INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:1131–1156
Published online 28 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1134

Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations

R. Hartmann[*,†]

Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

SUMMARY

We present the Interior Penalty discontinuous Galerkin method for the compressible Navier–Stokes equations. Shock-capturing is used to reduce over-shoots at discontinuities and sharp gradients. This stabilization introduces artificial viscosity at places of large local residuals, but preserves conservation and Galerkin orthogonality of the DG method. Based on this discretization we derive *a posteriori* error estimates for the error measured in terms of arbitrary target functionals, like, e.g. the drag and lift coefficients of an airfoil immersed in a viscous or inviscid fluid.

The performance of the nonlinear solution process, the *a posteriori* error estimation and an adaptive mesh refinement specially tailored for the accurate computation of the force coefficients are demonstrated for supersonic laminar flows around the NACA0012 airfoil. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: finite element methods; discontinuous Galerkin methods; compressible Navier–Stokes equations; shock-capturing

1. INTRODUCTION

In recent years discontinuous Galerkin methods have experienced a resurgence of interest in multivarious disciplines of numerical mathematics including compressible flows and aerodynamics [1–9], among many others. It can be observed that to an increasing extent discontinuous Galerkin methods are now applied to problems which traditionally were solved using finite volume methods. The reason for this trend can be identified in several advantages of the discontinuous Galerkin methods over finite volume methods. Second-order finite volume methods are achieved by employing a second-order accurate reconstruction. The extension of a second-order finite volume scheme to a (theoretically) third-order scheme requires a

[*] Correspondence to: R. Hartmann, Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany.
[†] E-mail: ralf.hartmann@dlr.de

Received 2 June 2005
Revised 4 October 2005
Accepted 20 October 2005

third-order accurate reconstruction which on unstructured meshes is very cumbersome and which in practice shows deterioration of order. On unstructured meshes finite volume methods of even higher order are virtually impossible. These difficulties bound the order of numerical computations in industrial applications to second-order. In contrast to this, the order of discontinuous Galerkin methods, applied to problems with regular solutions, depends on the degree of the approximating polynomials only which can easily be increased, dramatically simplifying the use of higher order methods on unstructured meshes. Furthermore, the stencil of most discontinuous Galerkin schemes is minimal in the sense that each element communicates only with its direct neighbours. In contrast to the increasing number of elements or mesh points communicating for increasing accuracy of finite volume methods, the inter-element communication of discontinuous Galerkin methods is the same for any order.

The compactness of the discontinuous Galerkin method has clear advantages in parallelization, which does not require additional element layers at partition boundaries. Also due to simple communication at element interfaces, elements with the so-called ‘hanging nodes’ can be treated just as easily as elements without hanging nodes, a fact that simplifies local mesh refinement (h-refinement). In addition to this, the communication at element interfaces is identical for any order of the method which simplifies the use of methods of differing orders in adjacent elements. This allows for the variation of the order of the numerical scheme over the computational domain, which in combination with h-refinement leads to the so-called hp-refinement, where p-refinement denotes the variation of the polynomial degree p which represents a numerical method of the formal order $p+1$ in the L_2 norm.

1.1. Shock-capturing

Discontinuous Galerkin schemes exhibit an inherent stability at discontinuities as can be seen, for example, when solving a linear advection equation with discontinuous boundary values imposed, or when solving compressible Navier–Stokes equations for problems including weak shocks on sufficiently coarse meshes. Indeed, these problems can be discretized and solved without any stabilization applied, although the resulting discrete solutions might suffer from spurious oscillations near the discontinuities. When these oscillations are to be suppressed, when nonlinear discontinuities as, e.g. shocks are strong enough or when computing on sufficiently fine meshes, the discontinuous Galerkin discretizations must be stabilized.

Several stabilization techniques have been proposed: two of the most frequently used are (i) (generalized) limiters as, e.g. proposed by Cockburn and Shu, e.g. Reference [10], in the framework of Runge–Kutta discontinuous Galerkin schemes, and (ii) the addition of artificial viscosity terms, also referred to as shock-capturing (or discontinuity-capturing) terms.

As discussed in Reference [9], local projection or slope limiters, such as the one proposed by Cockburn and Shu [10] have some disadvantages: one is attributed to serious problems which might occur when iterating a stationary solution to steady state. As the limited solution does not satisfy the steady-state discontinuous Galerkin equations, it is not possible to reduce the residual to machine accuracy, see Reference [9]. Instead, the scheme tries to converge to the unlimited solution, which suffers from numerical oscillations and the limiter must remain active to prevent this.

An alternative stabilization approach is the use of artificial viscosity terms which were originally introduced by Hughes and Johnson in the context of SUPG and SD finite element methods and later also in DG methods for scalar hyperbolic conservation laws, see

Reference [11]. This approach adds an artificial viscosity term of the form

$$\sum_{\kappa} \int_{\kappa} \varepsilon \nabla \mathbf{u}_h : \nabla \mathbf{v}_h \, \mathrm{d}\mathbf{x} \tag{1}$$

to the discretization scheme, where κ are the elements which cover the domain Ω and $\mathbf{u}_h$ and $\mathbf{v}_h$ are discrete ansatz and test functions taken from a finite element space $\mathbf{V}_h$. Most of these approaches differ in the specific choice of the coefficient ε, only. Several examples which have been employed for stabilizing discontinuous Galerkin discretizations of the stationary compressible Euler equations (for notation, see Section 2)

$$\nabla \cdot \mathscr{F}^c(\mathbf{u}) = 0 \tag{2}$$

near shocks will be given in the following.

Already in Reference [12], Bassi and Rebay employed an artificial viscosity term of the form (1), where the scalar coefficient ε depends on the residual of the finite element solution $\mathbf{u}_h$ and on the diameter h_κ of the element κ, namely

$$\varepsilon|_\kappa = C_\varepsilon h_\kappa^2 \left\{ \sum_i ((|u_h^i| + c)^{-1} [\nabla \cdot \mathscr{F}^c(\mathbf{u}_h)]_i)^2 \right\}^{-1/2} \tag{3}$$

where C_ε and c are positive parameters and i runs over all components of $\mathbf{u}$. Later, Bassi and Rebay [13], used a viscosity term which depends on the face residuals instead of on the element residuals with the scalar coefficient ε given by

$$\varepsilon|_\kappa = C_\varepsilon \left\{ \sum_i \left[\frac{\int_{\partial\kappa} \mathscr{H}_i - \mathscr{F}_i^c(\mathbf{u}_h) \cdot \mathbf{n}_\kappa \, \mathrm{d}s}{\left(\int_\kappa u_h^i \, \mathrm{d}\mathbf{x}\right) / |\kappa|} \right]^2 \right\}^{-1/2} \tag{4}$$

where $\mathbf{n}_\kappa$ denotes the unit outward normal vector to the boundary $\partial\kappa$ and $\mathscr{H}$ is a numerical flux function approximating the flux, $\mathscr{F}^c(\mathbf{u}_h) \cdot \mathbf{n}_\kappa$, at element interfaces taking into account possible discontinuities of $\mathbf{u}_h$ across the interfaces, see Section 3 for more detail.

Also, Baumann and Oden [4], employed an artificial viscosity term like in (1), with a scalar coefficient given by

$$\varepsilon|_\kappa = h_\kappa \lambda_\kappa(\mathbf{u}_h) \tag{5}$$

on elements κ close to sharp gradients, only, and zero elsewhere. Here, $\lambda_\kappa(\mathbf{u}_h)$ is the maximum characteristic speed, taken as $c + |\mathbf{v}|$, where c is the speed of sound and $\mathbf{v}$ the velocity vector.

Finally, also in the framework of space–time discontinuous Galerkin schemes [9], artificial viscosity stabilization terms have been used, one of which is closely related to (3).

In the current publication we employ an artificial viscosity term (1), with a coefficient similar to

$$\varepsilon|_\kappa = C_\varepsilon h_\kappa^{2-\beta} |\mathbf{R}(\mathbf{u}_h)| \tag{6}$$

with C and β positive constants and $\mathbf{R}(\mathbf{u}) = -\nabla \cdot \mathscr{F}^c(\mathbf{u})$. This is a slight simplification of the artificial viscosity term proposed in Reference [11] and has already been successfully applied in Reference [7] to the adaptive discontinuous Galerkin discretization for the solution of the compressible Euler equations. In fact, in the current publication not the isotropic version (6) is proposed, but a generalization of it for anisotropic meshes.

Like all artificial viscosity terms of the form (1) this stabilization leaves the conservativity property of the discontinuous Galerkin method unchanged. The coefficient ε in (6) is relatively simple in comparison to the coefficients in (3) and (4), which is useful when the scheme is to be treated implicitly. Furthermore, the coefficient $\varepsilon = \varepsilon(\mathbf{u})$ in (6) is continuous with respect to its argument which helps in the nonlinear solution iteration procedure. In particular, it does not include a switch which locally enables or disables the shock-capturing due to, e.g. sharp gradients as does the method proposed in Reference [3]. Finally, the artificial viscosity term (1) with the specific choice of the ε in (6) is consistent in the sense that it vanishes when evaluated for the exact and sufficiently regular solution $\mathbf{u}$ of Equation (2). This results in an artificial viscosity acting only in non-smooth parts of the solution where the residuals are large, and almost vanishing in smooth parts of the solution where the residuals are significantly smaller. Finally, the consistency of the artificial viscosity term ensures that the (local and global) Galerkin orthogonality of the discontinuous Galerkin scheme still holds after its addition. This is particularly important in the framework of *a posteriori* error estimation and adaptivity.

1.2. A posteriori error estimation and adaptivity

In aerodynamical computations like compressible flows around airfoils, much emphasis is placed on the accurate approximation of specific target quantities, in particular, the force coefficients, like the pressure induced as well as the viscous stress induced drag, lift and moment coefficients, respectively. While local mesh refinement is required for obtaining reasonably accurate results in applications, the goal of the adaptive refinement is either to compute these coefficients as accurate as possible within given computing resources or to compute these coefficients up to a given tolerance with the minimum computing resources required. In both cases a goal-oriented refinement is needed, i.e. an adaptive refinement strategy specifically targeted to the efficient computation of the quantities of interest. Futhermore, in the latter case, an estimate is required of how accurate the force coefficients are approximated, i.e. an *a posteriori* error estimate is required of the error of the numerical solution measured in terms of the quantity of interest.

This error can be represented by the element and face residuals of the primal (flow) solution multiplied by the solution of a dual (adjoint) problem with data coupling to the specific target quantity. By approximating the solution to the dual problem numerically, the resulting approximate error representation gives an estimate of the true error. Furthermore, the approximate error representation can be decomposed as a sum over all elements of the so-called dual-weighted residual indicators which can be used for goal-oriented (adjoint-based) refinement specifically tailored to the efficient computation of the quantities of interest.

The approach of *a posteriori* error estimation and adaptivity in finite element methods has been developed in Reference [14] and applied to various kinds of problems, see the survey article [15].

In Reference [16], this approach has been developed for the discontinuous Galerkin discretization of scalar hyperbolic problems. Then, in the series of publications [7, 17, 18], it has been extended to the two-dimensional compressible Euler equations, where a variety of problems have been considered, including the Ringleb flow problem, supersonic flow past a wedge, inviscid flows through a nozzle, and inviscid sub-, trans- and supersonic flows around different airfoil geometries; finally, in References [6, 8], this approach has been extended to the two-dimensional compressible Navier–Stokes equations and applied to subsonic viscous compressible flows around simple airfoil geometries. The current work now gives the extension of this approach to viscous compressible flows including shocks, as for supersonic flows, for example.

