, Re – Reynolds number. Pr – Prandtl number. In order to close the system, we consider the state equation for perfect gas and the definition of the total energy

$$p=(\gamma-1)\,(e-\rho|\boldsymbol{v}|^2/2),\quad e=c_V\rho\theta+\rho|\boldsymbol{v}|^2/2, \tag{4.8}$$

where c_V is the specific heat at constant volume which we assume to be equal to one in the dimensionless case.

The system (1.1) is of *hyperbolic-parabolic* type and it is equipped with the initial condition

$$\boldsymbol{w}(x,0)=\boldsymbol{w}^0(x),\quad x\in\Omega, \tag{4.9}$$

and the following set of boundary conditions on appropriate parts of boundary:

$$\begin{aligned}&a)\quad \rho=\rho_D,\quad \boldsymbol{v}=\boldsymbol{v}_D,\\ &\qquad \sum_{k=1}^{d}\left(\sum_{l=1}^{d}\tau_{lk}n_l\right)v_k+\frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\frac{\partial\theta}{\partial\vec{n}}=0\quad \text{on } \partial\Omega_i,\\ &b)\quad \sum_{k=1}^{d}\tau_{sk}n_k=0,\ s=1,\ldots,d,\quad \frac{\partial\theta}{\partial\vec{n}}=0\quad \text{on } \partial\Omega_o,\\ &c)\quad \boldsymbol{v}=0,\ \frac{\partial\theta}{\partial\vec{n}}=0\quad \text{on } \partial\Omega_w,\end{aligned} \tag{4.10}$$

where ρ_D and $\boldsymbol{v}_D$ are given function and $\vec{n}=(n_1,\ldots,n_d)$ is a unit outer normal to $\partial\Omega$. Another possibility is to replace the adiabatic boundary condition (4.10), c) by

$$c')\ \boldsymbol{v}=0,\quad \theta=\theta_D\quad \text{on } \partial\Omega_w. \tag{4.11}$$

The problem to solve the Navier-Stokes equations (1.1) with constitutive relations (4.7) – (4.8), equipped with the initial and boundary conditions (4.9) – (4.11) will be denoted by (CFP) (compressible flow problem).

If we omit the time derivative term on the left-hand side of (1.1), we obtain the *stationary Navier-Stokes equations*. The problem to solve the stationary Navier-Stokes equations equipped with the same boundary conditions as in the non-stationary case will be denoted by (sCFP) (stationary compressible flow problem).

Let us mention that the Euler fluxes $\boldsymbol{f}_s,\ s = 1, \ldots, d$, satisfy (see [FFS03, Lemma 3.1])

$$\boldsymbol{f}_s(\boldsymbol{w}) = \boldsymbol{A}_s(\boldsymbol{w})\boldsymbol{w}, \quad s = 1, \ldots, d, \tag{4.12}$$

where

$$\boldsymbol{A}_s(\boldsymbol{w}) = \frac{D\boldsymbol{f}_s(\boldsymbol{w})}{D\boldsymbol{w}},\ s = 1, \ldots, d, \tag{4.13}$$

are the Jacobi matrices of the mappings $\boldsymbol{f}_s$. Finally, we define the matrix

$$\boldsymbol{P}(\boldsymbol{w}, \vec{n}) = \sum_{s=1}^{d} \boldsymbol{A}_s(\boldsymbol{w}) n_s, \tag{4.14}$$

where $\vec{n} = (n_1, \ldots, n_d) \in \mathbb{R}^d,\ |n|^2 = \sum_{l=1}^{d} n_l^2 = 1$, which plays a role in the definition of a numerical flux.

4.2 Discretization

4.2.1 Triangulations

Let $\mathcal{T}_h$ $(h > 0)$ be a partition of the closure $\overline{\Omega}$ of the domain Ω into a finite number of closed d-dimensional elements K with mutually disjoint interiors. I.e., $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} K$. Moreover, let $F_K : \hat{K} \to \mathbb{R}^d$ be a polynomial mapping such that $F_K(\hat{K}) = K$ where

$$\hat{K} = \{(\hat{x}_1, \ldots, \hat{x}_d);\ \hat{x}_i \geq 0,\ i = 1, \ldots, d,\ \sum_{i=1}^{d} \hat{x}_i \leq 1\} \tag{4.15}$$

is a reference simplex.

It is natural to assume that if $K \cap \partial\Omega = \emptyset$ or $K \cap \partial\Omega$ is a straight line then F_K is an affine mapping and then K is a simplex. Otherwise, F_K is a polynomial mapping of the same degree as the segment $K \cap \partial\Omega$ and then K is a curved simplex. We call $\mathcal{T}_h = \{K\}_{K \in \mathcal{T}_h}$ a *triangulation* of Ω and do not require the conforming properties from the finite element method, see, e.g, [Cia79].

By $\mathcal{F}_h$ we denote the set of all open $(d-1)$-dimensional faces (open edges when $d = 2$ or open faces when $d = 3$) of all elements $K \in \mathcal{T}_h$. Further, the symbol $\mathcal{F}_h^I$ stands for the set of all $\Gamma \in \mathcal{F}_h$ that are contained in Ω (inner faces). Moreover, we introduce notations $\mathcal{F}_h^w$, $\mathcal{F}_h^i$ and $\mathcal{F}_h^o$ for the sets of all $\Gamma \in \mathcal{F}_h$ such that $\Gamma \subset \partial\Omega_w$, $\Gamma \subset \partial\Omega_i$ and $\Gamma \subset \partial\Omega_o$, respectively. Furthermore, we denote by $\mathcal{F}_h^D$ the set of all $\Gamma \in \mathcal{F}_h$ where the Dirichlet type of boundary conditions is prescribed at least for one component of $\boldsymbol{w}$ (i.e., $\mathcal{F}_h^D = \mathcal{F}_h^w \cup \mathcal{F}_h^i$) and by $\mathcal{F}_h^N$ the set of all $\Gamma \in \mathcal{F}_h$ where only the Neumann boundary conditions are prescribed (i.e., $\mathcal{F}_h^N = \mathcal{F}_h^o$). Obviously, $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^D \cup \mathcal{F}_h^N$. For a shorter notation we put $\mathcal{F}_h^{io} = \mathcal{F}_h^i \cup \mathcal{F}_h^o$, $\mathcal{F}_h^{ID} = \mathcal{F}_h^I \cup \mathcal{F}_h^D$ and $\mathcal{F}_h^{DN} = \mathcal{F}_h^D \cup \mathcal{F}_h^N = \mathcal{F}_h^w \cup \mathcal{F}_h^i \cup \mathcal{F}_h^o$.

Finally, for each $\Gamma \in \mathcal{F}_h$ we define a unit normal vector $\vec{n}_\Gamma$. We assume that for $\Gamma \in \mathcal{F}_h^{DN}$ the vector $\vec{n}_\Gamma$ has the same orientation as the outer normal of $\partial\Omega$. For $\vec{n}_\Gamma,\ \Gamma \in \mathcal{F}_h^I$, the orientation is arbitrary but fixed for each face.

4.2.2 DG FE spaces

In contrary to Chapters 2 and 3, we admit a use of different polynomial degrees of freedom on different elements $K \in \mathcal{T}_h$. Therefore, we assign a positive integer p_K (local polynomial degree) to each $K \in \mathcal{T}_h$ and define the vector

$$\mathsf{p} = \{p_K, K \in \mathcal{T}_h\}. \tag{4.16}$$

Over the triangulation $\mathcal{T}_h$ we define the space of discontinuous piecewise polynomial functions associated with the vector p by

$$S_{h\mathsf{p}} = \{v;\ v \in L^2(\Omega),\ v|_K \in P_{p_K}(K)\ \forall K \in \mathcal{T}_h\}, \tag{4.17}$$

where $P_{p_K}(K)$ denotes the space of all polynomials on K of degree $\le p_K,\ K \in \mathcal{T}_h$. Since $\boldsymbol{w}(x,t) \in \mathbb{R}^{d+2},\ x \in \Omega,\ t \in (0,T)$ from (4.1) is a vector function, we seek the approximate solution in the space of vector-valued functions

$$\boldsymbol{S}_{h\mathsf{p}} = \underbrace{S_{h\mathsf{p}} \times \ldots \times S_{h\mathsf{p}}}_{d+2 \text{ times}}. \tag{4.18}$$

Similarly, we put

$$\boldsymbol{H}^s(\Omega, \mathcal{T}_h) = \underbrace{H^s(\Omega, \mathcal{T}_h) \times \ldots \times H^s(\Omega, \mathcal{T}_h)}_{d+2 \text{ times}}, \tag{4.19}$$

where $H^s(\Omega, \mathcal{T}_h),\ s \ge 1$ is the broken Sobolev space defined by (2.9).

4.2.3 IPG formulation

The crucial item of the DGFE formulation of (4.1) is the treatment of the viscous terms. Let $\boldsymbol{w}$ be a sufficiently regular solution of (4.1), then multiplying the viscous term on the right-hand side of (4.1) by $\boldsymbol{\varphi} \in \boldsymbol{H}^2(\Omega, \mathcal{T}_h)$, integrating over $K \in \mathcal{T}_h$, summing over all $K \in \mathcal{T}_h$ and using notation (2.11), we obtain

$$\begin{aligned} &- \sum_{K \in \mathcal{T}_h} \int_K \sum_{s=1}^{d} \left(\sum_{k=1}^{d} \mathbb{K}_{s,k}(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_k} \right) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x_s} \,\mathrm{d}x \\ &+ \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sum_{s=1}^{d} \left\langle \left(\sum_{k=1}^{d} \mathbb{K}_{s,k}(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_k} \right) \right\rangle n_s \cdot [\boldsymbol{\varphi}] \,\mathrm{d}S. \end{aligned} \tag{4.20}$$

Using the strategy from Section 2.3 we add to this expression a *stabilization term* which we obtain by the formal exchange of arguments $\boldsymbol{w}$ and $\boldsymbol{\varphi}$ in the last term of (4.20) (compare with (2.22)), i.e.,

$$\theta \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sum_{s=1}^{d} \left\langle \sum_{k=1}^{d} \mathbb{K}_{s,k}(\boldsymbol{w}) \frac{\partial \boldsymbol{\varphi}}{\partial x_k} \right\rangle n_s \cdot [\boldsymbol{w}] \,\mathrm{d}S, \tag{4.21}$$

where $\theta = 1, 0, -1$ (depending of the treated IPG variant). However, numerical experiments indicate that this choice of stabilization is not suitable. It is caused by that fact that for $\boldsymbol{\varphi} = (1, 0, \ldots, 0)^{\mathrm{T}} \in I\!R^{d+2}$ we obtain a non-vanishing term (4.21) whereas both terms in (4.20) are equal to zero since the first rows of $K_{s,k}, s, k = 1, \ldots, d$ vanish, see [Dol08] for more details. Therefore, we use the approach from [BO99b], [HH06a], [HH06b], where the stabilization term

$$\theta \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sum_{s=1}^{d} \left\langle \sum_{k=1}^{d} I\!K_{s,k}^{\mathrm{T}}(\boldsymbol{w}) \frac{\partial \boldsymbol{\varphi}}{\partial x_k} \right\rangle n_s \cdot [\boldsymbol{w}] \, \mathrm{d}S. \tag{4.22}$$

was employed which avoids the drawback mentioned above. Here, $I\!K_{s,k}^{\mathrm{T}},\ s, k = 1, \ldots, d$ denotes the matrix transposed to $I\!K_{s,k},\ s, k = 1, \ldots, d$.

4.2.4 Viscous terms

In virtue of (2.24), (4.21) and (4.22), for $\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathsf{p}}$, we define the forms

$$\begin{aligned}
\boldsymbol{a}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= \sum_{K \in \mathcal{T}_h} \int_K \sum_{s,k=1}^{d} \left(I\!K_{s,k}(\bar{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{w}_h}{\partial x_k} \right) \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \, \boldsymbol{d}x \\
&\quad - \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sum_{s=1}^{d} \left\langle \sum_{k=1}^{d} I\!K_{s,k}(\bar{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{w}_h}{\partial x_k} \right\rangle n_s \cdot [\boldsymbol{\varphi}_h] \, \boldsymbol{d}S \\
&\quad - \theta \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sum_{s=1}^{d} \left\langle \sum_{k=1}^{d} I\!K_{s,k}^{\mathrm{T}}(\bar{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{\varphi}_h}{\partial x_k} \right\rangle n_s \cdot [\boldsymbol{w}_h] \, \boldsymbol{d}S, \\
\tilde{\boldsymbol{a}}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{\varphi}_h) &= -\theta \sum_{\Gamma \in \mathcal{F}_h^{D}} \int_\Gamma \sum_{s,k=1}^{d} I\!K_{s,k}^{\mathrm{T}}(\bar{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{\varphi}_h}{\partial x_k} \, n_s \cdot \boldsymbol{w}_B \, \boldsymbol{d}S,
\end{aligned} \tag{4.23}$$

The state vector $\boldsymbol{w}_B$ prescribed on $\partial\Omega_i \cup \partial\Omega_w$ is given by the boundary conditions, in particular, for the case (4.10) a)–c) we have

$$\begin{aligned}
\boldsymbol{w}_B &= (\rho|_\Gamma, 0, \ldots, 0, \rho|_\Gamma \theta|_\Gamma) \text{ for } \Gamma \in \mathcal{F}_h^w, \\
\boldsymbol{w}_B &= (\rho_D, \rho_D(\boldsymbol{v}_D)_1, \ldots, \rho_D(\boldsymbol{v}_D)_d, \rho|_D \theta|_\Gamma + \frac{1}{2}\rho_D |\boldsymbol{v}_D|^2)
\end{aligned} \tag{4.24}$$

for $\Gamma \in \mathcal{F}_h^i$, and for the case (4.10) a)–b), (4.11) c')

$$\begin{aligned}
\boldsymbol{w}_B &= (\rho|_\Gamma, 0, \ldots, 0, \rho|_\Gamma \theta_D) \text{ for } \Gamma \in \mathcal{F}_h^w, \\
\boldsymbol{w}_B &= (\rho_D, \rho_D(\boldsymbol{v}_D)_1, \ldots, \rho_D(\boldsymbol{v}_D)_d, \rho|_\Gamma \theta|_\Gamma + \frac{1}{2}\rho_D |\boldsymbol{v}_D|^2)
\end{aligned} \tag{4.25}$$

for $\Gamma \in \mathcal{F}_h^i$, where $\rho_D, \boldsymbol{v}_D$ and θ_D are given functions from the boundary conditions (4.10)–(4.11) and $\rho|_\Gamma$ and $\theta|_\Gamma$ are the values of density and temperature extrapolated from interior of Ω on the appropriate boundary part, respectively. For more details, see [Dol04], [Dol08].

