INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:2111–2138
Published online 4 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1608

Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations

Tobias Leicht[1, 2] and Ralf Hartmann[1, 2, *, †]

[1]*Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany*
[2]*Institute of Scientific Computing, TU Braunschweig, Germany*

SUMMARY

We derive and implement two types of anisotropic indicators which can be used within an anisotropic refinement algorithm for second but also for higher-order discontinuous Galerkin discretizations. Although the first type of indicator employs the possible inter-element discontinuities of the discrete functions, the second type of indicator estimates the approximation error in terms of second but possibly also higher-order derivatives. We implement a simple extension of these indicators to systems of equations which performs similar to the so-called metric intersection used to combine the metric information of several solution components and is applicable to higher-order discretizations as well. The anisotropic indicators are incorporated into an adaptive refinement algorithm which uses state-of-the-art residual-based or adjoint-based indicators for goal-oriented refinement to select the elements to be refined, whereas the anisotropic indicators determine which anisotropic case the selected elements shall be refined with. We demonstrate the performance of the anisotropic refinement algorithm for sub-, trans- and supersonic, inviscid and viscous compressible flows around a NACA0012 airfoil. Copyright © 2007 John Wiley & Sons, Ltd.

Received 8 February 2007; Revised 6 July 2007; Accepted 19 July 2007

KEY WORDS: discontinuous Galerkin method; anisotropic mesh refinement; adaptive mesh refinement; compressible flows

1. INTRODUCTION

Since their introduction by Reed and Hill in 1973, see also [1, 2], there has been a considerable progress in the development and analysis of discontinuous Galerkin (DG) methods, see [3] for an

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

Contract/grant sponsor: Helmholtz Association of German Research Centres

overview. In particular, considerable work has been devoted to the *a posteriori* error estimation and (goal-oriented) adaptive mesh refinement for compressible flows. Most of the work in this field has been concerned with isotropic refinement [4–6].

However, flow phenomena may exhibit a strong directional behavior. In particular, in boundary layers or interior layers like shocks, the flow variables change rapidly in the direction orthogonal to the layer, whereas the change parallel to the layer is much smaller. Highly stretched elements should be used for an optimal resolution of these features. Starting from a coarse initial mesh, such elements can be obtained by an anisotropic refinement which splits only some of an element's edges. In contrast to that, isotropic refinement splits all an element's edges, which may result in excessively small edge lengths parallel to the layers which would not be required for the overall accuracy of the flow solution.

Considerable work has been devoted to the anisotropic refinement for linear finite elements on triangular meshes where the information of an approximated Hessian-based mesh metric field is used within remeshing algorithms, see, for example, the work by Formaggia *et al.* [7], Frey and Alauzet [8, 9], Huang [10] or Sahni *et al.* [11]. Here, the metric field approximates the interpolation error of the solution and is used to determine both the local mesh density and the local mesh anisotropy. However, it has been shown in [4–6] that *ad hoc* indicators like second derivatives approximating the interpolation error but not taking into account any error transportation and accumulation effects may lead to highly *non*-optimal meshes. In contrast to that, error transportation and accumulation are naturally included in adjoint-based indicators involving the solution to an adjoint problem connected to the target quantities under consideration. Using these so-called *goal-oriented* indicators to determine the local mesh density results in meshes that may be considered optimal in the sense that they are specifically tailored to the accurate approximation of the target quantities. Venditti and Darmofal [12] have combined the directional information of the metric approach with a scaling based on adjoint-based error indicators, resulting in the so-called weighted metrics. Still, this technique requires a complete remeshing and is therefore not applicable in the framework of adaptive local refinement based on a hierarchical structure of triangulations. Furthermore, the presented results are restricted to a second-order finite volume scheme.

For determining an optimal local mesh anisotropy, Sun and Wheeler suggest several global refinements, each using anisotropic refinement in a fixed direction for all elements. Having re-computed the solution on each of the resulting discretizations, error indicators are evaluated for the refined elements. Choosing, individually for each element, the refinement strategy that results in a higher reduction of the estimated error yields quite optimal refined meshes. However, the effort of solving the problem several times on globally refined meshes is extensive. Using the obtained meshes for several time steps in the calculation of transient problems like in [13], the effort might be affordable, whereas it does not seem to be justified for the numerical approximation of steady-state flow problems.