The paper is structured as follows: After introducing, in Section 2, the compressible Navier–Stokes equations, in Section 3 we state its discontinuous Galerkin finite element approximation and formulate the consistent and anisotropic artificial viscosity term in Section 4. In Section 5, we propose a damped Newton–GMRES algorithm for the solution of the system of nonlinear equations. After outlining the approach adopted for the *a posteriori* error estimation and an adaptive mesh refinement algorithm specially tailored to the accurate computation of the force coefficients in Section 6, we demonstrate the performance of the proposed methods for supersonic laminar flows around the NACA0012 airfoil in Section 8, including a discussion of linearization and discretization errors of the dual (adjoint) problem introduced.

2. THE COMPRESSIBLE NAVIER–STOKES EQUATIONS

We consider the two-dimensional steady state compressible Navier–Stokes equations. Writing ρ, $\mathbf{v}=(v_1,v_2)^{\mathrm{T}}$, p, E and T to denote the density, velocity vector, pressure, specific total energy and temperature, respectively, the equations of motion are given by

$$\nabla\cdot(\mathscr{F}^c(\mathbf{u})-\mathscr{F}^v(\mathbf{u},\nabla\mathbf{u}))\equiv\frac{\partial}{\partial x_i}\mathbf{f}_i^c(\mathbf{u})-\frac{\partial}{\partial x_i}\mathbf{f}_i^v(\mathbf{u},\nabla\mathbf{u})=0\quad\text{in }\Omega \tag{7}$$

where Ω is an open bounded domain in $\mathbb{R}^2$; here, and throughout the rest of this article, we use the summation convention, i.e. repeated indices are summed through their range. The vector of conservative variables $\mathbf{u}$ and the convective fluxes $\mathbf{f}_i^c$, $i=1,2$, are defined by

$$\mathbf{u}=\begin{bmatrix}\rho\\ \rho v_1\\ \rho v_2\\ \rho E\end{bmatrix},\quad \mathbf{f}_1^c(\mathbf{u})=\begin{bmatrix}\rho v_1\\ \rho v_1^2+p\\ \rho v_1 v_2\\ \rho H v_1\end{bmatrix}\quad\text{and}\quad \mathbf{f}_2^c(\mathbf{u})=\begin{bmatrix}\rho v_2\\ \rho v_1 v_2\\ \rho v_2^2+p\\ \rho H v_2\end{bmatrix} \tag{8}$$

respectively, and the viscous fluxes $\mathbf{f}_i^v$, $i=1,2$, are defined by

$$\mathbf{f}_1^v(\mathbf{u},\nabla\mathbf{u})=\begin{bmatrix}0\\ \tau_{11}\\ \tau_{21}\\ \tau_{1j}v_j+\mathscr{K}T_{x_1}\end{bmatrix}\quad\text{and}\quad \mathbf{f}_2^v(\mathbf{u},\nabla\mathbf{u})=\begin{bmatrix}0\\ \tau_{12}\\ \tau_{22}\\ \tau_{2j}v_j+\mathscr{K}T_{x_2}\end{bmatrix} \tag{9}$$

respectively, where $\mathscr{K}$ is the thermal conductivity coefficient. Additionally, H is the total enthalpy given by

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

where e is the specific static internal energy, and the pressure is determined by the equation of state of an ideal gas

$$p = (\gamma - 1)\rho e \tag{10}$$

where $\gamma = c_p/c_v$ is the ratio of specific heat capacities at constant pressure, c_p, and constant volume, c_v; for dry air, $\gamma = 1.4$. Finally, the viscous stress tensor is defined by

$$\tau = \mu(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}} - \tfrac{2}{3}(\nabla\cdot\mathbf{v})I) \tag{11}$$

where μ is the dynamic viscosity coefficient, and the temperature T is given by $e = c_v T$; thus

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

where $Pr = 0.72$ is the Prandtl number.

For the purposes of discretization, we rewrite the compressible Navier–Stokes equations (7) in the following (equivalent) form:

$$\frac{\partial}{\partial x_i}\left(\mathbf{f}_i^c(\mathbf{u}) - G_{ij}(\mathbf{u})\frac{\partial\mathbf{u}}{\partial x_j}\right) = 0 \quad \text{in } \Omega \tag{12}$$

Here, the matrices $G_{ij}(\mathbf{u}) = \partial\mathbf{f}_i^v(\mathbf{u},\nabla\mathbf{u})/\partial\mathbf{u}_{x_j}$, for $i,j = 1,2$, i.e. $\mathbf{f}_i^v(\mathbf{u},\nabla\mathbf{u}) = G_{ij}(\mathbf{u})\partial\mathbf{u}/\partial x_j$, $i = 1,2$, cf. Reference [8].

Given that $\Omega \subset \mathbb{R}^2$ is a bounded region, with boundary Γ, the system of conservation laws (12) must be supplemented by appropriate boundary conditions. For simplicity of presentation, we assume that Γ may be decomposed as follows:

$$\Gamma = \Gamma_{\mathrm{D,sup}} \cup \Gamma_{\mathrm{D,sub\text{-}in}} \cup \Gamma_{\mathrm{D,sub\text{-}out}} \cup \Gamma_{\mathrm{N}} \cup \Gamma_{\mathrm{W}}$$

where $\Gamma_{\mathrm{D,sup}}$, $\Gamma_{\mathrm{D,sub\text{-}in}}$, $\Gamma_{\mathrm{D,sub\text{-}out}}$, Γ_{N}, and Γ_{W} are distinct subsets of Γ representing Dirichlet (supersonic), Dirichlet (subsonic-inflow), Dirichlet (subsonic-outflow), Neumann (supersonic-outflow), and solid wall boundaries, respectively. Thereby, we may specify the following boundary conditions:

$$\mathscr{B}(\mathbf{u}) = \mathscr{B}(\mathbf{g}_{\mathrm{D}}) \quad \text{on } \Gamma_{\mathrm{D,sup}} \cup \Gamma_{\mathrm{D,sub\text{-}in}} \cup \Gamma_{\mathrm{D,sub\text{-}out}}, \quad \mathscr{F}^v(\mathbf{u},\nabla\mathbf{u})\cdot\mathbf{n} = \mathbf{g}_{\mathrm{N}} \quad \text{on } \Gamma_{\mathrm{N}} \tag{13}$$

where $\mathbf{g}_{\mathrm{D}}$ and $\mathbf{g}_{\mathrm{N}}$ are given Dirichlet and Neumann boundary conditions, respectively. Here, $\mathscr{B}$ is a boundary operator employed to enforce appropriate Dirichlet conditions on $\Gamma_{\mathrm{D,sup}} \cup \Gamma_{\mathrm{D,sub\text{-}in}} \cup \Gamma_{\mathrm{D,sub\text{-}out}}$. For simplicity of presentation, we assume that $\mathscr{B}(\mathbf{u}) = \mathbf{u}$ on $\Gamma_{\mathrm{D,sup}}$, $\mathscr{B}(\mathbf{u}) = (u_1, u_2, u_3, 0)^{\mathrm{T}}$ on $\Gamma_{\mathrm{D,sub\text{-}in}}$ and $\mathscr{B}(\mathbf{u}) = (0,0,0,(\gamma-1)(u_4 - (u_2^2 + u_3^2)/(2u_1)))^{\mathrm{T}}$ on $\Gamma_{\mathrm{D,sub\text{-}out}}$; we note that this latter condition enforces a specific pressure $p_{\mathrm{out}} = (\mathscr{B}(\mathbf{g}_{\mathrm{D}}))_4$ on $\Gamma_{\mathrm{D,sub\text{-}out}}$.

For solid wall boundaries, we consider the distinction between *isothermal* and *adiabatic* conditions. To this end, decomposing $\Gamma_{\mathrm{W}} = \Gamma_{\mathrm{W,iso}} \cup \Gamma_{\mathrm{W,adia}}$, we set

$$\mathbf{v} = \mathbf{0} \quad \text{on } \Gamma_{\mathrm{W}}, \quad T = T_{\mathrm{wall}} \quad \text{on } \Gamma_{\mathrm{W,iso}}, \quad \mathbf{n}\cdot\nabla T = 0 \quad \text{on } \Gamma_{\mathrm{W,adia}}$$

where T_{wall} is a given wall temperature.

3. THE DISCONTINUOUS GALERKIN DISCRETIZATION

In this section we state the discontinuous Galerkin method with interior penalty for the discretization of the compressible Navier–Stokes equations (12) as already given in Reference [8].

We begin by first introducing some notation. We assume that Ω can be subdivided into shape-regular meshes $\mathcal{T}_h = \{\kappa\}$ consisting of quadrilateral elements κ. For each $\kappa \in \mathcal{T}_h$, we denote by $\mathbf{n}_\kappa$ the unit outward normal vector to the boundary $\partial\kappa$, and by h_κ the elemental diameter. An interior edge of $\mathcal{T}_h$ is the (non-empty) one-dimensional interior of $\partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$. Similarly, a boundary edge of $\mathcal{T}_h$ is the (non-empty) one-dimensional interior of $\partial\kappa \cap \Gamma$ which consists of entire edges of $\partial\kappa$. We denote by $\Gamma_\mathcal{I}$ the union of all interior edges of $\mathcal{T}_h$.

Next, we define average and jump operators. To this end, let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior edge $e = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_\mathcal{I}$. Moreover, let $\mathbf{v}$ and $\underline{\tau}$ be vector- and matrix-valued functions, respectively, that are smooth inside each element $\kappa^\pm$. By $(\mathbf{v}^\pm, \underline{\tau}^\pm)$ we denote the traces of $(\mathbf{v}, \underline{\tau})$ on e taken from within the interior of $\kappa^\pm$, respectively. Then, we define the averages at $\mathbf{x} \in e$ by $\{\!\{\mathbf{v}\}\!\} = (\mathbf{v}^+ + \mathbf{v}^-)/2$ and $\{\!\{\underline{\tau}\}\!\} = (\underline{\tau}^+ + \underline{\tau}^-)/2$. Similarly, the jumps at $\mathbf{x} \in e$ are given by $\underline{[\![\mathbf{v}]\!]} = \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}$ and $[\![\underline{\tau}]\!] = \underline{\tau}^+ \cdot \mathbf{n}_{\kappa^+} + \underline{\tau}^- \cdot \mathbf{n}_{\kappa^-}$. On a boundary edge $e \subset \Gamma$, we set $\{\!\{\mathbf{v}\}\!\} = \mathbf{v}$, $\{\!\{\underline{\tau}\}\!\} = \underline{\tau}$ and $\underline{[\![\mathbf{v}]\!]} = \mathbf{v} \otimes \mathbf{n}$. For matrices $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m\times n}$, $m, n \geqslant 1$, we use the standard notation $\underline{\sigma} : \underline{\tau} = \sum_{k=1}^m \sum_{l=1}^n \sigma_{kl}\tau_{kl}$; additionally, for vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$, the matrix $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{m\times n}$ is defined by $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k w_l$.

Finally, we introduce the finite element space

$$\mathbf{V}_h = \{\mathbf{v} \in [L_2(\Omega)]^4 : \mathbf{v}|_\kappa \in [\mathcal{Q}_p(\kappa)]^4, \kappa \in \mathcal{T}_h\} \tag{14}$$

for an approximation order $p \geqslant 1$. Here, $\mathcal{Q}_p(\kappa)$ denotes the space of tensor product polynomials on κ of degree p in each coordinate direction.