The value of θ appearing in (4.23) can be chosen arbitrarily but the values $-1, 0$ and 1 are the most usual. Then we obtain three variants of the DGFE scheme:
$\theta = 1$ – *symmetric interior penalty Galerkin* (SIPG),
$\theta = -1$ – *non-symmetric interior penalty Galerkin* (NIPG),
$\theta = 0$ – *incomplete interior penalty Galerkin* (IIPG).
These variants applied to the Poisson equation were analysed in [ABCM02]. A numerical study of these variants applied to the Navier-Stokes equations was given in [Dol08]. Within this paper, we uniquely employ the NIPG variant of the DGFE method. We expect the same qualitative behaviour of the IIPG and SIPG variants.

4.2.5 Interior and boundary penalties

Similarly as in (2.20) and (2.21), for $\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathtt{p}}$, we define the forms

$$\boldsymbol{J}_h^\sigma(\boldsymbol{w}, \boldsymbol{\varphi}) = \sum_{\Gamma \in \mathcal{F}_h^{ID}} \int_\Gamma \sigma [\boldsymbol{w}] \cdot [\boldsymbol{\varphi}] \, \boldsymbol{d}S, \qquad \tilde{\boldsymbol{J}}_h^\sigma(\boldsymbol{\varphi}) = \sum_{\Gamma \in \mathcal{F}_h^{D}} \int_\Gamma \sigma \boldsymbol{w}_B \cdot \boldsymbol{\varphi} \, \boldsymbol{d}S, \tag{4.26}$$

where $\boldsymbol{w}_B$ is the boundary state vector given either by (4.24) or (4.25) and the penalty parameter σ is chosen by

$$\sigma|_\Gamma = \frac{C_W}{\operatorname{diam}(\Gamma)\,\mathrm{Re}}, \quad \Gamma \in \mathcal{F}_h^{ID}, \tag{4.27}$$

where Re is the Reynolds number of the flow and $C_W > 0$ is a suitable constant whose choice depends on the used variant of the DGFE method (NIPG, IIPG or SIPG) and the degree of polynomial approximation, see [Dol08], where a numerical study was presented. For numerical experiments presented within this paper, we uniquely set $C_W = 10$.

4.2.6 Inviscid terms

Concerning the inviscid term, we use an approximation via numerical flux as in (3.19) very well known from *finite volume method*, see, [Fei93] and [FFS03]. We use the Vijayasundaram numerical flux [Vij86] for the approximation of inviscid fluxes trough faces $\Gamma \in \mathcal{F}_h$, which is suitable for the semi-implicit time discretization. Hence, for $\boldsymbol{w}_h, \bar{\boldsymbol{w}}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathtt{p}}$, we define the forms

$$\begin{aligned}
\boldsymbol{b}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \;:=\; & - \sum_{K \in \mathcal{T}_h} \int_K \sum_{s=1}^{d} \boldsymbol{A}_s(\bar{\boldsymbol{w}}_h) \boldsymbol{w}_h \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \, \boldsymbol{d}x \\
& + \sum_{\Gamma \in \mathcal{F}_h^I} \int_\Gamma \left(\boldsymbol{P}^+ \left(\langle \bar{\boldsymbol{w}}_h \rangle, \vec{n} \right) \boldsymbol{w}_h|_\Gamma^{(p)} + \boldsymbol{P}^- \left(\langle \bar{\boldsymbol{w}}_h \rangle, \vec{n} \right) \boldsymbol{w}_h|_\Gamma^{(n)} \right) \cdot [\boldsymbol{\varphi}_h] \boldsymbol{d}S \\
& + \sum_{\Gamma \in \mathcal{F}_h^{io}} \int_\Gamma \left(\boldsymbol{P}^+ \left(\langle \bar{\boldsymbol{w}}_h \rangle, \vec{n} \right) \boldsymbol{w}_h|_\Gamma^{(p)} \right) \cdot [\boldsymbol{\varphi}_h] \, \boldsymbol{d}S \\
& + \sum_{\Gamma \in \mathcal{F}_h^{w}} \int_\Gamma \boldsymbol{F}_W(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \vec{n}) \cdot \boldsymbol{\varphi}_h \, \boldsymbol{d}S,
\end{aligned} \tag{4.28}$$

$$\tilde{\boldsymbol{b}}_h(\bar{\boldsymbol{w}}_h,\boldsymbol{\varphi}_h) \;:=\; -\sum_{\Gamma\in\mathcal{F}_h^{io}}\int_\Gamma \left(\boldsymbol{P}^-\left(\langle\bar{\boldsymbol{w}}_h\rangle,\vec{n}\right)\bar{\boldsymbol{w}}_h|_\Gamma^{(n)}\right)\cdot[\boldsymbol{\varphi}_h]\,\boldsymbol{d}S,$$

where $\boldsymbol{A}_s(\cdot)$ are the Jacobi matrices of the mappings $\boldsymbol{f}_s$, $s=1,\ldots,d$, $\boldsymbol{P}^{\pm}(\cdot,\cdot)$ are the positive and negative parts of the matrix $\boldsymbol{P}(\cdot,\cdot)$ given by (4.14). Moreover,

$$\boldsymbol{F}_W(\bar{\boldsymbol{w}}_h,\boldsymbol{w}_h,\vec{n}) = (\gamma-1)D\boldsymbol{F}_W(\bar{\boldsymbol{w}}_h,\vec{n})\boldsymbol{w}_h, \tag{4.29}$$

where $D\boldsymbol{F}_W(\boldsymbol{w},\vec{n})$ is a $(d+2)\times(d+2)$ matrix obtained by the differentiation of $\sum_{s=1}^d \boldsymbol{f}_s(\boldsymbol{w})n_s$ with respect to $\boldsymbol{w}$,

$$D\boldsymbol{F}_W(\boldsymbol{w},\vec{n}) = \begin{pmatrix} 0 & 0 & \ldots & 0 & 0\\ |v|^2n_1/2 & -v_1n_1 & \ldots & -v_dn_1 & n_1\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ |v|^2n_d/2 & -v_1n_d & \ldots & -v_dn_d & n_d\\ 0 & 0 & \ldots & 0 & 0 \end{pmatrix}. \tag{4.30}$$

Here $\vec{n}=(n_1,\ldots,n_d)$ and $(v_1,\ldots,v_d)$ are components of velocity vector.

Finally,

$$\bar{\boldsymbol{w}}|_\Gamma^{(n)} = LRP(\bar{\boldsymbol{w}}|_\Gamma^{(p)},\boldsymbol{w}_D,\vec{n}_\Gamma),\ \Gamma\in\mathcal{F}_h^{io} \tag{4.31}$$

where $LRP(\cdot,\cdot,\cdot)$ represents a solution of the *local Riemann problem* considered on edge $\Gamma\in\mathcal{F}_h^{io}$ and $\boldsymbol{w}_D$ is a given state vector (e.g. from far-field boundary conditions), see [Dol06]. For more details, we refer to [DF04].

4.2.7 Semi-implicit BDF-DGFE discretization

In order to simplify the notation, for $\bar{\boldsymbol{w}}_h$, $\boldsymbol{w}_h$, $\boldsymbol{\varphi}_h\in\boldsymbol{S}_{h\mathsf{p}}$, we put

$$\begin{aligned} \boldsymbol{c}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{w}_h,\boldsymbol{\varphi}_h\right) &= \boldsymbol{a}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{w}_h,\boldsymbol{\varphi}_h\right)+\boldsymbol{b}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{w}_h,\boldsymbol{\varphi}_h\right)+\boldsymbol{J}_h^\sigma\left(\boldsymbol{w}_h,\boldsymbol{\varphi}_h\right),\\ \tilde{\boldsymbol{c}}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{\varphi}_h\right) &= \tilde{\boldsymbol{a}}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{\varphi}_h\right)+\tilde{\boldsymbol{b}}_h\left(\bar{\boldsymbol{w}}_h,\boldsymbol{\varphi}_h\right)+\tilde{\boldsymbol{J}}_h^\sigma\left(\boldsymbol{\varphi}_h\right). \end{aligned} \tag{4.32}$$

The forms $\boldsymbol{c}_h$ and $\tilde{\boldsymbol{c}}_h$ make sense not only for piecewise polynomial functions but also for functions from $\boldsymbol{H}^2(\Omega,\mathcal{T}_h)$.

It is possible to show (see, e.g., [Dol04], [Dol08]) that if $\boldsymbol{w}:\Omega\times(0,T)\to I\!R^{d+2}$ is a sufficiently regular function satisfying the Navier-Stokes equations (1.1) with constitutive relations (4.7) – (4.8) and the corresponding initial and boundary conditions, (4.9) – (4.11) then

$$\frac{\boldsymbol{d}}{\boldsymbol{d}t}\left(\boldsymbol{w},\boldsymbol{\varphi}\right)+\boldsymbol{c}_h\left(\boldsymbol{w},\boldsymbol{w},\boldsymbol{\varphi}\right)=\tilde{\boldsymbol{c}}_h\left(\boldsymbol{w},\boldsymbol{\varphi}\right)\quad\forall\boldsymbol{\varphi}\in\boldsymbol{S}_{h\mathsf{p}}, \tag{4.33}$$

where $(\cdot,\cdot)$ denotes L^2-scalar product over Ω.

Similarly, if $\boldsymbol{w}:\Omega\to I\!R^{d+2}$ is a sufficiently regular function satisfying the stationary Navier-Stokes equations and the corresponding boundary conditions, then

$$\boldsymbol{c}_h\left(\boldsymbol{w},\boldsymbol{w},\boldsymbol{\varphi}\right)=\tilde{\boldsymbol{c}}_h\left(\boldsymbol{w},\boldsymbol{\varphi}\right)\quad\forall\boldsymbol{\varphi}\in\boldsymbol{S}_{h\mathsf{p}}. \tag{4.34}$$

Now, we introduce the space semi-discretization of (CFP). Let $C^1([0,T];\boldsymbol{S}_{h\mathsf{p}})$ denote the space of continuously differentiable mappings of the interval $[0,T]$ into $\boldsymbol{S}_{h\mathsf{p}}$.

Definition 4 *Function* $\boldsymbol{w}_h \in C^1([0,T];\boldsymbol{S}_{h\mathsf{p}})$ *is called the* semi-discrete solution *of (*CFP*), if*

$$\text{a)} \qquad \left(\frac{\partial \boldsymbol{w}_h(t)}{\partial t}, \boldsymbol{\varphi}_h\right) + \boldsymbol{c}_h(\boldsymbol{w}_h(t), \boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = \tilde{\boldsymbol{c}}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) \qquad \forall\, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathsf{p}}\ \forall\, t \in (0,T), \quad (4.35)$$

$$\text{b)} \qquad \boldsymbol{w}_h(0) = \boldsymbol{w}_h^0,$$

where $\boldsymbol{w}_h^0 \in \boldsymbol{S}_{h\mathsf{p}}$ *denotes an* $\boldsymbol{S}_{h\mathsf{p}}$*-approximation of the initial condition* $\boldsymbol{w}^0$ *from (4.9).*

The problem (4.35), a) – b) represents a system of ordinary differential equations (ODEs) for $\boldsymbol{w}_h(t)$ which has to be discretized in time by a suitable method. Since these ODEs represent a stiff system, a use of a (semi-)implicit method is advantageous. Therefore, we employ the *semi-implicit* technique developed in [Dol08] and [DF04] which is based on the linearity of the form $\boldsymbol{c}_h(\cdot,\cdot,\cdot)$ with respect to its second argument following from expressions (4.28) – (4.32). Hence, for the first order scheme with respect to time, we approximate the time derivative term in (4.35), a) by backward Euler method, the second argument of $\boldsymbol{c}_h(\cdot,\cdot,\cdot)$ is treated implicitly and the first one explicitly.

Let $0 = t_0 < t_1 < t_2 < \ldots t_r = T$ be a partition of the time interval $(0,T)$, $\tau_k := t_k - t_{k-1}$, and $\boldsymbol{w}_h^k \in \boldsymbol{S}_{h\mathsf{p}}$ denotes a piecewise polynomial approximation of $\boldsymbol{w}_h(t_k)$, $k = 0, 1, \ldots, r$. We define the following scheme.

Definition 5 *The* approximate solution *of (*CFP*) by the semi-implicit DGFE scheme is defined as functions* $\boldsymbol{w}_{h,k}$, $k = 1, \ldots, r$, *satisfying the conditions*

$$\textit{a)} \qquad \boldsymbol{w}_{h,k} \in \boldsymbol{S}_{h\mathsf{p}}, \quad (4.36)$$

$$\textit{b)} \qquad \frac{1}{\tau_k}\left(\boldsymbol{w}_{h,k} - \boldsymbol{w}_{h,k-1},\ \boldsymbol{\varphi}_h\right) + \boldsymbol{c}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{w}_{h,k},\ \boldsymbol{\varphi}_h\right) = \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{\varphi}_h\right) \qquad \forall\, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathsf{p}},\ k = 1, \ldots, r,$$

$$\textit{c)} \qquad \boldsymbol{w}_{h,0} \in S_{h\mathsf{p}} \textit{ is an approximation of } \boldsymbol{w}^0.$$

The method (4.36) is a first order scheme with respect to the time which is sufficient for steady-state problems. Otherwise, it is possible to use multi-step backward difference formulae for the time discretization, see [Dol08]. Then all considerations presented in this paper have to be slightly modified. However, we consider only the first order scheme with respect to the time for simplicity.

The problem (4.36), a) – c) represents a linear algebraic system for each $k = 1, \ldots, r$ which should be solved by a suitable solver, which is discussed in Section 4.3. Numerical experiments show that the resulting semi-implicit DGFE method is practically unconditionally stable, i.e., the size of the time step can be chosen very large, see [Dol08].

Finally, we introduce the discrete problem for the stationary Navier-Stokes equations.

Definition 6 *Function* $\boldsymbol{w}_h \in \boldsymbol{S}_{h\mathsf{p}}$ *is called the* discrete solution *of (*sCFP*), if*

$$\boldsymbol{c}_h(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) = \tilde{\boldsymbol{c}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) \qquad \forall\, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathsf{p}}. \quad (4.37)$$

p_μ	1	2	3	4	5
$d=2$	12	24	40	60	84
$d=3$	20	50	100	175	280

Table 4.1: Values of dof_μ for $p_\mu = 1,\dots,5$ and $d = 2,3$

We already mentioned in Introduction, the non-stationary formulation (4.35) will be used for the solution of (sCFP).

4.3 Solution strategy

In this section, we deal with an efficient solution strategy of the (stationary) discrete problem (4.37). However, its direct solution causes some troubles mentioned bellow. Hence we employ the non-stationary problem (4.36) for its solution.