Trying to reduce the computational effort of this approach, Houston *et al.* [14], working on the numerical solution of the advection–diffusion–reaction equations, suggest to solve local problems. Only the element under consideration is refined according to the different anisotropic refinement cases. Then, for each of these alternatives an improved numerical primal and adjoint solution is computed locally on this element, assuming, that the solutions on the surrounding elements are not changed. By comparing the reduction in the estimated error of the different alternatives, the best refinement case is chosen. This technique does not depend on any *a priori* information like interpolation errors. Furthermore, it is possible to use goal-oriented error indicators, not only for selecting the elements to be refined but also for determining the anisotropic refinement direction.

However, the evaluation of these indicators—including the local recomputation of both the primal and adjoint problem on each element considered for refinement—is still quite expensive.

The purpose of this work, see also [25] is to derive anisotropic indicators that come computationally almost for free. In particular, no auxiliary problems shall be solved for obtaining anisotropic refinement information. Furthermore, these indicators shall be applicable to second but also higher-order DG discretizations and they shall be easily combined with reliable error indicators which select the elements to be refined. In particular, we derive and implement two types of anisotropic indicators which can be used within an anisotropic refinement algorithm for DG discretizations. The first type of indicators associates the jump of the solution across opposite element faces with the approximation quality orthogonal to the faces and herewith employs a characteristic property of DG discretizations, namely the possible discontinuity of the discrete solution between neighboring elements. The second type of anisotropic indicators estimates the interpolation or approximation error in terms of derivatives of the solution. For a second-order discretization, this corresponds to the second derivatives, i.e. the Hessian; for higher-order discretizations, however, this corresponds to higher-order derivatives.

We adopt the combined approach of using different indicators for selecting the elements to be refined and for choosing the anisotropic refinement case. In particular, we employ residual-based and adjoint-based indicators for goal-oriented refinement to select the elements to be refined. In a second step, the discrete solution is analyzed using one of the two anisotropic indicators to decide whether to use isotropic or anisotropic refinement on the selected elements and in the latter case to decide upon which anisotropic refinement cases to be used.

This paper is structured as follows. We begin by introducing the compressible Euler and Navier–Stokes equations in Section 2 and the DG discretization of these equations in Section 3. Then, in Section 4 we recall the *a posteriori* error estimation and the adjoint-based and residual-based indicators derived in [6]. After introducing the anisotropic refinement cases in Section 5, we introduce the two types of anisotropic indicators suitable for second- and higher-order DG discretizations in Section 6. In Section 7, we give some numerical examples demonstrating the performance of the anisotropic refinement algorithm developed and draw some conclusions in Section 8.

2. THE COMPRESSIBLE EULER AND NAVIER–STOKES EQUATIONS

We consider the two-dimensional steady-state compressible Navier–Stokes equations,

$$\nabla\cdot(\mathscr{F}^{c}(\mathbf{u})-\mathscr{F}^{v}(\mathbf{u},\nabla\mathbf{u}))\equiv\frac{\partial}{\partial x_i}\mathbf{f}_i^{c}(\mathbf{u})-\frac{\partial}{\partial x_i}\mathbf{f}_i^{v}(\mathbf{u},\nabla\mathbf{u})=0 \quad \text{in } \Omega \tag{1}$$

where Ω is a bounded connected domain in $\mathbb{R}^2$ with boundary Γ. The vector of conservative variables $\mathbf{u}$ and the convective fluxes $\mathscr{F}^{c}(\mathbf{u}):=(\mathbf{f}_1^{c}(\mathbf{u}),\mathbf{f}_2^{c}(\mathbf{u}))$ are given by

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

where ρ, $\mathbf{v}=(v_1, v_2)^{\mathrm{T}}$, p and E denote the density, velocity vector, pressure and specific total energy, respectively. Additionally, H is the total enthalpy given by $H=E+p/\rho=e+\frac{1}{2}\mathbf{v}^2+p/\rho$, where e is the specific static internal energy, and the pressure is determined by the equation of state of an ideal gas, $p=(\gamma-1)\rho e$, 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 at moderate temperatures, $\gamma=1.4$.