Then, the symmetric interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations (12), see Reference [8], including shock-capturing is defined as follows: find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$\begin{aligned}
\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) \equiv & -\int_\Omega \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa\in\mathcal{T}_h} \int_{\partial\kappa\setminus\Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa) \cdot \mathbf{v}_h^+ \,\mathrm{d}s \\
& + \int_\Omega \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \mathcal{N}_{\mathrm{sc}}(\mathbf{u}_h, \mathbf{v}_h) - \int_{\Gamma_\mathcal{I}} \{\!\{\mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h)\}\!\} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& - \int_{\Gamma_\mathcal{I}} \{\!\{(G_{i1}^{\mathrm{T}} \partial_h \mathbf{v}_h/\partial x_i, G_{i2}^{\mathrm{T}} \partial_h \mathbf{v}_h/\partial x_i)\}\!\} : \underline{[\![\mathbf{u}_h]\!]} \,\mathrm{d}s + \int_{\Gamma_\mathcal{I}} \delta \underline{[\![\mathbf{u}_h]\!]} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& + \int_\Gamma \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) \cdot \mathbf{v}_h^+ \,\mathrm{d}s + \int_{\Gamma\setminus\Gamma_\mathrm{N}} \delta(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \cdot \mathbf{v}_h^+ \,\mathrm{d}s - \int_{\Gamma_\mathrm{N}} \mathbf{g}_\mathrm{N} \cdot \mathbf{v}_h \,\mathrm{d}s
\end{aligned}$$

$$-\int_{\Gamma\backslash(\Gamma_{\mathrm{N}}\cup\Gamma_{\mathrm{W,adia}})}\mathscr{F}^{v}(\mathbf{u}_h^+,\nabla_h\mathbf{u}_h^+):\underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s-\int_{\Gamma_{\mathrm{W,adia}}}\mathscr{F}^{v,\mathrm{adia}}(\mathbf{u}_h^+,\nabla_h\mathbf{u}_h^+):\underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s$$

$$-\int_{\Gamma\backslash\Gamma_{\mathrm{N}}}(G_{i1}^{\mathrm{T}}(\mathbf{u}_h^+)\partial_h\mathbf{v}_h^+/\partial x_i,G_{i2}^{\mathrm{T}}(\mathbf{u}_h^+)\partial_h\mathbf{v}_h^+/\partial x_i):(\mathbf{u}_h^+-\mathbf{u}_\Gamma(\mathbf{u}_h^+))\otimes\mathbf{n}\,\mathrm{d}s=0 \qquad (15)$$

for all $\mathbf{v}_h$ in $\mathbf{V}_h$. Here, the numerical flux function $\mathscr{H}(\cdot,\cdot,\cdot)$, may be chosen to be any Lipschitz continuous, consistent and conservative flux. We note, that we employ the local Lax–Friedrichs flux in Section 8.

Furthermore, we set the discontinuity penalization matrix $\delta=\mathrm{diag}\{\delta_i,i=1,\dots,4\}$, where

$$\delta_i|_e=C_{\mathrm{IP}}\frac{\mu p^2}{\tilde{h}} \quad \text{for } e\subset\Gamma_{\mathscr{I}}\cup\Gamma \qquad (16)$$

$\tilde{h}=\min(\mathrm{meas}(\kappa),\mathrm{meas}(\kappa'))/\mathrm{meas}(e)$ represents the element dimension orthogonal to the edge e of elements κ and κ' adjacent to e, and C_{IP} is a positive constant, which, for reasons of stability, must be chosen sufficiently large, cf. Reference [19].

The boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ is given according to the type of boundary condition imposed. To this end, we set $\mathbf{u}_\Gamma(\mathbf{u})=\mathbf{g}_D$ on $\Gamma_{\mathrm{D,sup}}$, $\mathbf{u}_\Gamma(\mathbf{u})=\mathbf{u}$ on Γ_{N}, $\mathbf{u}_\Gamma(\mathbf{u})=((g_D)_1,(g_D)_2,(g_D)_3,(p(\mathbf{u})/\gamma-1)+((g_D)_2^2+(g_D)_3^2)/(2(g_D)_1))^{\mathrm{T}}$ on $\Gamma_{\mathrm{D,sub\text{-}in}}$, and $\mathbf{u}_\Gamma(\mathbf{u})=(u_1,u_2,u_3,(p_{\mathrm{out}}/\gamma-1)+(u_2^2+u_3^2)/(2u_1))^{\mathrm{T}}$ on $\Gamma_{\mathrm{D,sub\text{-}out}}$. Here, $p\equiv p(\mathbf{u})$ denotes the pressure evaluated using the equation of state (10). We set $\mathbf{u}_\Gamma(\mathbf{u})=(u_1,0,0,u_1c_vT_{\mathrm{wall}})^{\mathrm{T}}$ on $\Gamma_{\mathrm{W,iso}}$, $\mathbf{u}_\Gamma(\mathbf{u})=(u_1,0,0,u_4)^{\mathrm{T}}$ on $\Gamma_{\mathrm{W,adia}}$, and define $\mathscr{F}^{v,\mathrm{adia}}(\mathbf{u},\nabla\mathbf{u})$ such that

$$\mathscr{F}^{v,\mathrm{adia}}(\mathbf{u},\nabla\mathbf{u})\cdot\mathbf{n}=(0,\tau_{1j}n_{x_j},\tau_{2j}n_{x_j},\tau_{ij}v_jn_{x_i})^{\mathrm{T}}$$

Finally, the shock-capturing term $\mathscr{N}_{\mathrm{sc}}(\mathbf{u}_h,\mathbf{v}_h)$ will be specified in the following section.

4. SHOCK-CAPTURING

As already indicated in the introductory Section 1.1, the discontinuous Galerkin discretization (15) for the compressible Navier–Stokes equations is supplemented with an artificial viscosity term given by

$$\mathscr{N}_{\mathrm{sc}}(\mathbf{u}_h,\mathbf{v}_h)\equiv\sum_\kappa\int_\kappa\varepsilon(\mathbf{u}_h)\nabla\mathbf{u}_h:\nabla\mathbf{v}_h\,\mathrm{d}\mathbf{x}\equiv\sum_\kappa\int_\kappa\varepsilon_{ki}(\mathbf{u}_h)\partial_{x_i}u_h^k\,\partial_{x_i}v_h^k\,\mathrm{d}\mathbf{x} \qquad (17)$$

In particular, we choose the coefficient matrix ε_{ki} to be

$$\varepsilon_{ki}(\mathbf{u}_h)=C_\varepsilon h_i^{2-\beta}\mathscr{R}_k(\mathbf{u}_h), \quad i=1,2,\;\; k=1,\dots,4 \qquad (18)$$

where C_ε and β are positive constants and h_i represents the dimension of element κ in the ith coordinate direction, $i=1,2$. Finally, $\mathscr{R}_k(\mathbf{u}_h)$, $k=1,\dots,4$, is defined by

$$\mathscr{R}_k(\mathbf{u})=\sum_{q=1}^{4}|R_q(\mathbf{u})|, \quad k=1,\dots,4 \qquad (19)$$

where $\mathbf{R}(\mathbf{u}_h) = (R_q(\mathbf{u}_h), q = 1, \ldots, 4)$ denotes the residual of Equations (7), i.e.

$$\mathbf{R}(\mathbf{u}) = -\nabla \cdot (\mathscr{F}^c(\mathbf{u}) - \mathscr{F}^v(\mathbf{u}, \nabla \mathbf{u})) \tag{20}$$

Remarks

(i) Due to the specific choice of the artificial viscosity term $\mathscr{N}_{\mathrm{sc}}(\mathbf{u}_h, \mathbf{v}_h)$ depending on the residual $\mathbf{R}(\mathbf{u})$, see (20), this term vanishes when evaluated for the exact and sufficiently regular solution $\mathbf{u}$ to problem (7). This ensures that the discretization remains consistent in the sense that the Galerkin orthogonality,

$$\mathscr{N}(\mathbf{u}, \mathbf{v}_h) - \mathscr{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \mathbf{v}_h \in \mathbf{V}_h \tag{21}$$

with $\mathbf{u}$ and $\mathbf{u}_h$ denoting the solutions of (7) and (15), respectively, which is valid for the discretization without shock-capturing, see Reference [6], is still valid when the shock-capturing term $\mathscr{N}_{\mathrm{sc}}(\cdot, \cdot)$ is included.

(ii) Furthermore, we note that in (19), $\mathscr{R}_k(\mathbf{u})$ is the same constant for all components k, $k = 1, \ldots, 4$. A simpler choice

$$\mathscr{R}_k(\mathbf{u}) = R_k(\mathbf{u}), \quad k = 1, \ldots, 4$$

is cheaper in terms of assembling time of the Jacobian, cf. Section 5, but turned out to be unstable for various numerical test cases.

(iii) The choice of the coefficient matrix in (18) represents an extension of the shock-capturing term, already employed in Reference [7] for the compressible Euler equations, to the compressible Navier–Stokes equations. Furthermore, it represents a generalization to anisotropic meshes. In fact, its isotropic version, i.e. setting $h_i = h_\kappa$, $i = 1, 2$, where h_κ represents the diameter of the element κ, was found to not work on meshes with anisotropic elements.

5. GMRES–NEWTON ALGORITHM

To determine the numerical solution $\mathbf{u}_h$ of the system of nonlinear equations (15), we employ a damped Newton method. This nonlinear iteration generates a sequence of approximations $\mathbf{u}_h^n$, $n = 0, 1, \ldots,$ to the actual numerical solution $\mathbf{u}_h$, using the following algorithm. Given an iterate $\mathbf{u}_h^n$, the update $\mathbf{d}_h^n$ of $\mathbf{u}_h^n$ to get to the next iterate

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \omega^n \mathbf{d}_h^n$$

is defined by: find $\mathbf{d}_h^n \in \mathbf{V}_h$ such that

$$\mathscr{N}'_{\mathbf{u}}[\mathbf{u}_h^n](\mathbf{d}_h^n, \mathbf{v}_h) = R(\mathbf{u}_h^n, \mathbf{v}_h) \equiv -\mathscr{N}(\mathbf{u}_h^n, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h \tag{22}$$

Here, ω^n denotes a damping parameter, which is dynamically chosen to guarantee that the discrete l_2-norm of the residual computed with $\mathbf{u}_h^{n+1}$ is less than the same quantity computed with $\mathbf{u}_h^n$. Additionally, $\mathscr{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ denotes (an approximation to) the Fréchet derivative of

$\mathbf{u} \to \mathcal{N}(\mathbf{u}, \mathbf{v})$, for $\mathbf{v} \in \mathbf{V}_h$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$, where $\mathbf{V}$ is some suitable chosen function space such that $\mathbf{V}_h \in \mathbf{V}$. A detailed description of how $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ is approximated for the DG discretization (15) without shock-capturing term is given in Reference [8]. In addition to this, the shock-capturing part $\mathcal{N}'_{\mathrm{sc},\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ can be approximated as follows:

$$\hat{\mathcal{N}}'_{sc,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}) = \sum_{\kappa} \int_{\kappa} \varepsilon(\mathbf{w}) \nabla \boldsymbol{\phi} \cdot \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + \sum_{\kappa} \int_{\kappa} \hat{\varepsilon}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) \nabla \mathbf{w} \cdot \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} \tag{23}$$

where $\hat{\varepsilon}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi})$ is given by

$$\hat{\varepsilon}'_{ik,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) = C_\varepsilon h_k^{2-\beta} \mathcal{R}'_{i,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}), \quad i = 1, \ldots, 4, \;\; k = 1, 2$$

and

$$\mathcal{R}'_{i,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}) = \sum_{q=1}^{4} \mathrm{sgn}(R_q(\mathbf{w})) R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}), \quad i = 1, \ldots, 4$$

Recalling the definition of $R_q(\mathbf{u})$ in (20),

$$\begin{aligned} R_q(\mathbf{u}) &= -\partial_{x_p} f^c_{pq}(\mathbf{u}) + \partial_{x_p} f^v_{pq}(\mathbf{u}, \nabla \mathbf{u}) \\ &= -\partial_{u^r} f^c_{pq}(\mathbf{u}) \partial_{x_p} u^r + \partial_{u^r}(G(\mathbf{u})_{pl})_{qs} \partial_{x_p} u^r \partial_{x_l} u^s + (G(\mathbf{u})_{pl})_{qs} \partial_{x_p} \partial_{x_l} u^s \end{aligned}$$

we obtain the following expression for $R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi})$:

$$\begin{aligned} R'_{q,\mathbf{u}}[\mathbf{w}](\boldsymbol{\varphi}) = & -\partial_{u^j} f^c_{pq}(\mathbf{w}) \partial_{x_p} \varphi^j - \partial_{u^j} \partial_{u^r} f^c_{pq}(\mathbf{w}) \varphi^j \partial_{x_p} w^r \\ & + \partial_{u^j} \partial_{u^r} (G(\mathbf{w})_{pl})_{qs} \varphi^j \partial_{x_p} w^r \partial_{x_l} w^s \\ & + \partial_{u^j} (G(\mathbf{w})_{pl})_{qs} \partial_{x_p} \varphi^j \partial_{x_l} w^s \\ & + \partial_{u^r} (G(\mathbf{w})_{pl})_{qj} \partial_{x_p} w^r \partial_{x_l} \varphi^j \\ & + \partial_{u^j} (G(\mathbf{w})_{pl})_{qs} \varphi^j \partial_{x_p} \partial_{x_l} w^s \\ & + (G(\mathbf{w})_{pl})_{qj} \partial_{x_p} \partial_{x_l} \varphi^j \end{aligned} \tag{24}$$

Remark

Evaluating the term including the second derivatives of G in the second line of (24) is extremely time consuming. In Section 8 we demonstrate that neglecting this term in the assembly of the Jacobian matrix does not deteriorate the convergence of the Newton algorithm. In fact, the same number of Newton steps are required than when this term is included, but, the time savings in the matrix assembly when neglecting this term, finally lead to a significant decrease of the overall computing time.