4.3.1 Linear algebra representation

Basis of $\boldsymbol{S}_{h\mathsf{p}}$

Let us introduce an index set $I \subset \boldsymbol{Z}^+$(=set of all positive integers) numbering elements $K \in \mathcal{T}_h$, i.e., $\mathcal{T}_h = \{K_\mu,\ \mu \in I\}$. By $p_\mu = p_{K_\mu}$ we denote the degree of polynomial approximation on element $K_\mu,\ \mu \in I$. Since $\boldsymbol{S}_{h\mathsf{p}}$ is a space of discontinuous piecewise polynomial functions, it is possible to consider a set of linearly independent polynomial functions on K_μ for each $K_\mu \in \mathcal{T}_h$

$$B_\mu = \Big\{\boldsymbol{\psi}_{\mu,j};\ \boldsymbol{\psi}_{\mu,j} \in \boldsymbol{S}_{h\mathsf{p}},\ \mathrm{supp}(\boldsymbol{\psi}_{\mu,j}) \subseteq K_\mu, \quad \boldsymbol{\psi}_{\mu,j} \text{ are linearly independent for } j = 1,\dots,\mathsf{dof}_\mu\Big\}, \tag{4.38}$$

where

$$\mathsf{dof}_\mu = \frac{d+2}{d!}\Pi_{j=1}^{d}(p_\mu + j), \quad \mu \in I, \tag{4.39}$$

denotes the number of *local degrees of freedom* for each element $K_\mu \in \mathcal{T}_h$ (we recall that (CFP) and (sCFP) represent $d+2$ equations). The values dof_μ are shown in Table 4.1 for $p_\mu = 1,\dots,5$ and $d = 2,3$. We call B_μ the *local basis* on K_μ. For the construction of the basis $B_\mu,\ \mu \in I$, see Section 4.3.1.

A composition of the local basis $B_\mu,\ \mu \in I$ defines a basis of $\boldsymbol{S}_{h\mathsf{p}}$, i.e.,

$$B = \{\boldsymbol{\psi}_j;\ \boldsymbol{\psi}_j \in \boldsymbol{S}_{h\mathsf{p}},\ j = 1,\dots,\mathsf{dof}\}. \tag{4.40}$$

By dof, we denote the dimension of $\boldsymbol{S}_{h\mathsf{p}}$ (=number of elements of the basis B) which is equal to $\mathsf{dof} = \sum_{\mu\in I}\mathsf{dof}_\mu$.

Therefore, a function $\boldsymbol{w}_{h,k} \in \boldsymbol{S}_{h\mathsf{p}}$ can be written in the form

$$\boldsymbol{w}_{h,k}(x) = \sum_{\mu\in I}\sum_{j=1}^{\mathsf{dof}_\mu}\xi_{k,\mu,j}\boldsymbol{\psi}_{\mu,j}(x), \quad x \in \Omega,\ k = 0,1,\dots,r, \tag{4.41}$$

where $\xi_{k,\mu,j} \in I\!R,\ j = 1, \ldots, \mathsf{dof}_\mu,\ \mu \in I,\ k = 0, \ldots, r$. Moreover, for $\boldsymbol{w}_{h,k} \in \boldsymbol{S}_{h\mathsf{p}}$, we define a vector of its basis coefficients by

$$\boldsymbol{W}_k = \{\xi_{k,\mu,j}\}_{j=1,\ldots,\mathsf{dof}_\mu}^{\mu\in I} \in I\!R^{\mathsf{dof}}, \quad k = 0, 1, \ldots, r. \tag{4.42}$$

Therefore, using (4.41) – (4.42) we have an isomorphism

$$\boldsymbol{w}_{h,k} \in \boldsymbol{S}_{h\mathsf{p}} \quad \longleftrightarrow \quad \boldsymbol{W}_k \in I\!R^{\mathsf{dof}}. \tag{4.43}$$

Linear algebraic systems

Using isomorphism (4.43), problem (4.36) can be written in the matrix form

$$\underbrace{\left(\frac{1}{\tau_k}\boldsymbol{M} + \boldsymbol{C}(\boldsymbol{W}_{k-1})\right)}_{=:\boldsymbol{A}_k}\boldsymbol{W}_k = \underbrace{\frac{1}{\tau_k}\boldsymbol{m}(\boldsymbol{W}_{k-1}) + \boldsymbol{q}(\boldsymbol{W}_{k-1})}_{=:\boldsymbol{d}_k}, \quad k = 1, \ldots, r, \tag{4.44}$$

where the matrix $\boldsymbol{M}$ is a block-diagonal *mass matrix* given by

$$\boldsymbol{M} = \mathrm{diag}\{\boldsymbol{M}_{\mu,\mu},\ \mu \in I\}, \qquad \boldsymbol{M}_{\mu,\mu} = \{M_\mu^{i,j}\}_{i,j=1}^{\mathsf{dof}_\mu},\ \mu \in I, \tag{4.45}$$
$$M_\mu^{i,j} = \int_\Omega \boldsymbol{\psi}_{\mu,i} \cdot \boldsymbol{\psi}_{\mu,j}\, \boldsymbol{dx},$$

the matrix $\boldsymbol{C}$ is the *"flux" matrix* corresponding to form $\boldsymbol{c}_h$ defined by

$$\boldsymbol{C}(\boldsymbol{W}_{k-1})) = \{C_{(\mu,i),(\nu,j)}(\boldsymbol{W}_{k-1})\}_{\mu,\nu\in I}^{i=1,\ldots,\mathsf{dof}_\mu, j=1,\ldots,\mathsf{dof}_\nu}, \tag{4.46}$$
$$C_{(\mu,i),(\nu,j)}(\boldsymbol{W}_{k-1}) = \boldsymbol{c}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{\psi}_{\nu,j},\ \boldsymbol{\psi}_{\mu,i}\right),$$

$\boldsymbol{m} \in I\!R^{\mathsf{dof}}$ represents the "explicit" part of the approximation of the time derivative in (4.36), a) defined by

$$\boldsymbol{m}(\boldsymbol{W}_{k-1}) = \boldsymbol{M}\boldsymbol{W}_{k-1} = \{m_{\mu,i}(\boldsymbol{W}_{k-1})\}_{\mu\in I}^{i=1,\ldots,\mathsf{dof}_\mu}, \quad m_{\mu,i}(\boldsymbol{W}_{k-1}) = \left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{\psi}_{\mu,i}\right), \tag{4.47}$$

and $\boldsymbol{q} \in I\!R^{\mathsf{dof}}$ represents form $\tilde{\boldsymbol{c}}_h$ in (4.36), a) given by

$$\boldsymbol{q}(\boldsymbol{W}_{k-1}) = \{q_{\mu,i}(\boldsymbol{W}_{k-1})\}_{\mu\in I}^{i=1,\ldots,\mathsf{dof}_\mu}, \quad q_{\mu,i}(\boldsymbol{W}_{k-1}) = \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{\psi}_{\mu,i}\right). \tag{4.48}$$

In virtue of the local character of basis B it is easy to observe that the matrix $\boldsymbol{C}$ has a block structure. It follows from the expressions (4.28), (4.23), (4.26) and (4.32) that the matrix element $C_{(\mu,i),(\nu,j)}$ is non-vanishing if $\mu = \nu$ or if elements K_μ and K_ν share an face. The size of a non-diagonal block is equal to $\mathsf{dof}_\mu \times \mathsf{dof}_\nu$ and the number of non-diagonal blocks corresponding to an element $K_\mu \in \mathcal{T}_h$ is equal to the number of neighbouring elements of K_μ. Then we can write the block-structure of $\boldsymbol{C}$ by

$$\boldsymbol{C} \quad = \quad \{\boldsymbol{C}_{\mu,\nu}\}_{\mu,\nu\in I}, \quad \boldsymbol{C}_{\mu,\nu} = \{C_{(\mu,i),(\nu,j)}\}_{i=1,\ldots,\mathsf{dof}_\mu}^{j=1,\ldots,\mathsf{dof}_\nu}, \tag{4.49}$$

where $\boldsymbol{C}_{\mu,\nu}$ represents a $\mathsf{dof}_\mu \times \mathsf{dof}_\nu$-block with elements $C_{(\mu,i),(\nu,j)}$ given by (4.46), $\mu, \nu \in I$. Obviously, block $\boldsymbol{C}_{\mu,\nu}$ is non-vanishing only if $\mathrm{meas}_{d-1}(\partial K_\mu \cap \partial K_\nu) > 0$. Here, meas_{d-1} denotes the $d-1$ dimensional Lebesgue measure.

Remark 4 *Let us still mention that numerical experiments show that the norms of blocks of the mass matrix and the flux matrix satisfy the relations*

$$\|\boldsymbol{M}_{\mu,\mu}\| \approx 10^{-3}\|\boldsymbol{C}_{\mu,\mu}\|, \quad \mu \in I \qquad \text{and} \qquad \|\boldsymbol{C}_{\mu,\nu}\| \lessapprox \|\boldsymbol{C}_{\mu,\mu}\|, \quad \mu,\nu \in I. \tag{4.50}$$

This implies that $\boldsymbol{A}_k \approx \boldsymbol{C}_k$ *and* $\boldsymbol{d}_k \approx \boldsymbol{q}_k$ *for* $\tau_h \gtrapprox 10^{12}$ *in the double precision arithmetic.*

Obviously, the size of τ_k has a great impact on the property of the matrix $\boldsymbol{A}_k$. Let $\lambda_{\min}(\boldsymbol{C})$ and $\lambda_{\max}(\boldsymbol{C})$ denote the minimal and the maximal eigenvalues of $\boldsymbol{C}(\cdot)$, respectively. Similarly, $\lambda_{\min}(\boldsymbol{A}_k)$ and $\lambda_{\max}(\boldsymbol{A}_k)$ denote the minimal and the maximal eigenvalues of $\boldsymbol{A}_k$, respectively. If the basis B is orthonormal then $\boldsymbol{M}$ is the identity matrix and the condition number of matrix $\boldsymbol{A}_k$ is given by

$$\mathrm{cond}(\boldsymbol{A}_k) := \frac{\lambda_{\max}(\boldsymbol{A}_k)}{\lambda_{\min}(\boldsymbol{A}_k)} = \frac{\frac{1}{\tau_k} + \lambda_{\max}(\boldsymbol{C}(\boldsymbol{W}_k))}{\frac{1}{\tau_k} + \lambda_{\min}(\boldsymbol{C}(\boldsymbol{W}_k))} = \frac{1 + \tau_k\lambda_{\max}(\boldsymbol{C}(\boldsymbol{W}_k))}{1 + \tau_k\lambda_{\min}(\boldsymbol{C}(\boldsymbol{W}_k))}, \tag{4.51}$$

Therefore, for increasing τ_k the condition number of $\boldsymbol{A}_k$ is higher because the condition number of $\boldsymbol{C}(\boldsymbol{W}_k)$ is usually high. Finally, let us recall that a high condition number can cause troubles in convergence of an iterative method.

Construction of the basis of $\boldsymbol{S}_{h\mathsf{p}}$

We employ the local character of the shape functions and construct basis of $\boldsymbol{S}_{h\mathsf{p}}$ which is orthonormal with respect to the L^2-scalar product. We define a basis of the space of vector-valued polynomials of degree $\le p$ on the reference element $\hat{K}$ by

$$\begin{aligned} \hat{\boldsymbol{S}}_p &= \hat{S}_p \times \ldots \times \hat{S}_p \quad (d+2 \text{ times}), \\ \hat{S}_p &= \{\phi_{n_1,\ldots,n_d}(\hat{x}_1,\ldots,\hat{x}_d) = \Pi_{i=1}^d(\hat{x}_i - \hat{x}_i^c)^{n_i};\ n_1,\ldots,n_d \ge 0,\ \sum_{j=1}^d n_j \le p\}, \end{aligned} \tag{4.52}$$

where $(\hat{x}_1^c,\ldots,\hat{x}_d^c)$ is the barycentre of $\hat{K}$. Obviously, the set $\hat{\boldsymbol{S}}_p$ is a basis of the space of vector-valued polynomials on the reference element $\hat{K}$ of degree $\le p$. By the Gram-Schmidt orthogonalization process applied to $\hat{\boldsymbol{S}}_p$, we obtain the orthonormal set $\{\hat{\phi}_j,\ j = 1,\ldots,\mathsf{dof}_\mu\}$ where dof_μ is given by (4.39) with $p_\mu := p$. The Gram-Schmidt orthogonalization on the reference element can be carried out by a symbolical computing since dof_μ is small (moreover, the orthogonalization can be done for each component of $\boldsymbol{S}_{h\mathsf{p}}$ independently). Hence this orthogonalization does not cause any loose of the accuracy.

Furthermore, let $F_\mu := F_{K_\mu},\ \mu \in I$, be the mapping introduced in Section 4.2.1 such that $F_\mu(\hat{K}) = K_\mu$. We put

$$B_\mu := \{\boldsymbol{\psi}_{\mu,j};\ \boldsymbol{\psi}_{\mu,j}(x) = \boldsymbol{\psi}_{\mu,j}(F_\mu(\hat{x})) = \hat{\phi}_j(\hat{x}),\ \hat{x} \in \hat{K},\ j = 1,\ldots,\mathsf{dof}_\mu\}, \tag{4.53}$$

which define a basis B_μ introduced in (4.38) for each element $K_\mu \in \mathcal{T}_h$ separately. If F_μ is linear then basis B_μ is orthogonal with respect to the L^2-scalar product and

blocks $\boldsymbol{M}_{\mu,\mu}$ of the mass matrix $\boldsymbol{M}$ given by (4.45) are diagonal. If F_μ is not linear then the orthogonality of B_μ is violated. However, in practical applications, the curved face $K_\mu \cap \partial\Omega$ is close to a straight (polygonal) one and thus the matrix block $\boldsymbol{M}_{\mu,\mu}$ is strongly diagonally dominant.

Finally, (4.40) defines the (almost) orthogonal basis of $\boldsymbol{S}_{h\mathsf{p}}$ and the orthonormalization can be done by a simple scaling of basis functions.

4.3.2 Abstract solution strategy

In virtue of (4.44), the stationary discrete problem (4.37) can be written in the form: find $\boldsymbol{W} \in I\!R^{\mathsf{dof}}$ such that

$$\boldsymbol{C}(\boldsymbol{W})\boldsymbol{W} = \boldsymbol{q}(\boldsymbol{W}). \tag{4.54}$$

The problem (4.54) represents a system of strongly nonlinear algebraic equations. It is natural to define formally an iterative method for solving (4.54) by: find $\boldsymbol{W}_k \in I\!R^{\mathsf{dof}},\ k = 1, 2, \ldots$ such that

$$\boldsymbol{C}(\boldsymbol{W}_{k-1})\boldsymbol{W}_k = \boldsymbol{q}(\boldsymbol{W}_{k-1}), \qquad k = 1, 2, \ldots, \tag{4.55}$$

and put $\boldsymbol{W} := \lim_{k\to\infty} \boldsymbol{W}_k$. However, this simple method works only if the initial guess $\boldsymbol{W}_0$ is very close to solution $\boldsymbol{W}$ of (4.54). Otherwise, the iterative process (4.55) fails since nonphysical solutions appear.