Furthermore, the viscous fluxes $\mathscr{F}^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u}):=(\mathbf{f}_1^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u}),\mathbf{f}_2^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u}))$ are defined by

$$\mathbf{f}_1^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u})=\begin{bmatrix}0\\ \tau_{11}\\ \tau_{21}\\ \tau_{1j}v_j+\mathscr{K}T_{x_1}\end{bmatrix} \quad\text{and}\quad \mathbf{f}_2^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u})=\begin{bmatrix}0\\ \tau_{12}\\ \tau_{22}\\ \tau_{2j}v_j+\mathscr{K}T_{x_2}\end{bmatrix} \tag{3}$$

respectively, where $\mathscr{K}$ is the thermal conductivity coefficient and T denotes the temperature.

Finally, for a Newtonian fluid the viscous stress tensor is defined as

$$\tau=\mu(\nabla\mathbf{v}+(\nabla\mathbf{v})^{\mathrm{T}}-\tfrac{2}{3}(\nabla\cdot\mathbf{v})I) \tag{4}$$

where μ is the dynamic viscosity coefficient, and the temperature T is given by $e=c_vT$; thus $\mathscr{K}T=\mu\gamma/Pr(E-\frac{1}{2}\mathbf{v}^2)$, where $Pr=\mu c_p/\mathscr{K}$ is the Prandtl number with $Pr=0.72$ for dry air.

Finally, we note that the viscous fluxes $\mathbf{f}_i^{\mathrm{v}}$, $i=1,2$, are homogeneous with respect to the gradient of conservative variables $\nabla\mathbf{u}$, i.e. $\mathbf{f}_i^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u})=G_{ij}(\mathbf{u})\partial\mathbf{u}/\partial x_j$, $i=1,2$, where G denotes the homogeneity tensor given by $G_{ij}(\mathbf{u})=\partial\mathbf{f}_i^{\mathrm{v}}(\mathbf{u},\nabla\mathbf{u})/\partial u_{x_j}$, for $i,j=1,2$, cf. [15]. Thereby, the compressible Navier–Stokes equations (1) can be rewritten in the following form:

$$\frac{\partial}{\partial x_i}\left(\mathbf{f}_i^{\mathrm{c}}(\mathbf{u})-G_{ij}(\mathbf{u})\frac{\partial\mathbf{u}}{\partial x_j}\right)=0 \quad \text{in } \Omega \tag{5}$$

Given that $\Omega\subset\mathbb{R}^2$ is a bounded domain, the system of conservation equations (5) must be supplemented by appropriate boundary conditions. We distinguish between supersonic inflow (Dirichlet), subsonic inflow, subsonic outflow, supersonic outflow (Neumann) and solid wall boundaries, denoted by $\Gamma_{\mathrm{D,sup}}$, $\Gamma_{\mathrm{D,sub\text{-}in}}$, $\Gamma_{\mathrm{D,sub\text{-}out}}$, Γ_{N} and Γ_{W}, respectively.

For solid wall boundaries and *viscous* flows, we consider *isothermal* and *adiabatic* boundary conditions. To this end, we decompose $\Gamma_{\mathrm{W}}=\Gamma_{\mathrm{W,iso}}\cup\Gamma_{\mathrm{W,adia}}$ and impose

$$\mathbf{v}=\mathbf{0} \quad \text{on } \Gamma_{\mathrm{W}}, \qquad T=T_{\mathrm{wall}} \quad \text{on } \Gamma_{\mathrm{W,iso}}, \qquad \mathbf{n}\cdot\nabla T=0 \quad \text{on } \Gamma_{\mathrm{W,adia}} \tag{6}$$

where T_{wall} is a given wall temperature. Inviscid flows are described by the compressible Euler equations, see (1) with vanishing viscous fluxes $\mathbf{f}_i^{\mathrm{v}}$, $i=1,2$. For solid wall boundaries in inviscid flows, *reflective* (or slip wall) boundary conditions are given by $\mathbf{v}\cdot\mathbf{n}=0$ on $\Gamma_{\mathrm{W}}=\Gamma_{\mathrm{refl}}$.