6. *A POSTERIORI* ERROR ESTIMATION

In this section, we shall be concerned with controlling the error in the numerical solution measured in terms of specific target quantities.

In particular, we consider the estimation of the drag and lift coefficients, c_{d} and c_{l}, respectively, which, in the case of a viscous flow, are defined by

$$J_{c_{\mathrm{d}}}(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u}) + J_{c_{\mathrm{df}}}(\mathbf{u})$$

$$J_{c_{\mathrm{l}}}(\mathbf{u}) = J_{c_{\mathrm{lp}}}(\mathbf{u}) + J_{c_{\mathrm{lf}}}(\mathbf{u})$$

respectively, where c_{dp} and c_{lp} are the pressure induced force coefficients given by

$$J_{c_{\mathrm{dp}}}(\mathbf{u}) = \frac{2}{l\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_d)\,\mathrm{d}s, \quad J_{c_{\mathrm{lp}}}(\mathbf{u}) = \frac{2}{l\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_l)\,\mathrm{d}s$$

respectively, and c_{df} and c_{lf} are the viscous force coefficients given by

$$J_{c_{\mathrm{df}}}(\mathbf{u}) = \frac{2}{l\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_d\,\mathrm{d}s, \quad J_{c_{\mathrm{lf}}}(\mathbf{u}) = \frac{2}{l\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_l\,\mathrm{d}s$$

respectively. Here, S denotes the surface of the airfoil, l its chord length, $\bar{\mathbf{v}}$ and $\bar{\rho}$ are the reference (or free-stream) velocity and density, respectively, $(\underline{\tau}\,\mathbf{n})\cdot\psi = \tau_{ij}n_j\psi_i$, where τ is the viscous stress tensor defined in (11) and

$$\psi_d = \begin{pmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}, \quad \psi_l = \begin{pmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}$$

We note that since the pressure p and the viscous stress tensor τ are derived from the conserved variables $(\rho, \rho u, \rho v, \rho E)$, the pressure induced as well as the viscous stress induced force coefficients are nonlinear functionals.

Other examples of $J(\cdot)$ include the local mean value of the field or its flux through the outflow boundary of the computational domain Ω, and the point evaluation of a component of $\mathbf{u}$ in Ω. Assuming that the functional of interest $J(\cdot)$ is differentiable, we write $\bar{J}(\cdot;\cdot)$ to denote the mean value linearization of $J(\cdot)$ defined by

$$\bar{J}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h) = J(\mathbf{u}) - J(\mathbf{u}_h) = \int_0^1 J'[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u}-\mathbf{u}_h)\,\mathrm{d}\theta \tag{25}$$

where $J'[\mathbf{w}](\cdot)$ denotes the Fréchet derivative of $J(\cdot)$ evaluated at some $\mathbf{w}$ in $\mathbf{V}$. Here, $\mathbf{V}$ is some suitably chosen function space such that $\mathbf{V}_h \subset \mathbf{V}$. Analogously, we write $\mathscr{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ to denote the mean-value linearization of $\mathscr{N}(\cdot,\cdot)$ given by

$$\begin{aligned}\mathscr{M}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h,\mathbf{v}) &= \mathscr{N}(\mathbf{u},\mathbf{v}) - \mathscr{N}(\mathbf{u}_h,\mathbf{v})\\ &= \int_0^1 \mathscr{N}'_{\mathbf{u}}[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u}-\mathbf{u}_h,\mathbf{v})\,\mathrm{d}\theta\end{aligned} \tag{26}$$

for all $\mathbf{v}$ in $\mathbf{V}$. Here, $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\mathbf{v})$ denotes the Fréchet derivative of $\mathbf{u} \mapsto \mathcal{N}(\mathbf{u},\mathbf{v})$, for $\mathbf{v} \in V$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$. We remark that the linearization defined in (26) is only a *formal* calculation, in the sense that $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\cdot)$ may not in general exist. Instead, a suitable approximation to $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\cdot)$ must be determined, see Section 5 [8, 20]. For the proceeding analysis, we *assume* that the linearization (26) is well-defined. Under this hypothesis, we introduce the following *dual problem*: find $\mathbf{z} \in \mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u},\mathbf{u}_h;\mathbf{w},\mathbf{z}) = \bar{J}(\mathbf{u},\mathbf{u}_h;\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V} \tag{27}$$

We assume that (27) possesses a unique solution. Clearly, the validity of this assumption depends on both the definition of $\mathcal{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ and the choice of the target functional under consideration, cf. Reference [18]. For the proceeding error analysis, we must therefore *assume* that the dual problem (27) is well-posed. Under this assumption, we have the following result.

Theorem 6.1

Let $\mathbf{u}$ and $\mathbf{u}_h$ denote the solutions of (7) and (15), respectively, and suppose that the dual problem (27) is well-posed. Then

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \tag{28}$$

for all $\mathbf{z}_h$ in $\mathbf{V}_h$.

Proof

Choosing $\mathbf{w} = \mathbf{u} - \mathbf{u}_h$ in (27), recalling the linearization performed in (25), and exploiting the Galerkin orthogonality property (21) for all $\mathbf{v}_h$ in $\mathbf{V}_h$, we get

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \bar{J}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h) = \mathcal{M}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h,\mathbf{z}) \\ &= \mathcal{M}(\mathbf{u},\mathbf{u}_h;\mathbf{u}-\mathbf{u}_h,\mathbf{z}-\mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h,\mathbf{z}-\mathbf{z}_h) \end{aligned}$$

for all $\mathbf{z}_h$ in $\mathbf{V}_h$.

Based on the general error representation formula derived in Theorem 6.1, which can be written as follows:

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa \tag{29}$$

where η_κ includes the face and element residuals multiplied by the dual solution, see Reference [6], *a posteriori* error estimates bounding the error in the computed functional $J(\cdot)$ may be deduced. Here, we shall confine ourselves to considering a Type I *a posteriori* error bound; it is a straightforward consequence of the error representation formula stated in the previous theorem.

Under the assumptions of Theorem 6.1, we have

$$|J(\mathbf{u}) - J(\mathbf{u}_h)| \leqslant \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa| \tag{30}$$

This bound follows from (29) by application of the triangle inequality.

We end this section by noting that both the error representation formula (28) and the Type I *a posteriori* error bound (30) depend on the unknown analytical solution to the primal and dual problems. Thus, in order to render these quantities computable, both $\mathbf{u}$ and $\mathbf{z}$ must be replaced by suitable approximations. The linearizations leading to $\mathscr{M}(\mathbf{u},\mathbf{u}_h;\cdot,\cdot)$ and $\bar{J}(\mathbf{u},\mathbf{u}_h;\cdot)$ are performed about $\mathbf{u}_h$, resulting in $\mathscr{N}'[\mathbf{u}_h](\cdot,\cdot)$ and $J'[\mathbf{u}_h](\cdot)$, respectively. The *linearized dual problem*: find $\hat{\mathbf{z}}\in\mathbf{V}$ such that

$$\mathscr{N}'[\mathbf{u}_h](\mathbf{w},\hat{\mathbf{z}})=J'[\mathbf{u}_h](\mathbf{w})\quad\forall\mathbf{w}\in\mathbf{V}\tag{31}$$

is then discretized using discontinuous Galerkin finite elements, to yield following *approximate dual problem*: find $\hat{\mathbf{z}}_h\in\hat{\mathbf{V}}_h$ such that

$$\mathscr{N}'[\mathbf{u}_h](\mathbf{w}_h,\hat{\mathbf{z}}_h)=J'[\mathbf{u}_h](\mathbf{w}_h)\quad\forall\mathbf{w}_h\in\hat{\mathbf{V}}_h\tag{32}$$

Here, the approximate dual solution $\hat{\mathbf{z}}_h$ is computed on the same mesh $\mathscr{T}_h$ used for $\mathbf{u}_h$, but with a higher degree polynomial, i.e. $\hat{\mathbf{z}}\in\hat{\mathbf{V}}_h$ with

$$\hat{\mathbf{V}}_h=\{\mathbf{v}\in[L_2(\Omega)]^4:\mathbf{v}|_\kappa\in[\mathscr{Q}_{\hat{p}}(\kappa)]^4,\kappa\in\mathscr{T}_h\}\tag{33}$$

and $\hat{p}>p$. Replacing the dual solution $\mathbf{z}$ in (28) by its approximation $\hat{\mathbf{z}}_h$ results in following approximate error representation formula:

$$J(\mathbf{u})-J(\mathbf{u}_h)\approx-\mathscr{N}(\mathbf{u}_h,\hat{\mathbf{z}}_h-\mathbf{z}_h)\equiv\sum_{\kappa\in\mathscr{T}_h}\hat{\eta}_\kappa\tag{34}$$

and an analogous formula for the approximate Type I error bound. We note that the error introduced into the error representation through this replacement consists of the linearization and the discretization error of the dual problem, see Section 8.2.3 for a more detailed discussion.

7. ADAPTIVE MESH REFINEMENT

In this section we consider the design of an adaptive algorithm to ensure the efficient computation of the given target functional $J(\cdot)$ of practical interest. To this end, we employ the approximate Type I *a posteriori* error bound $\sum_{\kappa\in\mathscr{T}_h}|\hat{\eta}_\kappa|$ to determine when the desired level of accuracy has been achieved. For example, suppose that the aim of the computation is to compute $J(\cdot)$ such that the error $|J(\mathbf{u})-J(\mathbf{u}_h)|$ is less than some user-defined tolerance $\mathtt{TOL}$, i.e.

$$|J(\mathbf{u})-J(\mathbf{u}_h)|\leqslant\mathtt{TOL}$$

then, in practice we may enforce the stopping criterion

$$\sum_{\kappa\in\mathscr{T}_h}|\hat{\eta}_\kappa|\leqslant\mathtt{TOL}$$

If this condition is not satisfied on the current finite element mesh $\mathscr{T}_h$, then the elementwise terms $\hat{\eta}_\kappa^{(\mathrm{I})}$ are employed as local error indicators to guide mesh refinement and coarsening.