One possibility how to avoid this principle obstacle is a use of the unsteady formulation (4.44). The idea is natural. We start with a small time step for small k when the approximation $\boldsymbol{W}_k$ is far from $\boldsymbol{W}$. Then, when $\boldsymbol{W}_k$ is approaching to the limit vector $\boldsymbol{W}$ for increasing k, we successively increase the size of the time step τ_k. Consequently, the problems (4.44) lead to (4.55) for $\tau_k \to \infty$. On the other hand, the non-stationary discretization (4.44) can be considered as a "relaxation" of method (4.55) and the ratio $1/\tau_k$ as the relaxation parameter.

Remark 5 *Another possibility is a direct solution of the nonlinear algebraic systems (4.54) by the Newton method (see, e.g., [HH06a]). The advantage of the unsteady formulation (4.44) is its applicability to unsteady problems with any modification of the scheme.*

As we already mention, it is suitable to employ an iterative solver for the solution of the linear algebraic systems (4.44) since the solution $\boldsymbol{W}_k,\ k = 1, 2, \ldots$ obtained in k^{th}-iteration can be used as initial guess for $\boldsymbol{W}_{k+1}$. We employ the GMRES method [SS86] with a suitable preconditioner which represents a widely used technique for the solution of non-symmetric sparse linear algebraic systems.

Now, we are ready to introduce

Abstract algorithm (AA)

1. let $\boldsymbol{W}_0 \longleftrightarrow \boldsymbol{w}_h^0$ be given
2. for $k = 1$ to r
   (a) set τ_k

(b) from $\boldsymbol{W}_{k-1}$ evaluate $\boldsymbol{A}_k(\boldsymbol{W}_{k-1})$ and $\boldsymbol{d}_k(\boldsymbol{W}_{k-1})$

(c) solve $\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_k = \boldsymbol{d}_k(\boldsymbol{W}_{k-1})$ by:

i. $\boldsymbol{W}_k^0 := \boldsymbol{W}_{k-1}$

ii. $\boldsymbol{W}_k^{l+1} := \mathsf{GMRES\_iter}(\boldsymbol{W}_k^l),\ \ l = 1, \ldots, s_k$

iii. $\boldsymbol{W}_k := \boldsymbol{W}_k^{s_k}$

3. $\boldsymbol{W} := \boldsymbol{W}_r$.

Here, $\mathsf{GMRES\_iter}(\cdot)$ formally denotes a performance of one GMRES step.

In the abstract algorithm (AA), there are still open questions how to choose the total number of time levels r, the size of $\tau_k,\ k = 1, \ldots, r$ and the number of GMRES steps s_k at each time level. Therefore, in order to define a real algorithm we have to specify the *algorithm settings*, namely

- *stopping steady-state criterion*, i.e., when to stop the global iterative loop,
- *preconditioner*, i.e., how to carried out one GMRES step,
- *GMRES stopping criterion*, i.e., how many GMRES steps have to be employed at each time level,
- *choice of the time step* τ_k.

Our goal is to define the previous algorithm settings in order to achieve the final limit vector $\boldsymbol{W}$ as soon as possible (measured in terms of the computational time).

Obviously, all these aspects are not independent and have a significantly influence on the accuracy and efficiency of the method. E.g., in virtue of the comments of the end of Section 4.3.1, large τ_k leads to a higher number of GMRES steps for solving $\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_k = \boldsymbol{d}_k(\boldsymbol{W}_{k-1})$.

In the following we discuss these aspects separately.

4.3.3 Algorithm settings

Stopping steady-state criterion

We discuss when to stop the global loops in the algorithm (AA), i.e., when we achieve the *steady-state solution*. The usual steady-state criterion, often used for explicit time discretization, is

$$\left\| \frac{\partial \boldsymbol{w}_h}{\partial t} \right\| \approx \eta_k := \frac{1}{\tau_k} \|\boldsymbol{w}_h^k - \boldsymbol{w}_h^{k-1}\|_{L^2(\Omega)} = \frac{1}{\tau_k} \|\boldsymbol{W}_k - \boldsymbol{W}_{k-1}\|_{\ell^2} \leq \mathrm{TOL}, \tag{4.56}$$

where TOL is a given tolerance. However, this criterion does not make good sense for the semi-implicit time discretization when very large time steps can be employed. Then in virtue of Remark 4, there exists a limit value of τ_k when $\boldsymbol{A}_k(= \boldsymbol{C}_k)$ and $\boldsymbol{d}_k(= \boldsymbol{q}_k)$ are independent of τ_k (in the finite precision arithmetic) whereas (4.56) depends on τ_k. Then by a very large choice of τ_k we can achieve very small value of η_k in (4.56) although we are far from the steady-state solution.

On the other hand, in virtue of (4.54), it is possible to employ the *steady-state residual criterion*

$$\|\boldsymbol{C}(\boldsymbol{W}_k)\boldsymbol{W}_k - \boldsymbol{q}(\boldsymbol{W}_k)\|_{\ell^2} \leq \mathrm{TOL}, \tag{4.57}$$

which is independent of τ_k and measures the residuum of the nonlinear algebraic system (4.54).

The condition (4.57) has also a nice interpretation in the framework of the functional analysis. Let R denote the operator from $\boldsymbol{S}_{h\mathsf{p}}$ to its dual space defined by

$$\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_{h\mathsf{p}}: \quad \langle R\boldsymbol{w}_h, \boldsymbol{\varphi}_h\rangle := \boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \boldsymbol{\varphi}_h\right), \tag{4.58}$$

where the forms $\boldsymbol{c}_h$ and $\tilde{\boldsymbol{c}}_h$ are defined by (4.32). It follows from (4.35) that $\boldsymbol{w}_h$ is the (discrete) steady state solution if

$$R\boldsymbol{w}_h = 0 \qquad \Longleftrightarrow \qquad \langle R\boldsymbol{w}_h, \boldsymbol{\varphi}_h\rangle = 0 \quad \forall \boldsymbol{\varphi}_h. \tag{4.59}$$

Let $\varphi \in \boldsymbol{S}_{h\mathsf{p}}$ and let $\phi_i,\ i = 1, \ldots, \mathsf{dof}$, be its basis coefficients, i.e., $\varphi = \sum_{i=1}^{\mathsf{dof}} \phi_i \boldsymbol{\psi}_i$. Since $\{\boldsymbol{\psi}_i\}_{i=1}^{\mathsf{dof}}$ is the orthonormal basis of $\boldsymbol{S}_{h\mathsf{p}}$ we have $\|\varphi\|^2_{L^2(\Omega)} = \sum_{i=1}^{\mathsf{dof}} \phi_i^2$. Let $\boldsymbol{w}_h^k \leftrightarrow \boldsymbol{W}_k$ be the approximate solution of the steady-state problem (4.59), then the corresponding residuum satisfies the estimate

$$\begin{aligned}
\|R\boldsymbol{w}_h^k\|_{L^2(\Omega)} &= \sup_{\varphi \in \boldsymbol{S}_{h\mathsf{p}}} \frac{\langle R\boldsymbol{w}_h, \varphi\rangle}{\|\varphi\|_{L^2(\Omega)}} = \sup_{\varphi \in \boldsymbol{S}_{h\mathsf{p}}} \frac{\boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \varphi\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \varphi\right)}{\|\varphi\|_{L^2(\Omega)}} \\
&= \sup_{\phi_i,\ i=1,\ldots,\mathsf{dof}} \frac{\sum_{i=1}^{\mathsf{dof}} \phi_i \left(\boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\psi}_i\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \boldsymbol{\psi}_i\right)\right)}{\left(\sum_{i=1}^{\mathsf{dof}} \phi_i^2\right)^{1/2}} \\
&\leq \sup_{\phi_i,\ i=1,\ldots,\mathsf{dof}} \frac{\left(\sum_{i=1}^{\mathsf{dof}} \phi_i^2\right)^{1/2} \left(\sum_{i=1}^{\mathsf{dof}} \left(\boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\psi}_i\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \boldsymbol{\psi}_i\right)\right)^2\right)^{1/2}}{\left(\sum_{i=1}^{\mathsf{dof}} \phi_i^2\right)^{1/2}} \\
&= \left(\sum_{i=1}^{\mathsf{dof}} \left(\boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\psi}_i\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \boldsymbol{\psi}_i\right)\right)^2\right)^{1/2} = \|\boldsymbol{C}(\boldsymbol{W}_k)\boldsymbol{W}_k - \boldsymbol{q}(\boldsymbol{W}_k)\|_{\ell^2}.
\end{aligned} \tag{4.60}$$

The last equality follows from the fact that, in virtue of (4.46) and (4.48), the value

$$\boldsymbol{c}_h\left(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\psi}_i\right) - \tilde{\boldsymbol{c}}_h\left(\boldsymbol{w}_h, \boldsymbol{\psi}_i\right),\ i = 1, \ldots, \mathsf{dof}$$

is equal to the i^{th} component of the vector $\boldsymbol{C}(\boldsymbol{W}_k)\boldsymbol{W}_k - \boldsymbol{q}(\boldsymbol{W}_k),\ i = 1, \ldots, \mathsf{dof}$.

However, there is still open how to choose the tolerance TOL in (4.57) since the residuum depends on the size of the computational domain Ω, on the magnitude of components of $\boldsymbol{w}_h$, etc. Therefore, from the practical reasons, we use the *relative residuum steady-state criterion*

$$\mathrm{SSres}(k) := \frac{\|\boldsymbol{C}(\boldsymbol{W}_k)\boldsymbol{W}_k - \boldsymbol{q}(\boldsymbol{W}_k)\|_{\ell^2}}{\|\boldsymbol{C}(\boldsymbol{W}_0)\boldsymbol{W}_0 - \boldsymbol{q}(\boldsymbol{W}_0)\|_{\ell^2}} \leq \mathrm{TOL}, \tag{4.61}$$

which already does not suffer from the mentioned drawbacks.

Another possibility are the stopping criteria which follow from the physical background of the considered problem. Many often, we are interested in the *aerodynamic coefficients* of the considered flow, namely coefficients of *drag* (c_D), *lift* (c_L) and *momentum* (c_M). Then the natural choice is to stop global iterative loops when these coefficients achieve a given tolerance, e.g.,

$$\Delta c_x(k) \le \mathrm{tol}, \qquad \Delta c_x(k) := \max_{l=0.9k,\ldots,k} c_x(l) - \min_{l=0.9k,\ldots,k} c_x(l), \tag{4.62}$$

where tol is a given tolerance, subscript x takes the values D, L and M (drag, lift, momentum), $c_x(k)$ is the value of the corresponding aerodynamic coefficient at k^{th}-time level and the minimum and maximum in (4.62) are taken over the last 10% of the number of time levels.

Whereas the tolerance TOL in (4.61) has to be chosen empirically, the tolerance tol in (4.62) can be chosen only on the base of our accuracy requirements (without any previous numerical experiments). Since the absolute values of aerodynamic coefficient are (usually) less than one, the stopping criterion (4.62) with tolerance, e.g., $\mathrm{tol} = 10^{-4}$, gives accuracy of the aerodynamic coefficients for 2 or 3 decimal digits.

Choice of the preconditioner

The matrices $\boldsymbol{A}_k,\ k = 1,\ldots,r$ in (4.44) are ill-conditioned (for not too small values of τ_k). Then the GMRES iterative process requires many steps in order to solve the corresponding linear algebraic system. Therefore, it is convenient to apply a suitable preconditioner $\hat{\boldsymbol{P}}_k \in I\!R^{\mathsf{dof}\times\mathsf{dof}}$ to problem (4.44) which leads to the new linear algebraic system

$$\hat{\boldsymbol{P}}_k \boldsymbol{A}_k \boldsymbol{W}_k = \hat{\boldsymbol{P}}_k \boldsymbol{q}_k, \quad k = 1,\ldots,r, \tag{4.63}$$

equivalent to (4.44). Suitable preconditioner means that the condition number of the matrix $\hat{\boldsymbol{P}}_k \boldsymbol{A}_k$ is significantly smaller than the condition number of $\boldsymbol{A}_k$. In practice we need not to construct matrix $\hat{\boldsymbol{P}}_k$ but it is sufficient to evaluate the product $\hat{\boldsymbol{P}}_k \boldsymbol{y}$ for a given vector $\boldsymbol{y} \in I\!R^{\mathsf{dof}}$.

Our aim is to use a preconditioner which has

- a *high efficiency*, i.e., it significantly reduces the number of steps of the iterative (GMRES) solver,
- a *low costs*, i.e., its evaluation is fast and it does not require any additional memory.

Matrices $\boldsymbol{C}_k$ have block structure then it is more efficient to employ block preconditioners. We discuss two variants of block preconditioners.

The simplest one is the *block diagonal preconditioner* (BDP) when $\hat{\boldsymbol{P}}_k = \mathrm{diag}\{\hat{\boldsymbol{P}}_{k,\mu,\mu}, \mu \in I\}$ has only diagonal blocks given by

$$\hat{\boldsymbol{P}}_{k,\mu,\mu} := \left(\frac{1}{\tau_k}\boldsymbol{M}_{\mu,\mu} + \boldsymbol{C}_{k,\mu,\mu}\right)^{-1}, \quad k = 1,\ldots,r. \tag{4.64}$$

Practically, we do not evaluate the matrix inversion but we carried out the (full) LU-decomposition (see, e.g., [Wat02]) of each diagonal block. Then any additional memory

is not required. Since the diagonal blocks are relatively small (see Table 4.1), their LU decompositions are fast. Our numerical experiments indicate that the efficiency of BDP is not bad for the solution of viscous flows. However, GMRES iterative process with BDP does not converge for the solution of inviscid flow in some situations (for finer grids and higher degree of polynomial approximations).

Therefore, we employ the *block incomplete LU* (ILU) preconditioner, when the incomplete LU factorization is computed by performing block Gaussian elimination on $\boldsymbol{A}_k$ but ignoring blocks which would result in any additional fill of the matrix. Let $\boldsymbol{A} = \boldsymbol{A}_k \in I\!R^{\mathsf{dof}\times\mathsf{dof}}$ be a matrix consisting of blocks $\{\boldsymbol{A}_{\mu,\nu}\}_{\mu,\nu\in I}$. Its sparsity is given by the geometry of mesh $\mathcal{T}_h$ and the size of blocks of $\boldsymbol{A}$ is given by $p_\mu,\ \mu \in I$.