3. THE DISCONTINUOUS GALERKIN DISCRETIZATION

Before giving the discretization of the flow equations, we introduce some notation. We assume that Ω can be subdivided into shape-regular meshes $\mathscr{T}_h=\{\kappa\}$ consisting of quadrilateral elements κ. Here, h denotes the piecewise constant mesh function defined by $h|_\kappa\equiv h_\kappa=\mathrm{diam}(\kappa)$ for all $\kappa\in\mathscr{T}_h$. It can further be assumed that each $\kappa\in\mathscr{T}_h$ is an image of a fixed reference element $\hat{\kappa}$,

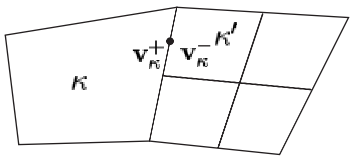

Figure 1. Definition of the interior and outer traces $\mathbf{v}_\kappa^\pm$ w.r.t. element κ.

that is, $\kappa = \sigma_\kappa(\hat{\kappa})$ for all $\kappa \in \mathscr{T}_h$. Here, we consider only the case when $\hat{\kappa}$ is the open unit square. Furthermore, the mapping σ_κ of the reference element $\hat{\kappa}$ to the element κ in real space is assumed to be bijective and smooth, with the eigenvalues of its Jacobian matrix being bounded from below and above. For elements in the interior of the domain, $\partial\kappa \cap \Gamma = \emptyset$, the mapping σ_κ is bilinear. In order to represent curved boundaries, mappings can be used that include polynomials of higher degree on boundary elements. For both cases we assume that mapping σ_κ can be decomposed into an affine part $\bar{\sigma}_\kappa$ and a higher-order part $\tilde{\sigma}_\kappa$, such that $\kappa = \sigma_\kappa(\hat{\kappa}) = \tilde{\sigma}_\kappa(\bar{\sigma}_\kappa(\hat{\kappa}))$. On the reference element $\hat{\kappa}$ spaces of tensor product polynomials of degree $p \geqslant 0$ are defined as follows:

$$\mathscr{Q}_p(\hat{\kappa}) = \operatorname{span}\{\hat{x}^\alpha : 0 \leqslant \alpha_i \leqslant p,\ 0 \leqslant i \leqslant 2\}$$

where α denotes a multi-index and $\hat{x}^\alpha = \prod_{i=1}^2 \hat{x}_i^{\alpha_i}$. Finally, we define the discrete function space $\mathbf{V}_h^p$ of discontinuous vector-valued piecewise polynomial functions of degree $p \geqslant 0$ by

$$\mathbf{V}_h^p = \{\mathbf{v}_h \in [L_2(\Omega)]^m : \mathbf{v}_h|_\kappa \circ \sigma_\kappa \in [\mathscr{Q}_p(\hat{\kappa})]^m\}$$

Suppose that $\mathbf{v}|_\kappa \in [H^1(\kappa)]^m$ for each $\kappa \in \mathscr{T}_h$. Let κ and κ' be two adjacent elements in $\mathscr{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior edge $e = \partial\kappa \cap \partial\kappa' \in \Gamma_{\mathscr{I}}$, where $\Gamma_{\mathscr{I}}$ denotes the union of all interior edges of $\mathscr{T}_h$. By $\mathbf{v}_\kappa^\pm$ ($\mathbf{v}^\pm$ for short) we denote the traces of $\mathbf{v}$ taken from within the interior of κ and κ', respectively, see Figure 1. Furthermore, we define average and jump operators for vector- and matrix-valued functions. To this end, let $\mathbf{v}$ and $\underline{\tau}$ be vector- and matrix-valued functions, respectively, that are smooth inside each element $\kappa \in \mathscr{T}_h$. Then, the averages at $\mathbf{x} \in e \in \Gamma_{\mathscr{I}}$ are defined by $\{\mathbf{v}\} = (\mathbf{v}^+ + \mathbf{v}^-)/2$ and $\{\underline{\tau}\} = (\underline{\tau}^+ + \underline{\tau}^-)/2$. Similarly, the jump at $\mathbf{x} \in e$ is given by $\underline{[\![\mathbf{v}]\!]} = \mathbf{v}^+ \otimes \mathbf{n}_\kappa + \mathbf{v}^- \otimes \mathbf{n}_{\kappa'}$. On a boundary edge $e \subset \Gamma$, the operators are set to $\{\mathbf{v}\} = \mathbf{v}$, $\{\underline{\tau}\} = \underline{\tau}$ and $\underline{[\![\mathbf{v}]\!]} = \mathbf{v} \otimes \mathbf{n}$. For matrices $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m\times n}$, $m, n \geqslant 1$, the standard notation $\underline{\sigma} : \underline{\tau} = \sum_{k=1}^m \sum_{l=1}^n \sigma_{kl}\tau_{kl}$ is used; 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 $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k w_l$.