The cycle of the adaptive mesh refinement is outlined as follows:

1. Construct an initial mesh $\mathscr{T}_h$.
2. Compute $\mathbf{u}_h \in \mathbf{V}_h$ on the current mesh $\mathscr{T}_h$.
3. Compute $\hat{\mathbf{z}}_h \in \hat{\mathbf{V}}_h$, where $\hat{\mathbf{V}}_h$ is a finite element space defined in an analogous manner to $\mathbf{V}_h$ based on the (same) computational mesh $\mathscr{T}_h$, but consisting of piecewise (discontinuous) polynomials of degree $\hat{p} > p$, see (33).
4. Evaluate the approximate *a posteriori* error bound $\sum_{\kappa \in \mathscr{T}_h} |\hat{\eta}_\kappa|$.
5. If $\sum_{\kappa \in \mathscr{T}_h} |\hat{\eta}_\kappa| \leqslant \mathtt{TOL}$, where TOL is a given tolerance, then STOP.
6. Otherwise, refine and coarsen a fixed fraction of the total number of elements according to the size of $|\hat{\eta}_\kappa|$ and generate a new mesh $\mathscr{T}_h$; GOTO 2.

8. NUMERICAL RESULTS

In this section we consider supersonic viscous flows around a NACA0012 airfoil at two different flow conditions; here, the upper and lower surfaces of the airfoil geometry are specified by the function $g^{\pm}$, respectively, where

$$g^{\pm}(s) = \pm 5 \times 0.12 \times (0.2969 s^{1/2} - 0.126 s - 0.3516 s^2 + 0.2843 s^3 - 0.1015 s^4)$$

As the chord length l of the airfoil is $l \approx 1.00893$ we use a rescaling of g in order to yield an airfoil of unit (chord) length. The computational domain Ω is subdivided into quadrilateral elements; cf. the C-type grid depicted in Figures 1(a) and (b) which extends about 20 cord length. Curved boundaries are approximated by piecewise quadratic polynomials. Furthermore, we use piecewise linear ansatz and test functions, i.e. set $p=1$ in (14) and we employ the local Lax–Friedrichs flux, cf. Reference [18] for example. We choose the constant of the discontinuity penalization term to be $C_{\mathrm{IP}} = 10$, cf. Reference [8], and the constants β and C_ε in the shock-capturing term (18) to be $\beta = 0.1$, cf. Reference [11], and $C_\varepsilon = 0.1$, respectively.

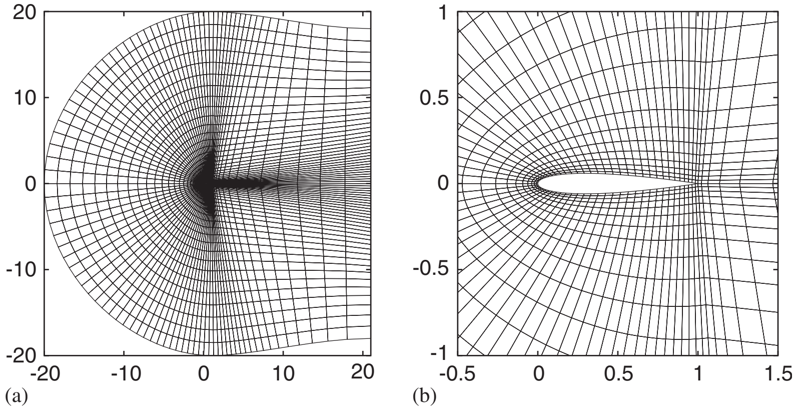


Figure 1. Computational mesh: (a) full view; and (b) zoom of coarse grid with 3072 elements.

8.1. *Viscous flow at $M = 2$, $Re = 106$ and $\alpha = 10°$ around the NACA0012 airfoil*

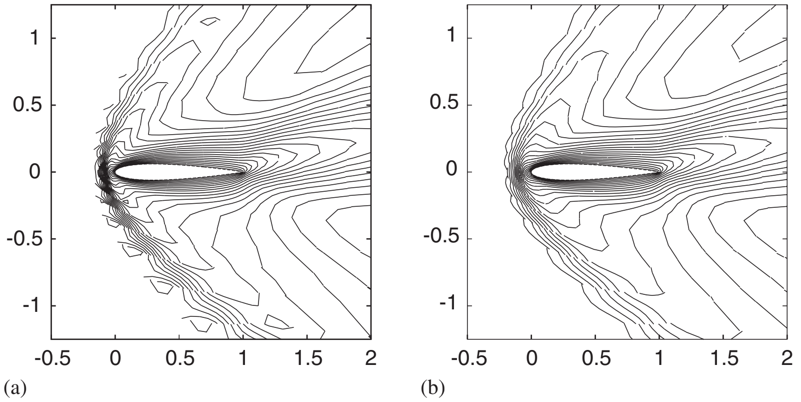
In this first example we consider a laminar flow at $M = 2$, $Re = 106$ and $\alpha = 10°$ with constant temperature on the profile, a test case previously being considered in References [1, 21], for example.

We note, that the discontinuous Galerkin discretization applied to this problem can be solved without any shock capturing employed as long as the numerical dissipation is sufficiently large, like on very coarse meshes, for example. In fact, in Reference [1] this problem has been solved without any stabilization on a rather coarse mesh (O-grid with 1024 elements). Also on the mesh, see Figure 1, with 3072 elements, the unstabilized DG discretization can be solved, see Figure 2 for a comparison of the numerical solutions both with and without shock-capturing. But, when this mesh is once globally refined, the unstabilized version cannot be solved any more. In contrast to this, the stabilized version can be solved, see Figures 3(a) and (b) for the discrete solutions on the mesh depicted in Figure 1 being once and twice globally refined, respectively.

8.2. *Viscous flow at $M = 1.2$, $Re = 1000$ and $\alpha = 0°$ around the NACA0012 airfoil*

In this second example we consider a horizontal viscous flow at $M = 1.2$ and $Re = 1000$, with an adiabatic no-slip boundary condition imposed on the profile. Due to the higher Reynold's number the bow shock of the flow, see Figure 4, is sharper and, due to the lower Mach number, it is located at a larger distance in front of the airfoil than in the previous test case. Furthermore, there are two weak shocks emanating from the trailing edge of the airfoil, see Figure 5.

8.2.1. Performance of the nonlinear solution process. First, we concentrate on the performance of the Newton iteration, cf. Section 5, on the mesh shown in Figure 1 with 3072 elements and 49 152 degrees of freedom for linear shape functions, $p = 1$, employed. Starting the Newton iteration on this mesh with a start solution being pre-iterated on a once coarser level of this grid, Figure 6 shows the convergence history of the discrete l_2-norm of the



(a) (b)

Figure 2. $M = 2$, $Re = 106$, $\alpha = 10°$ flow around the NACA0012 airfoil: Mach isolines of the discrete solution: (a) without; and (b) with shock capturing on the mesh of 3072 elements shown in Figure 1.

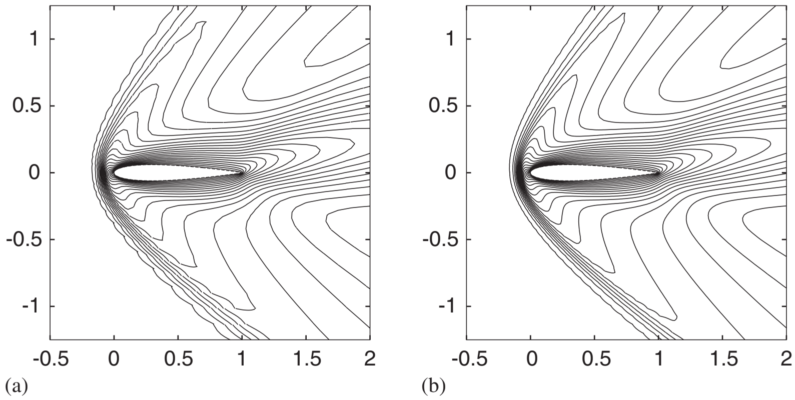


Figure 3. $M=2$, $Re=106$, $\alpha=10^{\circ}$ flow around the NACA0012 airfoil: Mach isolines of the discrete solution with shock-capturing on mesh Figure 1, which is: (a) once; and (b) twice globally refined.

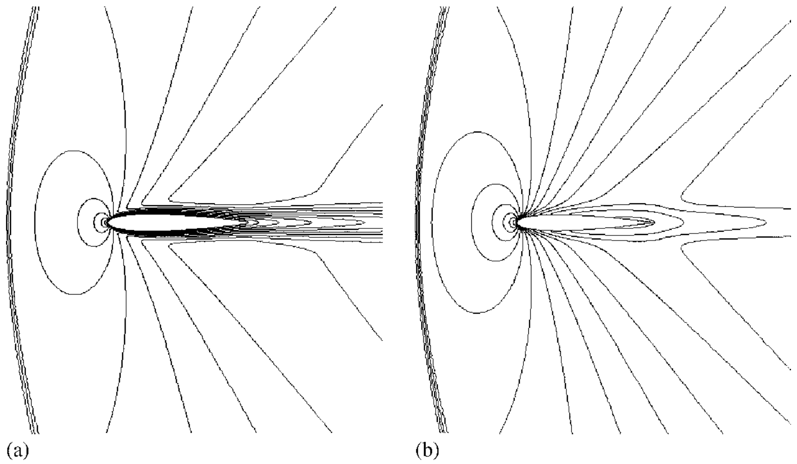


Figure 4. $M=1.2$, $Re=1000$, $\alpha=0^{\circ}$ flow around the NACA0012 airfoil: (a) Mach isolines; and (b) density isolines.

nonlinear residual for different approximations to the Jacobian (23) of the shock-capturing term (17). In particular, in Figure 6, we compare the convergence for the full Jacobian employed as given by (23) including all terms in the Jacobian (24) of the equation residual (20), for the approximation (`no ddG`) to the Jacobian neglecting the second derivatives of G in (24), for the approximation (`no ddG, no ddF`) neglecting the second derivatives of G and $\mathscr{F}^c$ and finally for the approximation (`no dR`) neglecting all terms in (24), i.e. assembling only the first term in (23). Furthermore, these different approximations are tested with the linear problems arising in each Newton step being solved up to two different accuracies, namely reducing the ℓ_2-norm of the linear residual by a factor of 10^{-3} and 10^{-6}, respectively.

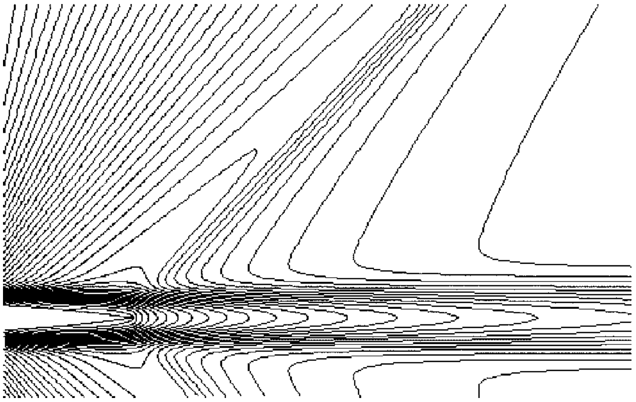


Figure 5. $M = 1.2$, $Re = 1000$, $\alpha = 0°$ flow around the NACA0012 airfoil: zoom of density isolines at trailing edge.