Let I denote a generic identity square matrix and let $\mathbf{0}$ denote a generic zero matrix. Let us recall that

$$\text{if} \quad \mathrm{meas}_{d-1}(\partial K_\mu \cap \partial K_\nu) = 0 \quad \text{then} \quad \boldsymbol{A}_{\mu,\nu} \equiv \mathbf{0} \quad \text{else} \quad \boldsymbol{A}_{\mu,\nu} \not\equiv \mathbf{0}. \tag{4.65}$$

We define block matrices $\boldsymbol{L} = \{\boldsymbol{L}_{\mu,\nu}\}_{\mu,\nu\in I}, \boldsymbol{U} = \{\boldsymbol{U}_{\mu,\nu}\}_{\mu,\nu\in I}$ such that

i)

$$\boldsymbol{L}_{\mu,\nu}\begin{cases} = \mathrm{I} & \text{if } \mu = \nu, \\ \equiv \mathbf{0} & \text{if } \mu < \nu \text{ or } \boldsymbol{A}_{\mu,\nu} \equiv \mathbf{0}, \\ \not\equiv \mathbf{0} & \text{elsewhere,} \end{cases} \qquad \boldsymbol{U}_{\mu,\nu}\begin{cases} \equiv \mathbf{0} & \text{if } \mu > \nu \text{ or } \boldsymbol{A}_{\mu,\nu} \equiv \mathbf{0}, \\ \not\equiv \mathbf{0} & \text{elsewhere,} \end{cases} \tag{4.66}$$

ii) the blocks $\boldsymbol{L}_{\mu,\nu}$ and $\boldsymbol{U}_{\mu,\nu}$ have the same shape as $\boldsymbol{A}_{\mu,\nu},\ \mu,\nu \in I$,

iii) if $\boldsymbol{A}_{\mu,\nu}$ is non-vanishing for some pair $\mu,\nu \in I$ then

$$\boldsymbol{A}_{\mu,\nu} = \sum_{\rho\in I} \boldsymbol{L}_{\mu,\rho}\boldsymbol{U}_{\rho,\nu}. \tag{4.67}$$

The block ILU preconditioner is formally defined by $\hat{\boldsymbol{P}}_k = (\boldsymbol{LU})^{-1}$. Obviously, it is not necessary to construct matrix $(\boldsymbol{LU})^{-1}$ itself but it is sufficient to evaluate the product $(\boldsymbol{LU})^{-1}\boldsymbol{y}$ for a given vector $\boldsymbol{y} \in I\!R^{\mathsf{dof}}$. I.e., let $\boldsymbol{z} \in I\!R^{\mathsf{dof}}$ such that $\boldsymbol{z} = (\boldsymbol{LU})^{-1}\boldsymbol{y}$ then

$$\boldsymbol{z} = (\boldsymbol{LU})^{-1}\boldsymbol{y} \quad \Longrightarrow \quad \boldsymbol{L}\bar{\boldsymbol{z}} = \boldsymbol{y}, \quad \text{where} \quad \bar{\boldsymbol{z}} = \boldsymbol{U}\boldsymbol{z}. \tag{4.68}$$

Then in order to obtain $\boldsymbol{z}$ it is sufficient to solve two block triangular linear algebraic systems $\boldsymbol{L}\bar{\boldsymbol{z}} = \boldsymbol{y}$ and $\boldsymbol{U}\boldsymbol{z} = \bar{\boldsymbol{z}}$. The construction of $\boldsymbol{L}$ and $\boldsymbol{U}$ can be done by a standard LU decomposition (see, e.g., [Wat02], [QSS00]) when we simply omit the vanishing blocks.

The relation (4.67) implies that it is sufficient to store only blocks of $\boldsymbol{L}$ and $\boldsymbol{U}$ instead of $\boldsymbol{A}$ and thus no additional memory is needed. Moreover, if mesh $\mathcal{T}_h$ satisfies that neighbours of an element do not neighbor one another then the sum in (4.67) has only one non-vanishing term since only two blocks of $\boldsymbol{A}_{\mu,\nu}$, $\boldsymbol{L}_{\mu,\rho}$ and $\boldsymbol{U}_{\rho,\nu}$ are non-vanishing for any $\mu \neq \nu \neq \rho$. Mostly mesh generators produce grids satisfying this property. Hence, the construction of the ILU precondition is relatively fast. Our numerical experiments show that the construction of ILU preconditioner requires the same computational time as one or two GMRES steps.

The block ILU factorization is described in details in [DD09, Section 3.4], where also the computational efficiency is analysed and studied.

GMRES stopping criterion

In this section we deal with the stopping criterion of the inner loops in (AA), i.e., when to stop the GMRES iterative process at each time level $k = 1, \dots, r$. It is clear that too weak criterion can decrease accuracy and on the other hand, too strong criterion decreases the efficiency. Usually, one uses some (preconditioned) residuum criterion but the main problem is the setting of the given tolerance. Since our aim is to obtain the final steady-state solution $\boldsymbol{W}$ from (AA) as fast as possible it does not make sense to solve the linear algebraic systems (4.44) too precisely.

In [DES82], the so-called *inexact Newton method* was proposed for the solution of a system of nonlinear algebraic equations. The main idea is that the linear algebraic systems (arising from the Newton method) are solved by an iterative solver till the residuum is only a few times smaller than the residuum of the initial guess of the solution (taken from the previous level).

Using this idea, we propose the following stopping criterion for the GMRES method at each time level:

$$\|\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_k - \boldsymbol{d}_k(\boldsymbol{W}_{k-1})\| \leq \delta_k \|\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_{k-1} - \boldsymbol{d}_k(\boldsymbol{W}_{k-1})\|, \tag{4.69}$$

$k = 1, \dots, r$, where $\delta_k \in (0, 1)$ is a given value, the left-hand side is the residuum and the term on the right-hand side can be considered either as the *consistency residuum* from the previous time level or the *initial residuum* since the solution of the previous time level is taken as an initial solution on the next time level. Therefore, the iterative process is stopped if the residuum is $1/\delta_k$-times smaller than the initial one.

However, in order to save the computational time, we employ the stopping criterion in the form

$$\|\hat{\boldsymbol{P}}(\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_k - \boldsymbol{d}_k(\boldsymbol{W}_{k-1}))\| \leq \delta_k \|\hat{\boldsymbol{P}}(\boldsymbol{A}_k(\boldsymbol{W}_{k-1})\boldsymbol{W}_{k-1} - \boldsymbol{d}_k(\boldsymbol{W}_{k-1}))\|, \tag{4.70}$$

$k = 1, \dots, r$, since the norm on the left-hand side of (4.70) is available at each GMRES step and thus we need not to evaluate the residuum at the left-hand-side of (4.69). Numerical experiments show that both stopping criteria (4.69) and (4.70) have very similar behavior.

There is still an open question, how to choose the parameter δ_k. In [EW96], there were presented two choices of δ_k and the corresponding orders of convergence of the inexact Newton method were proved. However, our numerical experiments presented in Section 4.4 show that the efficiency of the method only weakly depends on value δ_k chosen around 1/2. Therefore, we put $\delta_k = \delta = 1/2,\ k = 1, \dots, r$ in our algorithm.

Choice of the time step

The strategy of the choice of the time step has a great influence to the efficiency of the discussed method. We already mentioned that the semi-implicit time discretization allows to choose the time step many times larger than an explicit scheme. On the other hand at the beginning of the computation, we usually start from an unphysical initial condition and then too large time step can cause a fail of the computational process. Therefore, our aim is to construct sufficiently robust algorithm which automatically

increase the time step from small values at the beginning of the computation to large values.

In [Dol10a] and [Dol10b] we employed a heuristic strategy which exponentially increases the size of the time step. Although this approach leads to an efficient strategy its drawback is a presence of a user-defined parameter which has no reasonable interpretation.

Standard (more rigorous) ODEs strategies choose the size of the time step in this way that the corresponding *local discretization error* has to be under a given tolerance, see, e.g., [HNW00]. Very often, the local discretization error is estimated by a difference of two numerical solutions obtained by two time integration methods. In [DK08], we employed a combination of two multi-step formulae which gives a rigorous strategy and produces satisfactory results. However, we have to solve two linear algebraic systems at each time level which requires higher computational costs than approach from [Dol10a] and [Dol10b].

In this paper, we present a new strategy which is based on an very low cost estimation of the local discretization error. For simplicity, we deal only with the first order method but these considerations can be simply extended to higher order schemes. Let us consider the ordinary differential equation

$$y' := \frac{\boldsymbol{d}y(t)}{\boldsymbol{d}t} = f(y), \quad y(0) = y_0, \tag{4.71}$$

where $y : [0,T] \to I\!R$, $f : I\!R \to I\!R$ and $y_0 \in I\!R$. We assume that problem (4.71) has a unique solution. Moreover, let $0 = t_0 < t_1 < t_2 < \ldots < t_r = T$ be a partition of $(0,T)$. We denote by $y_k \approx y(t_k)$ an approximation of the solution y at t_k, $k = 1, \ldots, r$. The *backward Euler method* reads

$$y_k = y_{k-1} + \tau_k f(y_k), \qquad k = 1, 2, \ldots, r, \tag{4.72}$$

where $\tau_k := t_k - t_{k-1}$. Using the Taylor theorem, we estimate the corresponding local discretization error L_k by

$$L_k := y(t_k) - y_k \approx \frac{1}{2}\tau_k^2 y''(\theta_k), \qquad \theta_k \in (t_{k-1}, t_k), \tag{4.73}$$

where y'' denotes the second order derivative of y.

Our idea is the following. We define a quadratic function $\tilde{y} : (t_{k-2}, t_k) \to I\!R$ such that $\tilde{y}(t_{k-l}) = y_{k-l}$, $l = 0, 1, 2$. The second order derivative of $\tilde{y}$ is constant on (t_{k-2}, t_k) and hence we put

$$L_k \approx L_k^{\mathrm{app}} := \frac{1}{2}\tau_k^2 \tilde{y}''. \tag{4.74}$$

Let $\omega > 0$ be a given tolerance for the local discretization error. Our aim is to choose the time step as large as possible but $L_k \leq \omega$, $k = 1, \ldots, r$. Using (4.74), we obtain a relation for the optimal size of τ_k by

$$\tau_k^{\mathrm{opt}} = \tau_k \left(\frac{\omega}{L_k^{\mathrm{app}}} \right)^{1/2}. \tag{4.75}$$

Hence, we define
Adaptive time step algorithm

1) let $\omega > 0$, $k > 1$, $y_{k-1} \in I\!R$, $y_{k-2} \in I\!R$ and $\tau_k > 0$ be given,

2) compute y_k by (4.72),

3) from $[t_{k-l}, y_{k-l}]$, $l = 0, 1, 2$ construct $\tilde{y}_k$,

4) compute τ_k^{opt} by (4.74) and (4.75),

5) *if* $\tau_k^{\text{opt}} \geq \tau_k$
then

   i) put $\tau_{k+1} := \min(\tau_k^{\text{opt}},\ c_1\tau_k,\ \tau^{\max})$,
   ii) put $k := k + 1$
   iii) go to step 2)

else

   i) put $\tau_k := \tau_k^{\text{opt}}$,
   ii) go to step 2).

The constant c_1 restricts the maximal ratio of two following time steps, we use the value $c_1 = 2.5$. The value $\tau^{\max}$ restricts the maximal size of the time step from practical reasons. We employ value $\tau^{\max} = 10^6$ but any sufficiently large value yields to similar results.

There is still an open question how to choose the first two time steps. We employ the same strategy as in [Dol10a] where

$$\tau_k = \frac{1}{2} \max_{K \in \mathcal{T}_h} |K|^{-1} \max_{\Gamma \in \partial K} \lambda(\boldsymbol{w}_{h,k}|_\Gamma)|\Gamma|, \quad k = 0, 1, \tag{4.76}$$

where $\lambda(\boldsymbol{w}_{h,k}|_\Gamma)$ is the spectral radius of matrix $\boldsymbol{P}(\boldsymbol{w}_{h,k}|_\Gamma, \vec{n}_\Gamma)$ given by (4.14). Thus τ_0 and τ_1 correspond to the time steps used for the explicit time discretization with CFL = 0.5, see [FFS03]. This approach can be simply extend to a system of ODEs component-wise.

4.3.4 New solution strategy

We summarize the solution strategy for the solution of (4.36). We use the algorithm (AA) where

i) the global loop is stopped if the steady state conditions (4.61) and (4.62) are satisfied,

ii) the size of the time step is chosen by the *Adaptive time step algorithm* presented in Section 4.3.3,

iii) the corresponding linear systems (4.44) are solved by GMRES method with ILU preconditioner (4.66) – (4.68),

iv) at each time level k, the GMRES steps are performed till the stopping criterion (4.70) is satisfied.

4.4 Numerical experiments

In this section, we present a set of numerical experiments in order to

a) choose the parameters δ in (4.70) and ω in (4.75) and demonstrate that the efficiency of the new approach is not sensitive with respect to these choices,

b) demonstrate the efficiency of the method, namely the relative computational times necessary for the setting and the solution of the linear algebraic systems,

c) compare the presented new strategy with the explicit time discretization from [Dol04] and the old semi-implicit time discretization from [Dol08].

Finally, we present two additional numerical experiments demonstrating the robustness of the solution strategy with respect to the flow regime.

4.4.1 Data settings

We consider a laminar viscous subsonic flow around the NACA 0012 profile with inlet Mach number $M_{\text{inlet}} = 0.5$, the angle of attack $\alpha = 2^\circ$ and the Reynolds number $\text{Re} = 5\,000$. We employ the IIPG variant of DGFE method with the penalty parameter $C_W = 200$ and numerical experiments were carried out by P_1, P_2 and P_3 polynomial approximations on four triangular unstructured grids (r2, r3, r4, r5) adaptively refined by the ANGENER code [Dol00]. Figure 4.1 shows details of these grids. We employ the stopping criteria (4.61) with $\text{TOL} = 10^{-4}$ and (4.62) with $\text{tol} = 10^{-4}$.

4.4.2 Dependence of the efficiency on the GMRES stopping criterion (4.70)

Our aim in this section is to find "an optimal" value of the parameter $\delta \in (0, 1)$ from the GMRES stopping criterion (4.70). The optimality is measured in terms of the computational time. Therefore, we carried out computations on grids r2, r3, r4, r5 with the aid of P_1, P_2 and P_3 polynomial approximations for the values $\delta = 0.8$, $\delta = 0.65$, $\delta = 0.5$, $\delta = 0.35$, $\delta = 0.2$ and $\delta = 0.05$. Moreover, the tolerance for the local discretization error in (4.75) is chosen uniquely $\omega = 0.5$ which causes that the size of the time step is increased relatively fast.