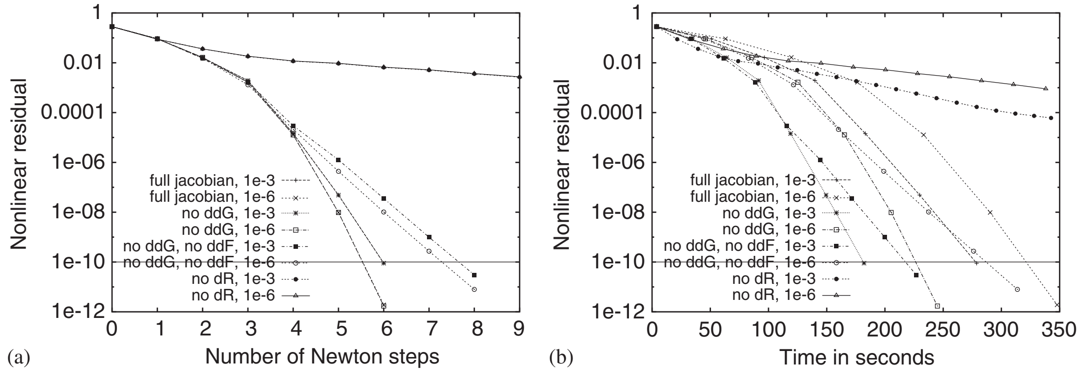


Figure 6. $M = 1.2$, $Re = 1000$, $\alpha = 0°$ flow around the NACA0012 airfoil: different approximations to the Jacobian of the artificial viscosity term on the performance of the Newton iteration: the nonlinear residual versus: (a) the number of Newton steps and versus; and (b) the computing time in seconds.

In Figure 6(a), we see that—as expected—the (`full jacobian`) performs the best in terms of number of Newton steps required for reducing the nonlinear residual below the prescribed tolerance of 10^{-10}, followed by the approximations neglecting various parts of the Jacobian, (`no ddG`), (`no ddG, no ddF`) and (`no dR`). In fact, only six Newton steps are required when the full Jacobian is employed or the second derivatives of G are neglected, in comparison to eight Newton steps when the second derivatives of both, G and $\mathscr{F}^v$, are neglected, and an overly large number of Newton steps when all terms in (24) are neglected. Finally, we note that there is a small difference only, in the performance of the Newton iteration when each linear step is solved up to a linear residual reduced by a factor of 10^{-3} or 10^{-6}, respectively.

In Figure 6(b), we show the respective convergence histories plotted versus the computing time elapsed. Here, we see that while the full Jacobian performed best with respect to nonlinear residual versus number of Newton steps, see Figure 6(a), it performes worse than (`no ddG`), Table I. $M = 1.2$, $Re = 1000$, $\alpha = 0°$ flow around the NACA0012 airfoil: nonlinear residuals and convergence rates of the Newton iteration on a sequence of globally refined meshes.

Mesh 1		Mesh 2		Mesh 3	
Res.	Rate	Res.	Rate	Res.	Rate
2.82-01	—	1.75-01	—	1.01-01	—
9.18-02	3	4.59-02	4	2.33-02	4
1.61-02	6	6.05-03	8	1.95-03	12
1.93-03	8	3.98-04	15	8.35-05	23
1.44-05	134	1.70-06	234	2.02-06	41
4.85-08	296	7.02-09	241	7.15-09	283
8.98-11	539	1.34-11	524	5.25-11	136

when measured in terms of computing time. This is due to the second derivatives of G being overly time-consuming in the matrix assembly. We conclude that the approach of employing the approximation to the Jacobian which neglects the second derivatives of G and reducing the linear residuals by a factor of 10^{-3} only, is performing the best in terms of computing time of the approaches considered.

For this approximation to the Jacobian, Table I shows the history of nonlinear residuals on a sequence of globally refined meshes. On each mesh the nonlinear residual is reduced below 10^{-10}, then the mesh is refined and the solution is interpolated providing a start solution of the Newton iteration on the next finer mesh. As before, the Newton iteration proceeds with full Newton steps (damping parameter $\omega^n = 1$) leading to a very fast convergence of the Newton iteration. On each mesh the rate of convergence increases significantly clearly indicating a superlinear convergence. In fact, on several single Newton steps the nonlinear residual is reduced by a factor of more than 100.

8.2.2. Performance of the error estimation and the adaptive goal-oriented refinement. In this subsection we present a numerical example demonstrating that the approximate error representation $-\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$, cf. (34), which was derived from the (exact) error representation (29) by replacing the dual solution $\mathbf{z}$ by an approximate dual solution $\hat{\mathbf{z}}_h$, gives a good approximation to the true error measured in terms of the specific target quantity $J(\mathbf{u})$. Furthermore, we highlight the advantages of designing an adaptive finite element algorithm as outlined in Section 7, based on *dual-weighted residual* indicators (also referred to as Type I error indicators), $\eta_\kappa^{(\mathrm{I})} := |\hat{\eta}_\kappa|$, in comparison to *residual-based* indicators (also referred to as Type II error indicators), $\eta_\kappa^{(\mathrm{II})}$, that originate from a Type II error bound, cf. Reference [6], but does not require the solution of an auxiliary (dual) problem.

To this end, we consider the M=1.2, Re=1000, α=0° viscous flow around the NACA0012 airfoil, the same test case as in the previous subsection. As this flow is symmetric about the x-axis, both lift coefficients, c_{lp} and c_{lf}, vanish. On the basis of fine grid computations the reference values of the pressure induced drag, c_{dp}, and the viscous drag, c_{df}, are given by $J_{c_{\mathrm{dp}}}(\mathbf{u}) \approx 0.10109$ and $J_{c_{\mathrm{df}}}(\mathbf{u}) \approx 0.10773$, respectively.

In the following, we consider the approximation of the pressure induced drag, c_{dp}, i.e. the target quantity is $J(\cdot) = J_{c_{\mathrm{dp}}}(\cdot)$. In Table II, we collect the data of the adaptive algorithm based

Table II. Viscous $M = 1.2$, $Re = 1000$, $\alpha = 0°$ flow around the NACA0012 airfoil: adaptive algorithm for the accurate approximation of c_{dp}.

# elements	# DoFs	$J(\mathbf{u}) - J(\mathbf{u}_h)$	$\sum_\kappa \hat{\eta}_\kappa$	θ
768	12 288	−1.363e-02	−6.312e-03	0.46
1260	20 160	−3.203e-03	−2.995e-03	0.94
2154	34 464	−4.844e-04	−5.368e-04	1.11
3570	57 120	−3.474e-04	−3.333e-04	0.96
6021	96 336	−1.835e-04	−1.856e-04	1.01
10 038	160 608	−1.644e-04	−1.653e-04	1.01

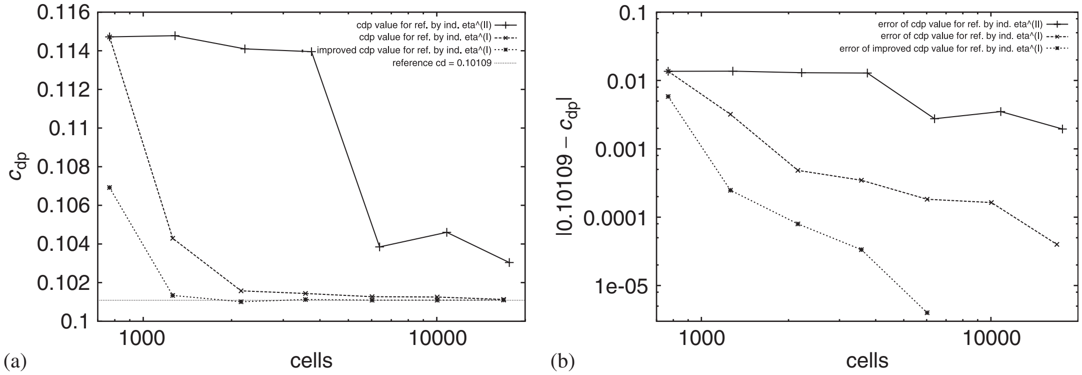


Figure 7. $M = 1.2$, $Re = 1000$, $\alpha = 0°$ flow around the NACA0012 airfoil: (a) $J_{c_{\mathrm{dp}}}(\mathbf{u})$ values on adaptive refined meshes using indicator $\eta_\kappa^{(\mathrm{II})}$, $J_{c_{\mathrm{dp}}}(\mathbf{u})$ and the improved values, $\tilde{J}_{c_{\mathrm{dp}}}(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u}) + \sum_\kappa \hat{\eta}_\kappa$, on adaptive refined meshes using indicator $\hat{\eta}_\kappa^{(\mathrm{I})}$ versus number of elements; and (b) error of these values versus number of elements.

on employing the weighted-residual indicators $\eta_\kappa^{(\mathrm{I})}$. Here, we show the number of elements and degrees of freedom (DoF) for $p = 1$ (bilinear elements), the true error in the target quantity, $J(\mathbf{u}) - J(\mathbf{u}_h)$, the approximate error representation formula $\sum_{\kappa \in \mathscr{T}_h} \hat{\eta}_\kappa$ and the effectivity index $\theta = \sum_{\kappa \in \mathscr{T}_h} \hat{\eta}_\kappa / (J(\mathbf{u}) - J(\mathbf{u}_h))$ of the error estimation. First, we note that on all meshes the right sign of the error is predicted, which is always negative in this computation, i.e. the computed c_{dp} values converge to the reference value from above. Furthermore, from the second mesh onwards, the approximate error representations represent a very good approximation to the true errors, also indicated by the effectivity indices θ being very close to one.

In Figure 7 we compare the true error in the target quantity for the two mesh refinement strategies based on the weighted-residual indicator $\eta_\kappa^{(\mathrm{I})}$ and on the residual-based indicator $\eta_\kappa^{(\mathrm{II})}$, respectively. We see, that on the first three refinement steps when employing the residual-based indicator the accuracy in the target quantity is hardly improved. In contrast to that, when using weighted-residual indicators, the error decreases significantly faster, being a factor of more than three smaller already after the second refinement step than the error on the finest residual-based refined mesh. Furthermore, the computed values of the target quantity $J(\mathbf{u}_h)$ can be enhanced by employing the approximate error representation $\sum_{\kappa \in \mathscr{T}_h} \hat{\eta}_\kappa$ to yield an *improved value* of the target quantity, $\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_h) + \sum_{\kappa\in\mathcal{T}_h} \hat{\eta}_\kappa$. In Figure 7 we see that the improved values $\tilde{J}(\mathbf{u}_h)$, are significantly more accurate than the (baseline) $J(\mathbf{u}_h)$ values, and even show a higher rate of convergence. In fact, it can be shown, see Reference [6], that this value has a higher order of convergence than $J(\mathbf{u}_h)$, provided the primal and the dual solutions are smooth and the dual solution is approximated using higher-order polynomials. Furthermore, the approximate error representation is close to the true error even in cases of smooth dual solutions but possibly non-smooth primal solutions, see Section 8.2.3 for a more detailed discussion.

The large difference in the performance, see Figure 7, of the weighted-residual indicator $\eta_\kappa^{(\mathrm{I})}$ and the residual-based indicator $\eta_\kappa^{(\mathrm{II})}$ in producing adaptively refined meshes for the accurate approximation of the target quantity c_{dp}, is due to the very different parts of the computational meshes being marked for refinement by the two types of indicators. Figures 8(a) and (b) show the finest mesh produced by employing the residual-based indicator $\eta_\kappa^{(\mathrm{II})}$. We see, that this refinement criterion aims at resolving all flow features: the extensive bow shock, the wake of the flow behind the airfoil as well as the weak shocks emanating from the trailing edge of the airfoil. In contrast to that, the refinement of the mesh produced by employing the weighted-residual indicator $\eta_\kappa^{(\mathrm{I})}$, see Figures 8(c) and (d), is very concentrated close to the airfoil. In particular, the bow shock is mainly resolved in a small region upstream of the profile only, and there is even no refinement at all at the position of the bow shock beyond six chord lengths above and below the profile. Furthermore, the weak shocks emanating from the trailing edge are not resolved and there is no refinement in the wake of the flow beyond three chord lengths behind the profile. Instead, the refinement of the mesh is concentrated near the leading edge of the profile and in the boundary layer of the flow. All other parts of the computational domain are recognized by the weighted-residual indicator to be of minor importance for the accuracy of the c_{dp} target quantity. In fact, the dual (adjoint) solution, see Figures 9 and 10, includes the crucial information concerning which local residuals contribute to the error in the target quantity and to what extent. Herewith, it offers all necessary information of error transport and accumulation. Finally, the weighted-residual indicator including the information of the dual solution, mark only those parts of the domain for refinement where residuals of the flow solution significantly contribute to the error of the target quantity, i.e. all parts which are important for the accurate approximation of the target quantity.