Table 4.2 shows the number of time levels necessary to achieve the steady state solution, the corresponding steady-state residuum $\text{SSres}(k)$ given by (4.61), the final values of the drag, lift and momentum (with respect to one fourth of the chord) coefficients and the total computational time in seconds.

We observe that:

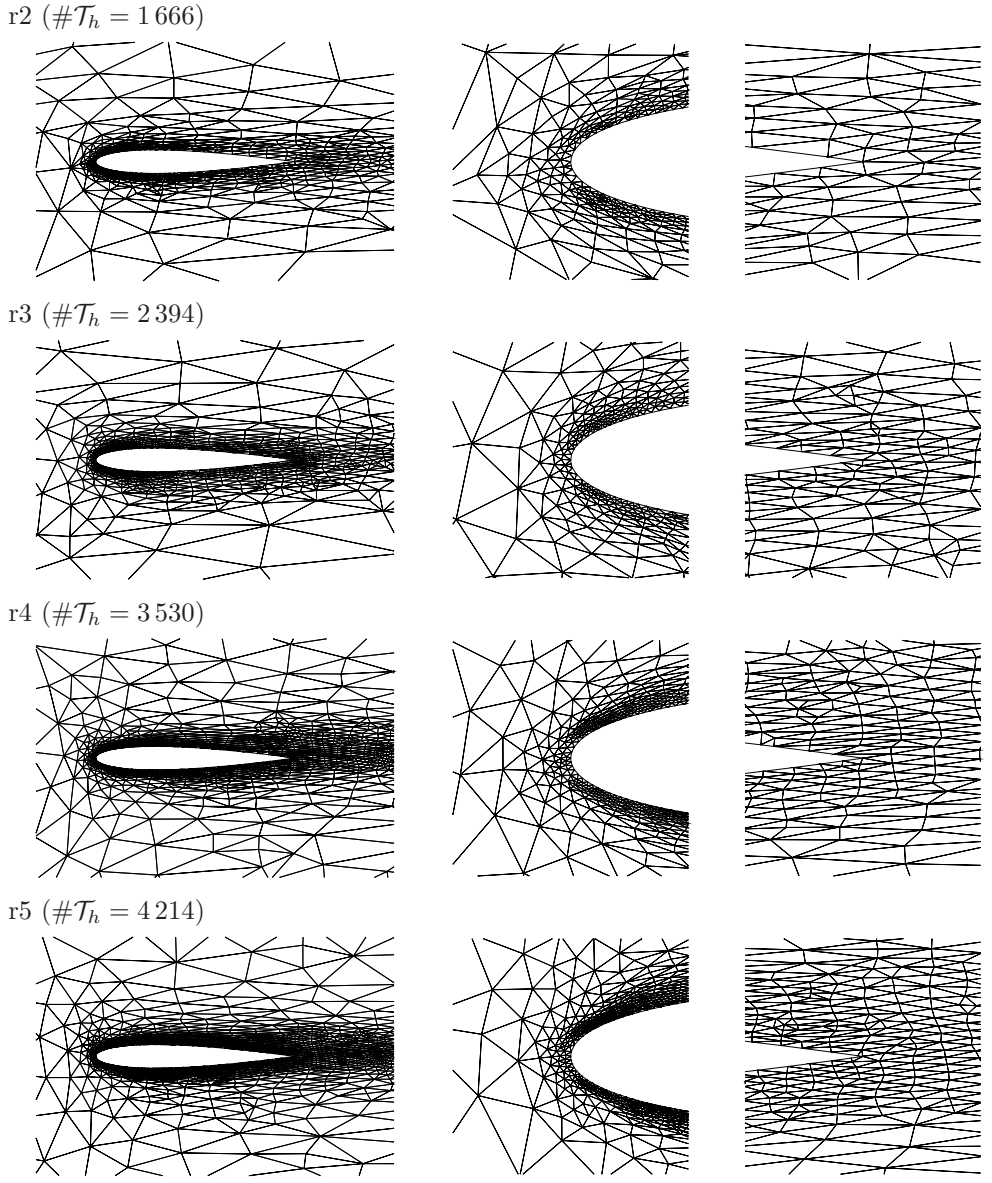


Figure 4.1: The used grids r2, r3, r4, r5, details of the profile (left column), leading edge (middle column) and trailing edge (right column)

mesh	P_k	δ	iter	SSres(k)	c_D	c_L	c_M	CPU(s)
r2	P_1	0.80	212	1.00E-04	0.0589	0.0610	-0.0158	5.90E+01
r2	P_1	0.65	175	9.99E-05	0.0590	0.0606	-0.0158	5.31E+01
r2	P_1	0.50	177	9.99E-05	0.0590	0.0603	-0.0158	5.60E+01
r2	P_1	0.35	1000	5.22E-05	0.0582	0.0645	-0.0156	3.61E+02
r2	P_1	0.20	1000	5.26E-05	0.0582	0.0645	-0.0156	4.49E+02
r2	P_1	0.05	1000	5.28E-05	0.0583	0.0644	-0.0156	7.87E+02
r3	P_1	0.80	300	8.67E-05	0.0572	0.0433	-0.0175	1.23E+02
r3	P_1	0.65	235	8.69E-05	0.0572	0.0431	-0.0176	1.02E+02
r3	P_1	0.50	174	8.81E-05	0.0573	0.0428	-0.0176	8.44E+01
r3	P_1	0.35	123	9.13E-05	0.0573	0.0427	-0.0176	6.29E+01
r3	P_1	0.20	117	9.32E-05	0.0574	0.0424	-0.0176	7.78E+01
r3	P_1	0.05	104	9.92E-05	0.0574	0.0424	-0.0176	1.01E+02
r4	P_1	0.80	366	5.87E-05	0.0557	0.0441	-0.0174	2.55E+02
r4	P_1	0.65	308	6.22E-05	0.0557	0.0439	-0.0174	2.22E+02
r4	P_1	0.50	253	7.60E-05	0.0558	0.0436	-0.0174	2.02E+02
r4	P_1	0.35	224	9.93E-05	0.0558	0.0434	-0.0174	1.91E+02
r4	P_1	0.20	223	9.92E-05	0.0558	0.0432	-0.0175	2.20E+02
r4	P_1	0.05	218	9.99E-05	0.0558	0.0432	-0.0175	4.00E+02
r5	P_1	0.80	574	5.36E-05	0.0576	0.0417	-0.0171	4.67E+02
r5	P_1	0.65	314	5.28E-05	0.0577	0.0409	-0.0171	3.02E+02
r5	P_1	0.50	218	5.17E-05	0.0577	0.0407	-0.0171	2.22E+02
r5	P_1	0.35	167	5.14E-05	0.0577	0.0402	-0.0171	1.99E+02
r5	P_1	0.20	187	5.12E-05	0.0577	0.0402	-0.0171	2.78E+02
r5	P_1	0.05	139	6.19E-05	0.0577	0.0402	-0.0171	3.53E+02
r2	P_2	0.80	224	5.97E-05	0.0563	0.0418	-0.0166	1.93E+02
r2	P_2	0.65	175	5.79E-05	0.0563	0.0418	-0.0166	1.56E+02
r2	P_2	0.50	148	5.65E-05	0.0564	0.0418	-0.0166	1.44E+02
r2	P_2	0.35	125	5.62E-05	0.0564	0.0415	-0.0166	1.28E+02
r2	P_2	0.20	108	5.76E-05	0.0564	0.0415	-0.0166	1.29E+02
r2	P_2	0.05	98	6.18E-05	0.0564	0.0415	-0.0166	1.83E+02
r3	P_2	0.80	250	4.99E-05	0.0562	0.0398	-0.0168	3.12E+02
r3	P_2	0.65	207	5.00E-05	0.0562	0.0399	-0.0169	2.73E+02
r3	P_2	0.50	168	5.07E-05	0.0562	0.0400	-0.0169	2.43E+02
r3	P_2	0.35	119	9.85E-05	0.0562	0.0397	-0.0169	1.74E+02
r3	P_2	0.20	118	9.68E-05	0.0562	0.0397	-0.0169	1.94E+02
r3	P_2	0.05	115	9.83E-05	0.0562	0.0398	-0.0169	2.73E+02
r4	P_2	0.80	317	2.97E-05	0.0559	0.0382	-0.0170	6.15E+02
r4	P_2	0.65	264	4.46E-05	0.0559	0.0383	-0.0170	5.23E+02
r4	P_2	0.50	210	9.94E-05	0.0559	0.0384	-0.0170	4.36E+02
r4	P_2	0.35	212	9.93E-05	0.0559	0.0382	-0.0170	4.70E+02
r4	P_2	0.20	212	9.87E-05	0.0559	0.0382	-0.0170	5.52E+02
r4	P_2	0.05	207	9.92E-05	0.0559	0.0383	-0.0170	7.77E+02
r5	P_2	0.80	268	2.37E-05	0.0564	0.0406	-0.0168	5.95E+02
r5	P_2	0.65	199	2.60E-05	0.0564	0.0404	-0.0168	5.05E+02
r5	P_2	0.50	207	2.90E-05	0.0564	0.0399	-0.0168	5.62E+02
r5	P_2	0.35	213	2.82E-05	0.0564	0.0400	-0.0168	6.51E+02
r5	P_2	0.20	195	4.10E-05	0.0564	0.0401	-0.0168	7.00E+02
r5	P_2	0.05	166	9.62E-05	0.0564	0.0401	-0.0168	8.89E+02
r2	P_3	0.80	243	3.16E-05	0.0556	0.0423	-0.0167	6.40E+02
r2	P_3	0.65	185	3.19E-05	0.0556	0.0424	-0.0167	5.08E+02
r2	P_3	0.50	154	3.33E-05	0.0556	0.0424	-0.0167	4.44E+02
r2	P_3	0.35	135	3.56E-05	0.0556	0.0425	-0.0167	4.29E+02
r2	P_3	0.20	115	4.33E-05	0.0556	0.0422	-0.0167	4.25E+02
r2	P_3	0.05	105	5.88E-05	0.0556	0.0421	-0.0168	6.13E+02
r3	P_3	0.80	206	2.51E-05	0.0558	0.0427	-0.0168	7.88E+02
r3	P_3	0.65	186	2.38E-05	0.0558	0.0425	-0.0168	7.48E+02
r3	P_3	0.50	145	9.55E-05	0.0558	0.0423	-0.0168	6.04E+02
r3	P_3	0.35	144	9.61E-05	0.0558	0.0423	-0.0168	6.35E+02
r3	P_3	0.20	139	9.57E-05	0.0558	0.0421	-0.0168	6.97E+02
r3	P_3	0.05	135	9.81E-05	0.0558	0.0421	-0.0168	9.62E+02
r4	P_3	0.80	270	1.89E-05	0.0557	0.0399	-0.0170	1.57E+03
r4	P_3	0.65	230	5.05E-05	0.0557	0.0399	-0.0170	1.38E+03
r4	P_3	0.50	199	9.97E-05	0.0557	0.0398	-0.0170	1.25E+03
r4	P_3	0.35	198	9.81E-05	0.0557	0.0396	-0.0170	1.35E+03
r4	P_3	0.20	194	9.94E-05	0.0558	0.0397	-0.0169	1.49E+03
r4	P_3	0.05	190	9.78E-05	0.0558	0.0397	-0.0169	2.23E+03
r5	P_3	0.80	280	1.40E-05	0.0559	0.0416	-0.0168	1.98E+03
r5	P_3	0.65	258	3.51E-05	0.0559	0.0412	-0.0168	2.04E+03
r5	P_3	0.50	255	3.59E-05	0.0559	0.0412	-0.0168	2.21E+03
r5	P_3	0.35	238	4.69E-05	0.0559	0.0411	-0.0168	2.25E+03
r5	P_3	0.20	220	6.96E-05	0.0559	0.0411	-0.0168	2.52E+03
r5	P_3	0.05	202	9.83E-05	0.0559	0.0411	-0.0168	3.56E+03

Table 4.2: Dependence of the efficiency of the algorithm on δ from the GMRES stopping criterion (4.70)

– Smaller values of δ generally cause an increase of the computational time since higher number of GMRES steps has to be carried out in order to fulfill the stronger stopping criterion (4.70).

– The large values of δ give shorter computational time and there is no essential difference for δ around 1/2. This is very important property since in practice we have no trouble with the choice of δ, any reasonable large value gives similarly efficient scheme.

– The values of the aerodynamic coefficients do not depend on δ except small exceptions for coarser grids and smaller degree of polynomial approximations. On the other hand, some differences (up to 5 or 10 %) are observed for different grids and different degrees of polynomial approximations.

– Moreover, some numerical experiments (not presented here) show that using higher tolerance ω (e.g., $\omega = 1$) and small tolerance δ (e.g., $\delta = 0.05$), computations carried out on finner grids and higher degrees of polynomial approximations fail. The reason is that the GMRES solver does not achieve the given (small) tolerance.

4.4.3 Dependence of the efficiency on the choice of the time step

In this section we study the dependence of the efficiency of the proposed solution strategy on the local discretization error tolerance ω in (4.75). Again, the optimality is measured in terms of the computational time. In virtue of the similarity of results carried out on different grids and different orders of polynomial approximations, we present only results obtained on the finest grid r5 and for P_3 polynomial approximations. We employ the tolerances $\omega = 2$, $\omega = 1$, $\omega = 0.5$, $\omega = 0.25$, $\omega = 0.125$ and $\omega = 0.0625$. Moreover, the parameter δ in the GMRES stopping criterion (4.70) is chosen uniquely $\delta = 0.5$.

Figure 4.2, left, shows the dependence of the steady-state residuum SSres(k) with respect to the computational time. We observe that smaller values of ω give several times slower convergence than the larger ones. This is caused by the fact that the adaptive time step algorithm increases the size of the time step slowly than for large ω. This is documented by Figure 4.2, right where are shown the time step lengths for different values ω with respect to the computational time. We observe that the limit value $\tau^{\max} = 10^6$ is always achieved but for smaller values of ω it requires longer time. Very important observation is that there is almost negligible difference for the larger values of ω. Therefore, the choice $\omega > 1/2$ gives in fact identical efficiency of the method. Hence, we have no troubles with a choice of some optimal value of ω, any reasonable large value gives similarly efficient scheme.

4.4.4 Efficiency of the new approach

In this section, we demonstrate the efficiency of the new solution strategy. Firstly, we compare the relative computational times necessary for the preparing of the linear

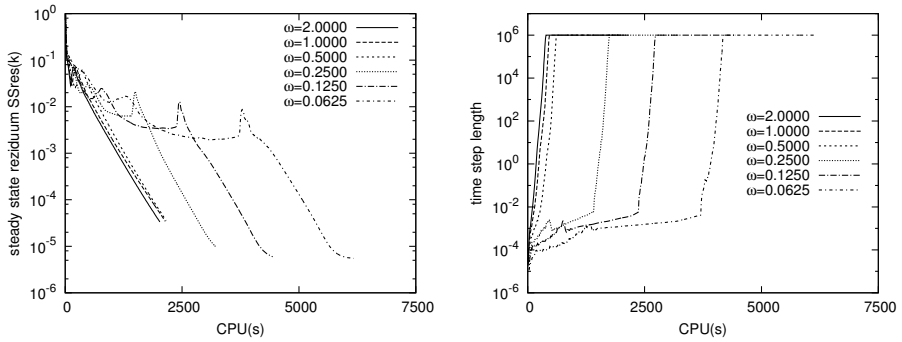


Figure 4.2: The dependence of the steady state residuum SSres(k) (left) and the time step lengths τ_k (rights) on the computational time for values $\omega = 2$, $\omega = 1$, $\omega = 0.5$, $\omega = 0.25$, $\omega = 0.125$ and $\omega = 0.0625$

algebraic systems (i.e., the evaluations of the matrices $\boldsymbol{A}_k$ and the right-hand side $\boldsymbol{d}_k,\ k = 1, \ldots, r$ using (4.44) – (4.48)) and themselves solutions with the aid of this strategy. Based on the numerical experiments presented in Sections 4.4.2 and 4.4.3, we employ the values $\delta = 0.5$ in (4.70) and $\omega = 0.5$ in (4.75). Table 4.3 shows the relative computational times for the preparing of the linear algebraic systems and the solution by the new solution strategy for all grids and all degrees of polynomial approximations. We observe that the solution of the sequence of the linear algebraic systems (4.44) requires less than 50% of the computational time. Therefore, our strategy is in fact optimal since any additional increase of the efficiency of the solution of (4.44) does not cause any essential increase of the efficiency of the global scheme (4.36). An additional acceleration the scheme (4.36) would require also an acceleration of the setting of matrices $\boldsymbol{A}_k,\ k = 1, \ldots, r$, e.g., by a matrix-free implementation. Last three columns show the total computational times in seconds, the computational time per one degree of freedom (dof) in seconds and the average number of GMRES steps performed at each time level.

Moreover, we compare the proposed strategy with the explicit time discretization from [Dol04] and the old semi-implicit time discretization from [Dol08]. We consider the viscous flow with the data setting from Section 4.4.1 and also the limit inviscid flow (Re $\to \infty$) with the same data setting. Table 4.4 shows a comparison of the computational times and the memory requirements of these three approaches for the inviscid and viscous flows using P_1 and P_3 polynomial approximation. We simply observe a significant decrease of the computational times. On the other hand, semi-implicit approaches requires more memory since the matrices are stored in our implementations.

4.4.5 Additional numerical examples

In order to demonstrate the robustness of the presented approach with respect to the flow regime, we present two additional examples from [Mit98].

P_k	$\#\mathcal{T}_h$	dof	preparing $\boldsymbol{A}_k, \boldsymbol{d}_k$	solving $\boldsymbol{A}_k \boldsymbol{W}_k = \boldsymbol{d}_k$	CPU(s)	$\frac{\text{CPU(s)}}{\text{dof}}$	Σ
P_1	1 666	19 992	56%	43%	55.8	2.79E-03	4
P_1	2 394	28 728	57%	42%	81.1	2.82E-03	4
P_1	3 530	42 360	61%	38%	188.7	4.45E-03	6
P_1	4 214	50 568	65%	34%	224.0	4.43E-03	7
P_2	1 666	39 984	56%	43%	139.4	3.49E-03	3
P_2	2 394	57 456	58%	41%	243.3	4.23E-03	3
P_2	3 530	84 720	56%	43%	424.2	5.01E-03	3
P_2	4 214	101 136	62%	37%	571.3	5.65E-03	5
P_3	1 666	66 640	58%	41%	442.2	6.64E-03	3
P_3	2 394	95 760	56%	43%	602.4	6.29E-03	3
P_3	3 530	141 200	59%	40%	1 274.9	9.03E-03	3
P_3	4 214	168 560	63%	36%	2 158.4	1.28E-02	5

Table 4.3: Relative computational times for the preparing of the linear algebraic systems and their solution by the new solution strategy, Σ denotes an average number of GMRES steps per each time level

		P_1		P_3	
case	method	CPU time	memory	CPU time	memory
inviscid	explicit [Dol04]	6 194 s	6 MB	—	41 MB
	semi-implicit [Dol08]	232 s	34 MB	2 283 s	177 MB
	new semi-implicit	47 s	30 MB	226 s	168 MB
viscous	explicit [Dol04]	11 680 s	5 MB	—	38 MB
	semi-implicit [Dol08]	362 s	25 MB	2 292 s	172 MB
	new semi-implicit	97 s	24 MB	613 s	162 MB

Table 4.4: Comparison of the computational times and the memory requirements of the explicit, semi-implicit and new semi-implicit time discretization

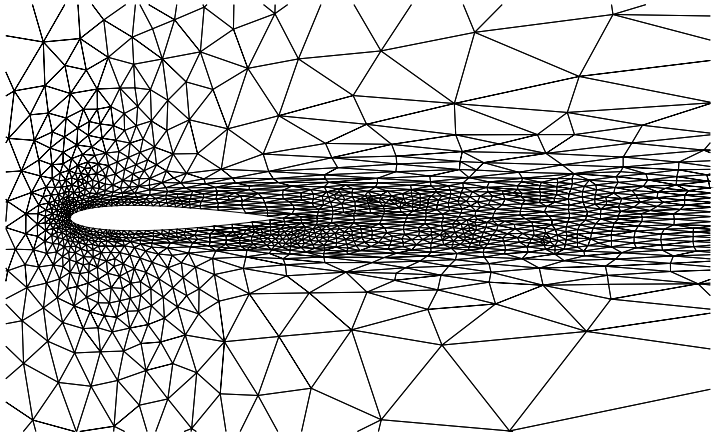


Figure 4.3: NACA 0012, $M_{\text{inlet}} = 0.85$, $\alpha = 0°$ and $\text{Re} = 2\,000$, triangular grid

Transonic viscous flow

We consider a transonic flow around the NACA 0012 profile with the inlet Mach number $M_{\text{inlet}} = 0.85$, angle of attack $\alpha = 0°$ and the Reynolds number $\text{Re} = 2\,000$. We employ a triangular grid from Figure 4.3, P_3 polynomial approximation and the algorithm (AA) with $\delta = 0.5$ in (4.70) and $\omega = 0.5$ in (4.75), i.e., the same values as in Section 4.4.4.

We observe that the computational time for the solution of the corresponding linear algebraic systems (4.44) is shorter than the computational time for their preparation. Moreover, approximately 5 GMRES steps is performed at each time level. Therefore, the efficiency of the solution strategy (AA) is high similarly as for the subsonic flow presented in Section 4.4.1. Figure 4.4 shows the corresponding isolines of the Mach number and the pressure.

Unsteady viscous flow

We consider a transonic flow around the NACA 0012 profile with the inlet Mach number $M_{\text{inlet}} = 0.85$, angle of attack $\alpha = 0°$ and the Reynolds number $\text{Re} = 10\,000$. This problem is more challenging since the flow is unsteady with a periodic propagation of vortexes behind the profiles, see [Mit98]. As we already mention, our solution strategy, originally developed for steady flows, can be used without any modification also for unsteady flows. We employ again the value $\delta = 0.5$ in the stopping criterion (4.70).

On the other hand, the problem is unsteady then it is necessary to choose the time step smaller in order to guarantee an accuracy with respect to the time. Obviously, it is more efficient to use a higher order scheme with respect to time (e.g., BDF formulae as in [Dol08]). This will be a subject of further research. Here, we only demonstrate the capability of our approach to solve unsteady problems. Therefore, we use the adaptive (first order) time step algorithm from Section 4.3.3 and put the tolerance $\omega = 0.01$ in

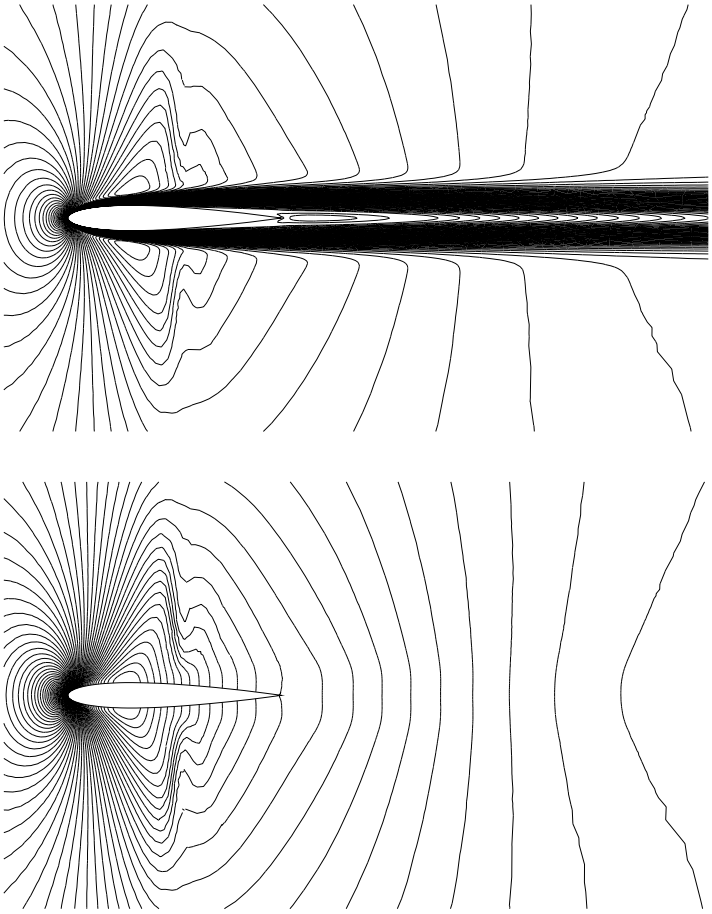


Figure 4.4: NACA 0012, $M_{\text{inlet}} = 0.85$, $\alpha = 0°$ and $\text{Re} = 2\,000$, isolines of Mach number (top) and isolines of pressure (bottom)

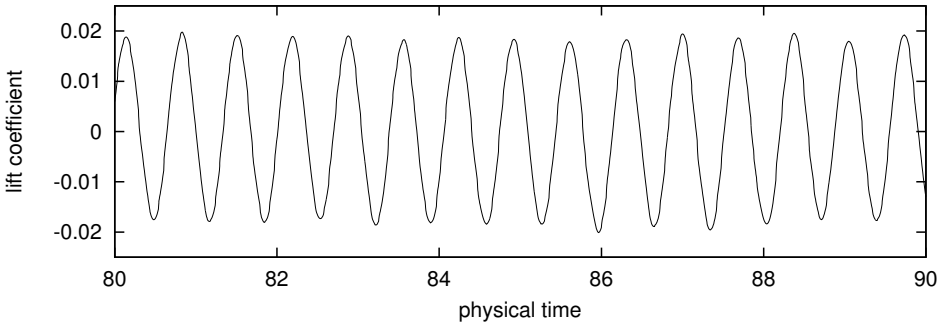
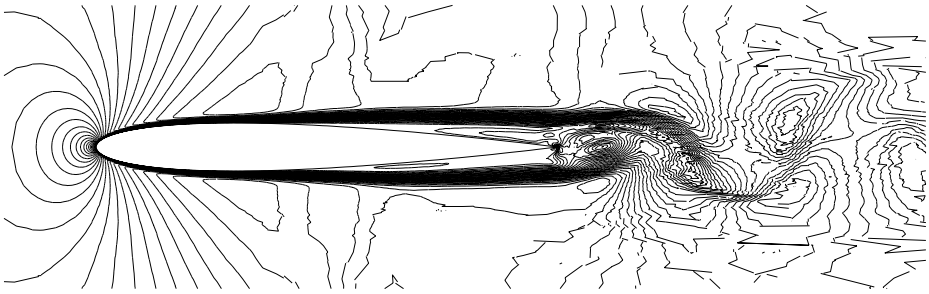


Figure 4.5: NACA 0012, $M_{\text{inlet}} = 0.85$, $\alpha = 0°$ and $\text{Re} = 10\,000$, dependence of the lift coefficient c_L on time t during $t = (80, 90)$

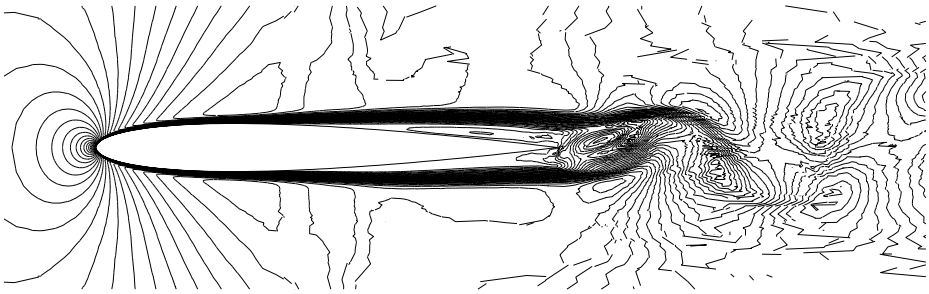
(4.75).

We employed a grid similar to grid from Figure 4.3 and P_2 polynomial approximation. We carried out a computation for the physical (dimensionless) time $t \in (0, 90)$. Figure 4.5 shows the dependence of the lift coefficient c_L on time for $t \in (80, 90)$ (in order to see better details). We observe a periodic oscillations with the period approximately equal to $\Delta t = 0.7$. Figures 4.6 and 4.7 show the Mach number isolines at times $t_i = 89.4 + i/7\Delta t,\ i = 1, 2, \dots 7$ demonstrating the periodic propagation of vortexes behind profile. These results are in good agreement with the results from [Mit98] and [Dol08] and moreover, the computational time was shorter in comparison with [Dol08].