8.2.3. Comparison of the approximate error representation for viscous and inviscid flow. We recall, that the approximate error representation, $-\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa\in\mathcal{T}_h} \hat{\eta}_\kappa$, cf. (34), was obtained by replacing the exact solution $\mathbf{z}$ to the (exact) dual problem (27) in the error representation (29) by the solution $\hat{\mathbf{z}}_h$ to an approximate dual problem which is linearized about the discrete flow solution $\mathbf{u}_h$ and discretized. In order to discuss the error introduced by this replacement, we split the (exact) error representation (28) in three terms as follows:

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \\ &= -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \hat{\mathbf{z}}) - \mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}} - \hat{\mathbf{z}}_h) - \mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) \end{aligned} \tag{35}$$

where the first term represents the error incurred through linearization of the dual problem, the second term is the error due to the numerical approximation of the (linearized) dual solution

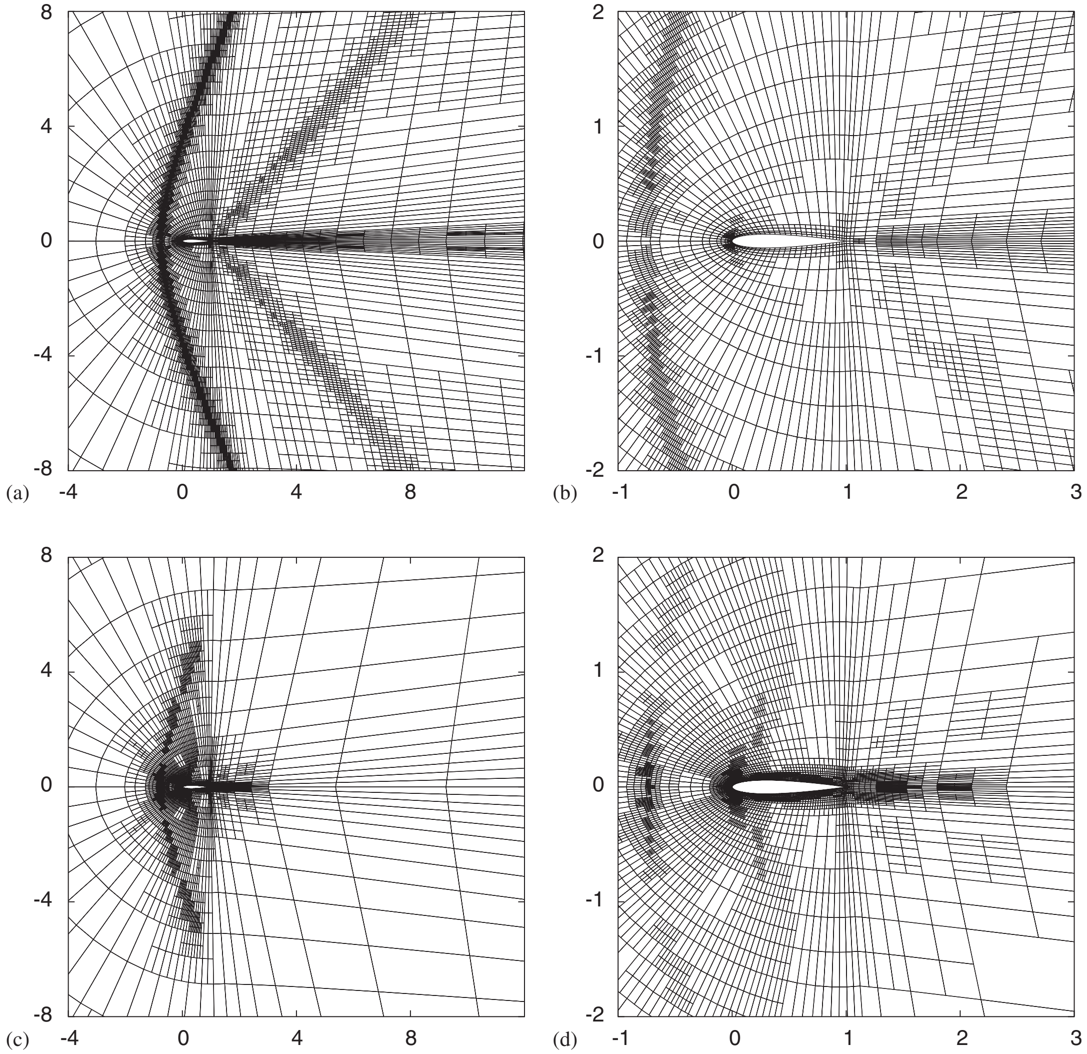


Figure 8. $M = 1.2$, $Re = 1000$, $\alpha = 0^\circ$ flow around the NACA0012 airfoil: (a) and (b) residual-based refined mesh of 17 670 elements with 282 720 degrees of freedom and $|J_{c_{\mathrm{dp}}}(\mathbf{u}) - J_{c_{\mathrm{dp}}}(\mathbf{u}_h)| = 1.9 \times 10^{-3}$; (c) and (d) goal-oriented refined mesh for c_{dp}: mesh of 10 038 elements with 160 608 degrees of freedom and $|J_{c_{\mathrm{dp}}}(\mathbf{u}) - J_{c_{\mathrm{dp}}}(\mathbf{u}_h)| = 1.6 \times 10^{-4}$.

and the last term is the approximate error representation formula which is actually computed in practice. The error $\mathscr{N}(\mathbf{u}_h, \hat{\mathbf{z}} - \hat{\mathbf{z}}_h)$ due to the discretization of the dual problem will be of higher-order than the approximate error representation, provided that the dual solution is sufficiently regular and is approximated by higher order polynomials. The linearization error term $\mathscr{N}(\mathbf{u}_h, \mathbf{z} - \hat{\mathbf{z}})$ is expected to be small in cases when the analytical solution $\mathbf{u}$ is smooth.

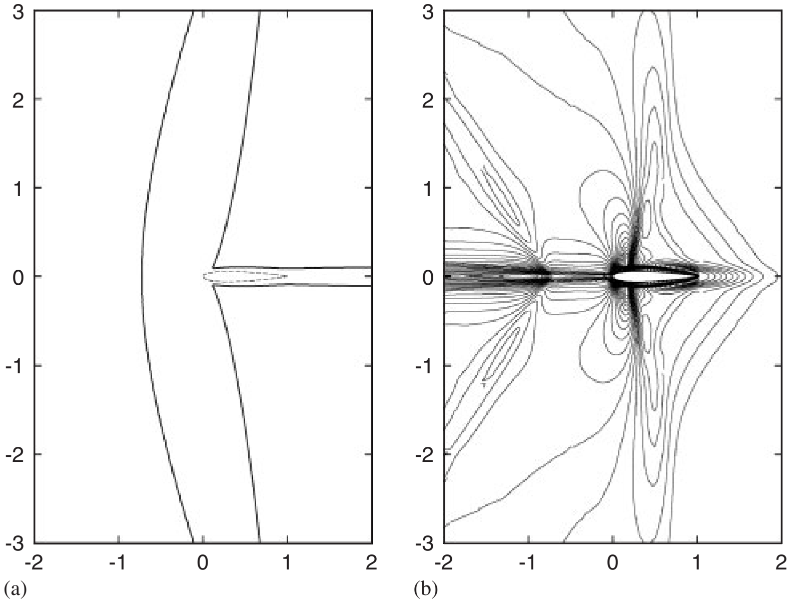


Figure 9. Viscous flow at $M = 1.2$, $Re = 1000$, $\alpha = 0°$ around the NACA0012 airfoil: (a) sonic isolines of the flow solution; and (b) isolines of the $\hat{z}_1$ component of the computed adjoint solution $\hat{\mathbf{z}}$.

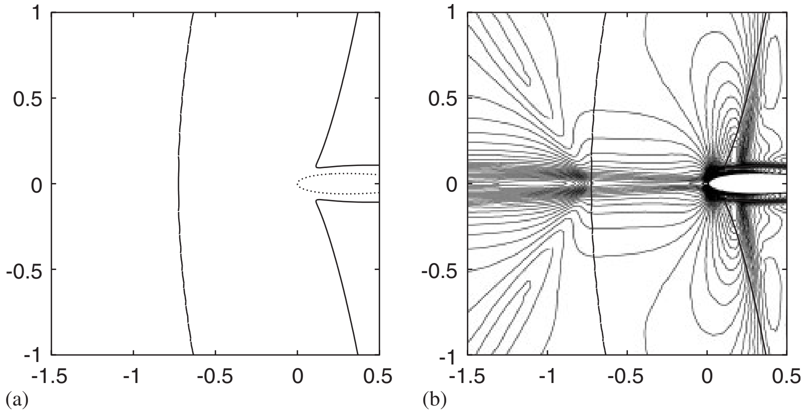


Figure 10. Viscous flow at $M = 1.2$, $Re = 1000$, $\alpha = 0°$ around the NACA0012 airfoil: zoom of: (a) sonic ($M = 1$) isolines of the flow solution; and (b) together with $\hat{z}_1$ isolines.

Rewriting the linearization term using $\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0$ for any $\mathbf{v}_h \in \mathbf{V}_h$

$$\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \hat{\mathbf{z}}) = \mathcal{N}(\mathbf{u}_h, (\mathbf{z} - \hat{\mathbf{z}}) - I_h(\mathbf{z} - \hat{\mathbf{z}})) = \mathcal{N}(\mathbf{u}_h, \mathbf{z} - I_h\mathbf{z}) - \mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}} - I_h\,\hat{\mathbf{z}}) \qquad (36)$$

where $I_h\mathbf{z} \in \mathbf{V}_h$ denotes a discrete approximation of $\mathbf{z}$, we see, that the linearization term can also be expected to be small when the dual solution is smooth.

We note that the supersonic flow considered in this example includes an extensive bow shock where the solution $\mathbf{u}$ is not smooth. In fact, all information of the flow crosses the shock from upstream before reaching the airfoil where the force coefficients are evaluated. *Vice versa*, all information of the dual problem, travelling in opposite direction along the flow characteristics, crosses the shock from downstream. According to the discussion above and given that $\mathbf{u}$ is not smooth, the linearization error term, $\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \hat{\mathbf{z}})$, can only be expected to be small, when the dual solution is smooth. In this case, also the discretization error of the dual solution, $\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}} - \hat{\mathbf{z}}_h)$, will be small, provided the dual solution is approximated with higher-order polynomials.

As can be seen in Figures 9 and 10, the dual solution is in fact smooth in most parts of the domain. In particular at the position of the shock where the linearization error of flow solution is large, the dual solution is smooth. This, as already discussed above, is necessary for the linearization error term and the discretization error of the dual solution to be small, and finally for the approximate error representation to be close to the true error in the target quantity.

In fact, as shown in Table II for the viscous flow case considered, the approximate error representation represents a remarkably close estimate of the true error in the target quantity. In particular, the accuracy of the error estimation presented in Table II is significantly better than that presented in a previous publication [7], for a supersonic inviscid flow around a BAC3-11 airfoil with a target quantity representing a (regularized) point evaluation. This difference clearly is attributed to both, a smaller linearization error of the flow solution due to a smoother solution at a viscous shock, in contrast to at an inviscid shock, and to a smaller discretization error of a dual solution which is smoother for a dual problem being connected to a target quantity, $J(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u})$, given by an integration of flow variables over a line (profile), than the solution to a dual problem which is connected to a (regularized) point evaluation.