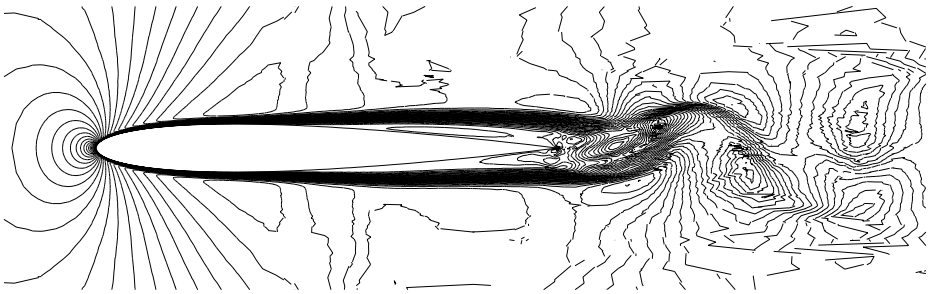
It may be surprising that the "weak" stopping condition (4.70) with $\delta = 0.5$ works also for unsteady flows, particularly any lost of the accuracy was not observed. This is caused by the fact that τ_k is significantly smaller than for steady-state problems and therefore there is a small difference between $\boldsymbol{W}_{k-1}$ and $\boldsymbol{W}_k$. Hence, the initial residuum at the right-hand side of (4.70) is already small.



time at 1 / 7 of period



time at 2 / 7 of period



time at 3 / 7 of period

Figure 4.6: NACA 0012, $M_{\text{inlet}} = 0.85$, $\alpha = 0°$ and $\text{Re} = 10\,000$, the isolines of Mach number at 1/7, 2/7 and 3/7 of one period

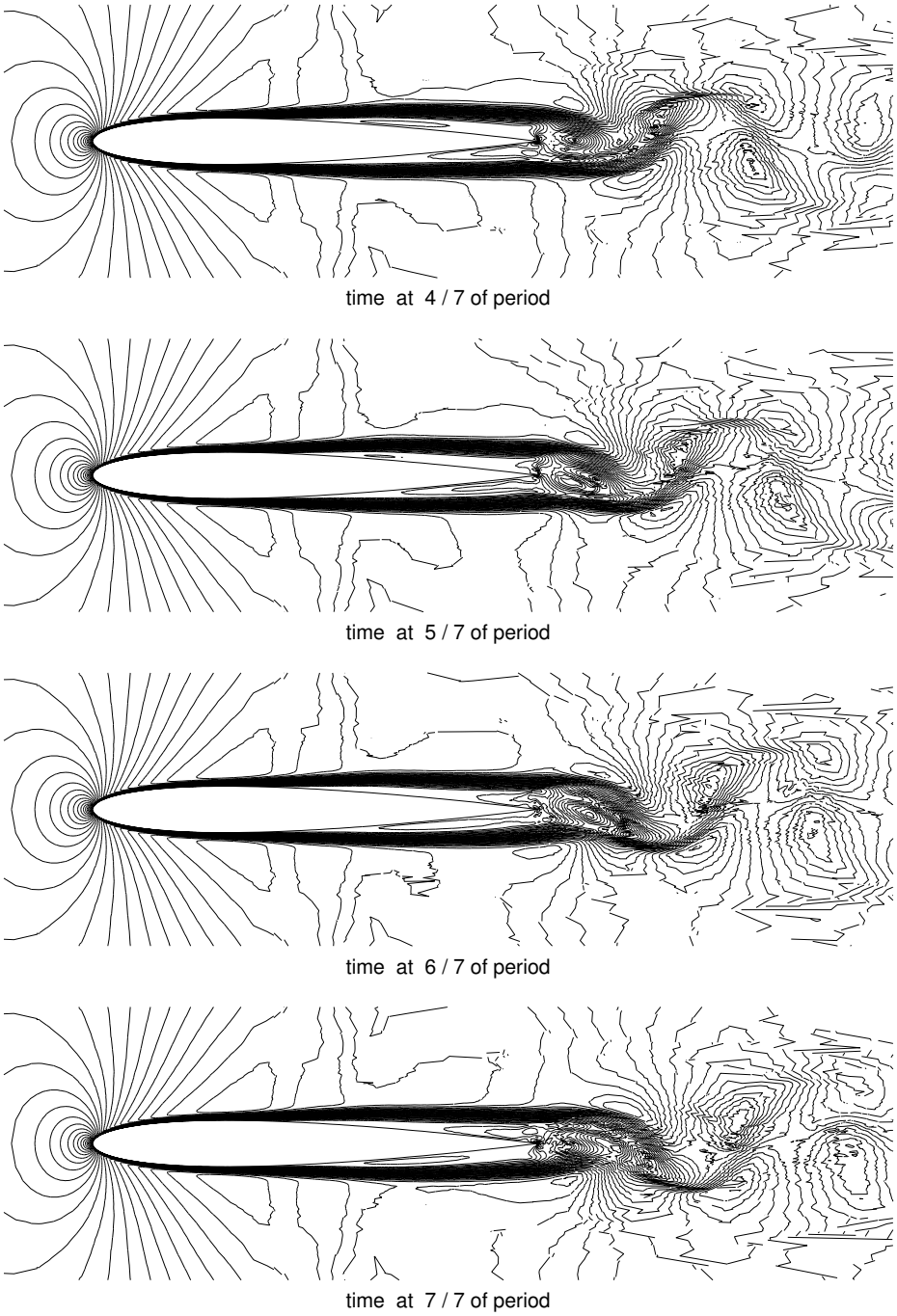


Figure 4.7: NACA 0012, $M_{\text{inlet}} = 0.85$, $\alpha = 0°$ and $\text{Re} = 10\,000$, the isolines of Mach number at 4/7, 5/7, 6/7 and 7/7 of one period

Bibliography

[ABCM02] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.

[Arn82] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[BCRS05] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds averaged Navier-Stokes and k-ω turbulence model equations. *Comput. Fluids*, 34:507–540, 2005.

[BO99a] C. E. Baumann and J. T. Oden. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 31(1):79–95, 1999.

[BO99b] C. E. Baumann and J. T. Oden. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 31:79–95, 1999.

[BR97] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *J. Comput. Phys.*, 131:267–279, 1997.

[BR00] F. Bassi and S. Rebay. A high order discontinuous Galerkin method for compressible turbulent flow. In B. Cockburn, G. E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Method: Theory, Computations and Applications*, Lecture Notes in Computational Science and Engineering 11, pages 113–123. Springer-Verlag, 2000.

[Bre03] S. C. Brenner. Poincare-Friedrichs inequalities for piecewise H-1 functions. *SIAM Journal on Numerical Analysis*, 41(1):306–324, 2003.

[BS90] I. Babuška and M. Suri. The p- and h-p versions of the finite element method. an overview. *Comput. Methods Appl. Mech. Eng.*, 80:5–26, 1990.

[BS94] S. Brenner and R. L. Scott. *The Mathematical Theory of Finite Element Methods*. Spriger, New York, 1994.

[BS01] I. Babuška and T. Strouboulis. *The Finite Element Method and its Reliability,*. Clarendon Press, Oxford, 2001.

[Cia79] P. G. Ciarlet. *The Finite Elements Method for Elliptic Problems*. North-Holland, Amsterdam, New York, Oxford, 1979.

[CKS00] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, editors. *Discontinuous Galerkin Methods*. Springer, Berlin, 2000.

[Coc99] B. Cockburn. Discontinuous Galerkin methods for convection dominated problems. In T. J. Barth and H. Deconinck, editors, *High–Order Methods for Computational Physics*, Lecture Notes in Computational Science and Engineering 9, pages 69–224. Springer, Berlin, 1999.

[DD09] L. T. Diosady and D. L. Darmofal. Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations. *J. Comput. Phys.*, 228:3917–3935, 2009.

[DES82] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[DF04] V. Dolejší and M. Feistauer. Semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comput. Phys.*, 198(2):727–746, 2004.

[DF05] V. Dolejší and M. Feistauer. Error estimates of the discontinuous Galerkin method for nonlinear nonstationary convection-diffusion problems. *Numer. Funct. Anal. Optim.*, 26(3):349–383, 2005.

[DFH07] V. Dolejší, M. Feistauer, and J. Hozman. Analysis of semi-implicit DGFEM for nonlinear convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 196:2813–2827, 2007.

[DFKS08] V. Dolejší, M. Feistauer, V. Kučera, and V. Sobotíková. An optimal $L^\infty(L^2)$-error estimate of the discontinuous galerkin method for a nonlinear nonstationary convection-diffusion problem. *IMA J. Numer. Anal.*, 28(3):496–521, 2008.

[DFS05] V. Dolejší, M. Feistauer, and V. Sobotíková. Analysis of the discontinuous galerkin method for nonlinear convectiondiffusion problems. *Comput. Methods Appl. Mech. Eng.*, 194:2709–2733, 2005.

[DK08] V. Dolejší and P. Kůs. Adaptive backward difference formula – discontinuous Galerkin finite element method for the solution of conservation laws. *Int. J. Numer. Methods Eng.*, 73(12):1739–1766, 2008.

[DM06] M. Dumbser and C.-D. Munz. Building blocks for arbitrary high-order discontinuous Galerkin methods. *J. Sci. Comput.*, 27:215–230, 2006.

[Dol00] V. Dolejší. *ANGENER – software package.* Charles University Prague, Faculty of Mathematics and Physics, version 3.0 edition, 2000. webpage: http://www.karlin.mff.cuni.cz/dolejsi/angen.html.

[Dol04] V. Dolejší. On the discontinuous Galerkin method for the numerical solution of the Navier–Stokes equations. *Int. J. Numer. Methods Fluids*, 45:1083–1106, 2004.

[Dol06] V. Dolejší. Discontinuous Galerkin method for the numerical simulation of unsteady compressible flow. *WSEAS Transactions on Systems*, 5(5):1083–1090, 2006.

[Dol08] V. Dolejší. Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. *Commun. Comput. Phys.*, 4(2):231–274, 2008.

[Dol10a] V. Dolejší. On the solution of linear algebraic systems arising from the semi-implicit DGFE discretization of the compressible Navier-Stokes equations. *Kybernetika*, 2010. (in press).

[Dol10b] V. Dolejší. Semi-implicit DGFE discretization of the compressible Navier-Stokes equations: efficient solution strategy. In *Numerical Mathematics and Advanced Applications, ENUMATH 2009*. Springer-Verlag Berlin Heidelberg, 2010. (in press).

[DSW04] C. N. Dawson, S. Sun, and M. F. Wheeler. Compatible algorithms for coupled flow and transport. *Comput. Meth. Appl. Mech. Engng.*, 193:2565–2580., 2004.

[DV08] V. Dolejší and M. Vlasák. Analysis of a BDF – DGFE scheme for nonlinear convection-diffusion problems. *Numer. Math.*, 110:405–447, 2008.

[EW96] S. C. Eisenstat and H.F. Walker. Choosing the forcing terms in inexact newton method. *SIAM J. Sci. Comput.*, 17(1):16–32, 1996.

[FDK07] M. Feistauer, V. Dolejší, and V. Kučera. On the discontinuous Galerkin method for the simulation of compressible flow with wide range of mach numbers. *Comput Visual Sci*, 10:17–27, 2007.

[Fei89] M. Feistauer. On the finite element approximation of functions with noninteger derivatives. *Numer. Funct. Anal. and Optimiz.*, 10(91-110), 1989.

[Fei93] M. Feistauer. *Mathematical Methods in Fluid Dynamics.* Longman Scientific & Technical, Harlow, 1993.

[FFS03] M. Feistauer, J. Felcman, and I. Straškraba. *Mathematical and Computational Methods for Compressible Flow.* Oxford University Press, Oxford, 2003.

[FK07] M. Feistauer and V. Kučera. On a robust discontinuous galerkin technique for the solution of compressible flow. *J. Comput. Phys.*, 224(1):208–221, 2007.

[Har06] R. Hartmann. Adaptive discontinuous galerkin methods with shock-capturing for the compressible navier-stokes equations. *Int. J. Numer. Meth. Fluids*, 51:1131–1156, 2006.

[HH02] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183(2):508–532, 2002.

[HH06a] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: Method formulation. *Int. J. Numer. Anal. Model.*, 1:1–20, 2006.

[HH06b] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations II: Goal-oriented a posteriori error estimation. *Int. J. Numer. Anal. Model.*, 3:141–162, 2006.

[HNW00] E. Hairer, S. P. Norsett, and G. Wanner. *Solving ordinary differential equations I, Nonstiff problems*. Number 8 in Springer Series in Computational Mathematics. Springer Verlag, 2000.

[Joh88] C. Johnson. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, Cambridge, 1988.

[KJk77] A. Kufner, O. John, and S. Fučík. *Function Spaces*. Academia, Prague, 1977.

[KvdVdV06a] C. M. Klaij, J.J.W. van der Vegt, and H. Van der Ven. Pseudo-time stepping for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 219(2):622–643, 2006.

[KvdVdV06b] C. M. Klaij, J.J.W. van der Vegt, and H. Van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 217(2):589–611, 2006.

[LBK98] I. Lomtev, C. B.Quillen, and G. E. Karniadakis. Spectral/*hp* methods for viscous compressible flows on unstructured 2d meshes. *J. Comput. Phys.*, 144(2):325–357, 1998.

[Lio96] P. L. Lions. *Mathematical Topics in Fluid Mechanics*. Oxford Science Publications, 1996.

[Mit98] S. Mittal. Finite element computation of unsteady viscous compressible flows. *Comput. Methods Appl. Mech. Eng.*, 157:151–175, 1998.

[QSS00] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, 2000.

[Rek82] K. Rektorys. *The Method of Discretization in Time and Partial Differential Equations*. Reidel, Dodrecht, 1982.

[RWG99] B. Rivière, M. F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. I. *Comput. Geosci.*, 3(3-4):337–360, 1999.

[Sch00] C. Schwab. Discontinuous Galerkin method. Technical report, ETH Zürich, 2000.

[SS86] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[vdVvdV02a] J. J. W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I: General formulation. *J. Comput. Phys.*, 182(2):546–585, 2002.

[vdVvdV02b] H. van der Ven and J. J. W. van der Vegt. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows II. efficient flux quadrature. *Comput. Methods Appl. Mech. Engrg.*, 191:4747–4780, 2002.

[Vij86] G. Vijayasundaram. Transonic flow simulation using upstream centered scheme of Godunov type in finite elements. *J. Comput. Phys.*, 63:416–433, 1986.

[Wat02] D. S. Watkins. *Fundamentals of Matrix Computations*. Pure and Applied Mathematics, Wiley-Interscience Series of Texts, Monographs, and Tracts. A John Wiley & Sons, Inc., Publications, 2002.

[Wes01] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, Berlin, 2001.

[Žen90] A . Ženíšek. *Nonlinear Elliptic and Evolution Problems and Their Finite Element Approximations*. Academic Press, London, 1990.

SDE 2010
XXVII Seminar in Differential Equations
Volume I

Editor: Gabriela Holubová, Petr Nečesal
Publisher: University of West Bohemia, Univerzitní 8, 306 14, Pilsen
Printing: TYPOS, tiskařské závody, spol. s r.o., Pilsen

1. edition

ISBN 978-80-261-0168-0