In order to give a direct comparison with the viscous flow example at $M = 1.2$, $Re = 1000$ and $\alpha = 0°$, we consider the corresponding inviscid test case, with $M = 1.2$, $\alpha = 0°$ and the c_{dp} target quantity, in the following. Given the same freestream flow conditions and the same target quantity, this comparison shall give us a closer insight to a possibly increased linearization and discretization error of the dual solution for the inviscid flow in comparison to the viscous flow problem.

Given the c_{dp} reference value for the inviscid computation based on fine grid computations to be $J_{c_{\mathrm{dp}}}(\mathbf{u}) \approx 0.09549$, the data of the adaptive refinement targeted at the accurate approximation of this value is given in Table III. Here, we see that the approximate error

Table III. Inviscid $M = 1.2$, $\alpha = 0°$ flow around the NACA0012 airfoil: adaptive algorithm for the accurate approximation of c_{dp}.

# elements	# DoFs	$J(\mathbf{u}) - J(\mathbf{u}_h)$	$\sum_\kappa \hat{\eta}_\kappa$	θ
768	12 288	−1.184e-02	−2.218e-03	0.19
1260	20 160	−4.214e-03	−6.197e-03	1.47
2151	34 416	−9.285e-04	−5.458e-04	0.59
3687	58 992	−2.472e-04	−3.666e-04	1.48
6165	98 640	−9.057e-05	−9.796e-05	1.08
10 605	169 680	−6.057e-05	−6.150e-05	1.02

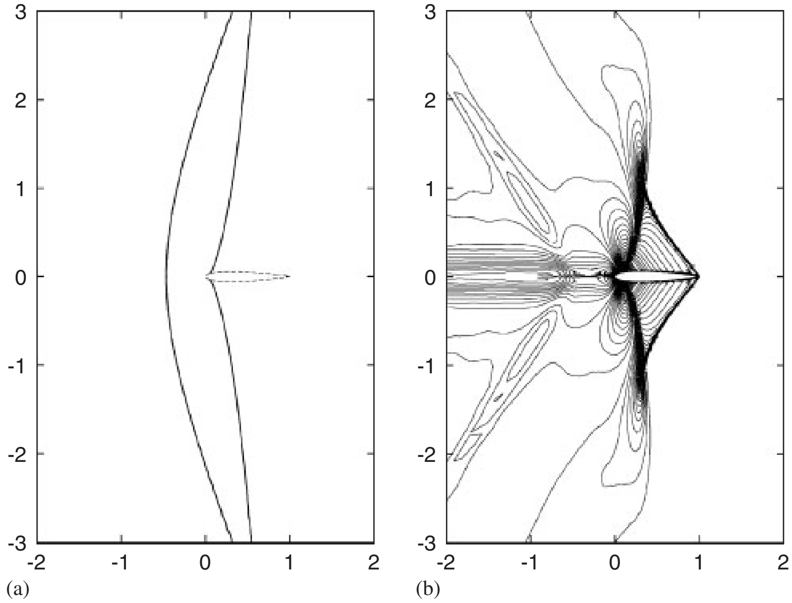


Figure 11. Inviscid flow at $M = 1.2$ and $\alpha = 0°$ around the NACA0012 airfoil: (a) sonic isolines of the flow solution; and (b) isolines of the $\hat{z}_1$ component of the computed adjoint solution $\hat{\mathbf{z}}$.

representation $\sum_{\kappa\in\mathscr{T}_h} \hat{\eta}_\kappa$ is still reasonably close the true error. But, there is a significant difference in the range of effectivity indices θ, which in the inviscid case is about 0.6–1.5 from the second mesh onwards, see Table III, whereas in the viscous case this is about 0.94–1.11, cf. Table II. This difference in the accuracy of the approximate error representation can be attributed to the increased linearization error at the (inviscid) shock and to a significantly less smooth dual solution in comparison to the viscous flow case. In fact, in Figures 11 and 12 we see that there are discontinuities of the dual solution near the trailing edge of the profile due to the supersonic nature of the flow in this part of the domain. Furthermore, there are discontinuities evolving close to the sonic lines of the flow above and below the profile. In addition, we see a number of wiggles upstream the airfoil which are not observed in the dual solution to the viscous flow problem, see Figures 9 and 10. This additional roughness is introduced from the primal solution, which is smoothed-out by the numerical (and artificial) viscosity of the discontinuous Galerkin scheme only, and as being an inviscid flow solution, lacks of any physical smoothing introduced by the governing differential equations. This results in the respective dual solution being significantly more rough than the dual solution to the (smoother) viscous flow solution. Finally, the dual solution shows some wiggles right at the position of the shock. Here we have a coincidence in place of a large linearization error of the flow solution and an oscillatory dual solution, which results in some of the approximate error representations in Table III being less close to the true error, which is also indicated by the respective effectivity indices θ noticeably differing from one.

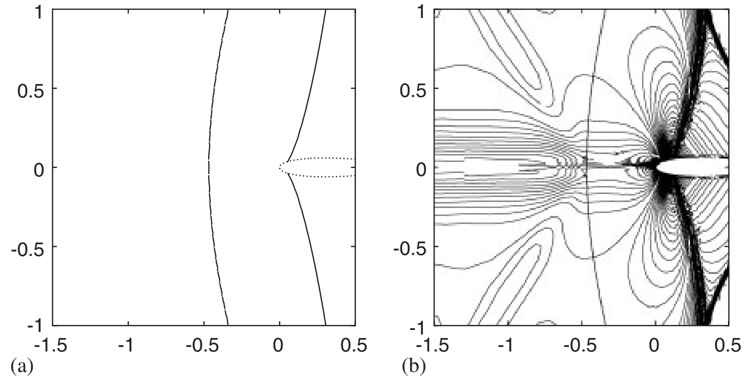


Figure 12. Inviscid flow at $M = 1.2$ and $\alpha = 0°$ around the NACA0012 airfoil: zoom of: (a) sonic $(M = 1)$ isolines of the flow solution; and (b) together with $\hat{z}_1$ isolines.

9. CONCLUSION

In this article we have extended an artificial viscosity stabilization for the discontinuous Galerkin discretization of the compressible Euler equation to the compressible Navier–Stokes equations. Furthermore, this viscosity term has been extended from isotropic to anisotropic viscosity which was found to be necessary on highly stretched meshes. This stabilization preserves conservativity, is relatively simple in structure involving element terms, only, without introducing additional dependencies to neighbouring elements. As the artificial viscosity term depends on the residual of the governing differential equations, it vanishes when evaluated for the exact solution, resulting the scheme to be consistent, i.e. the Galerkin orthogonality of the DG discretization is still valid after addition of the artificial viscosity term. We demonstrated the performance of this discontinuity stabilization on a standard test problem considered in literature. Furthermore, we proposed and demonstrated the use of a Newton–GMRES algorithm for solving the nonlinear discrete problems.

Based on a duality argument, we derived an error representation for the error measured in terms of target quantities like the force coefficient of an airfoil immersed in a viscous or inviscid fluid. The error representation includes primal residuals multiplied by the solution to a dual problem which couples to the target quantity under consideration. The dual (or adjoint) solution is in general not known and must be replaced by an approximate dual solution of a linearized dual problem. The resulting approximate error representation was shown for a supersonic flow example to be very accurate and was demonstrated to offer a reliable error estimation for the true error in terms of a target quantity of physical interest. We have demonstrated and were able to explain the difference in the accuracy of the approximate error representation for a viscous flow problem in a direct comparison to the one for an inviscid flow problem. Finally, we have demonstrated the performance of a goal-oriented adaptive refinement algorithm specifically tailored to the efficient computation of the quantity of interest. This way very efficient meshes have been produced on which the computed target quantities are significantly more accurate than on meshes of comparable size which were refined by a residual-based indicator that does not involve the information of error transport and accumulation inherent in the solution of the dual (adjoint) problem.

ACKNOWLEDGEMENTS

All computations have been performed using a DG flow solver based on the deal.II library [22].

REFERENCES

1. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics* 1997; **131**:267–279.
2. Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2d Euler equations. *Journal of Computational Physics* 1997; **138**:251–285.
3. Baumann C, Oden J. A discontinuous *hp* finite element method for the Euler and Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1999; **31**:79–95.
4. Baumann C, Oden J. An adaptive-order discontinuous Galerkin method for the solution of the Euler equations of gas dynamics. *International Journal for Numerical Methods in Engineering* 2000; **47**:61–73.
5. Dolejsi V. On the discontinuous Galerkin method for the numerical solution of the Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2004; **45**:1083–1106.
6. Hartmann R, Houston P. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: goal-oriented *a posteriori* error estimation. *International Journal of Numerical Analysis and Mathematical Modeling* 2006; **3**(2):141–162.
7. Hartmann R, Houston P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics* 2002; **183**:508–532.
8. Hartmann R, Houston P. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: method formulation. *International Journal of Numerical Analysis and Mathematical Modeling* 2006; **3**(1): 1–20.
9. van der Vegt J, van der Ven H. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows, I. General formulation. *Journal of Computational Physics* 2002; **182**:546–585.
10. Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin finite element method for conservation laws V: multidimensional systems. *Journal of Computational Physics* 1998; **141**:199–224.
11. Jaffre J, Johnson C, Szepessy A. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Mathematical Models and Methods in Applied Sciences* 1995; **5**:367–386.
12. Bassi F, Rebay S. Accurate 2d Euler computations by means of a high order discontinuous finite element method. *XIVth ICNMFD*, Bangalore, 11–15 July 1994, Lecture Notes in Physics. Springer: Berlin, 1996.
13. Bassi F, Rebay S, Mariotti G, Pedinotti S, Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In *2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, Antwerpen, Belgium, 5–7 March 1997, Decuypere R, Dibelius G (eds). Technologisch Instituut: 1997; 99–108.
14. Becker R, Rannacher R. A feed-back approach to error control in finite element methods: basic analysis and examples. *East–West Journal of Numerical Mathematics* 1996; **4**:237–264.
15. Becker R, Rannacher R. An optimal control approach to error estimation and mesh adaptation in finite element methods. *Acta Numerica* 2001; **10**:1–102.
16. Hartmann R. Adaptive FE methods for conservation equations. In *Hyperbolic Problems*: *Theory*, *Numerics*, *Applications*: *8th International Conference in Magdeburg*, February, March 2000, Freistühler H, Warnecke G (eds), *International Series of Numerical Mathematics*, vol. 141. Birkhäuser: Basel, 2001; 495–503.
17. Hartmann R. Adaptive finite element methods for the compressible Euler equations. *Ph.D. Thesis*, University of Heidelberg, 2002.
18. Hartmann R, Houston P. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM Journal on Scientific Computing* 2002; **24**:979–1004.
19. Arnold D, Brezzi F, Cockburn B, Marini D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; **39**(5):1749–1779.
20. Hartmann R. The role of the Jacobian in the adaptive discontinuous Galerkin method for the compressible Euler equations. In *Analysis and Numerics for Conservation Laws*, Warnecke G (ed.). Springer: Berlin, 2005; 301–316.
21. Bristeau MO, Glowinski R, Periaux J, Viviand H. Numerical simulation of compressible Navier–Stokes flows. *Numerical Simulation of Compressible Navier–Stokes Flows. Notes on Numerical Fluid Mechanics*, vol. 18. Vieweg-Verlag: Braunschweig, 1986. *Proceedings of the GAMM Workshop held at INRIA*, France, 4–6 December 1985.
22. Bangerth W, Hartmann R, Kanschat G. deal.II *Differential Equations Analysis Library*, *Technical Reference* (1st edn, 1999; 5.1 edn, 2004). http://www.dealii.org/