Then, according to [15], the symmetric interior penalty discontinuous Galerkin discretization of degree p ('DG(p) method' for short) of the compressible Navier–Stokes equations (1) is given by: find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$\begin{aligned}\mathscr{N}(\mathbf{u}_h, \mathbf{v}) \equiv &-\int_\Omega \mathscr{F}^{\mathrm{c}}(\mathbf{u}_h) : \nabla_h \mathbf{v}\, \mathrm{d}\mathbf{x} + \sum_{\kappa \in \mathscr{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathscr{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) \cdot \mathbf{v}^+\, \mathrm{d}s \\ &+ \int_\Omega \mathscr{F}^{\mathrm{v}}(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v}\, \mathrm{d}\mathbf{x} - \int_{\Gamma_{\mathscr{I}}} \{\mathscr{F}^{\mathrm{v}}(\mathbf{u}_h, \nabla_h \mathbf{u}_h)\} : \underline{[\![\mathbf{v}]\!]}\, \mathrm{d}s \\ &- \int_{\Gamma_{\mathscr{I}}} \{G^\top(\mathbf{u}_h)\nabla_h \mathbf{v}\} : \underline{[\![\mathbf{u}_h]\!]}\, \mathrm{d}s + \int_{\Gamma_{\mathscr{I}}} \delta \underline{[\![\mathbf{u}_h]\!]} : \underline{[\![\mathbf{v}]\!]}\, \mathrm{d}s + \mathscr{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) = 0 \end{aligned} \tag{7}$$

for all $\mathbf{v}$ in $\mathbf{V}_h^p$. Here, $\mathscr{H}(\mathbf{u}_h^+,\mathbf{u}_h^-,\mathbf{n})$ may be any Lipschitz continuous, consistent and conservative *numerical flux* function which depends on both the interior and outer trace of $\mathbf{u}_h$ on $\partial\kappa$, $\kappa\in\mathscr{T}_h$, and the unit outward normal $\mathbf{n}$ to $\partial\kappa$. Furthermore, δ denotes the discontinuity penalization matrix, which is set to $\delta=\mathrm{diag}\{\delta_i, i=1,\dots,4\}$, where $\delta_i|_e = C_{\mathrm{IP}}(\mu p^2/\tilde{h})$ for $e\subset\Gamma_{\mathscr{I}}\cup\Gamma$, and $\tilde{h}=\min(\mathrm{meas}(\kappa),\mathrm{meas}(\kappa'))/\mathrm{meas}(e)$ represents the element dimension orthogonal to the edge e of elements κ and κ' adjacent to e. Here, C_{IP} is a positive constant, which, for reasons of stability, must be chosen sufficiently large. For a wide range of problems $C_{\mathrm{IP}}=10$ is a good choice.

Finally, the boundary terms included in $\mathscr{N}_\Gamma(\mathbf{u}_h,\mathbf{v})$ are given by

$$\begin{aligned}\mathscr{N}_\Gamma(\mathbf{u}_h,\mathbf{v}) = &\int_\Gamma \mathscr{H}_\Gamma(\mathbf{u}_h^+,\mathbf{u}_\Gamma(\mathbf{u}_h^+),\mathbf{n}^+)\cdot\mathbf{v}^+\,\mathrm{d}s + \int_\Gamma \delta\left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right)\cdot\mathbf{v}^+\,\mathrm{d}s\\ &-\int_\Gamma \mathbf{n}\cdot\mathscr{F}_\Gamma^{\mathrm{v}}(\mathbf{u}_h^+,\nabla_h\mathbf{u}_h^+)\mathbf{v}^+\,\mathrm{d}s\\ &-\int_\Gamma (G_\Gamma^\top(\mathbf{u}_h^+)\nabla_h\mathbf{v}_h^+) : (\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+))\otimes\mathbf{n}\,\mathrm{d}s \qquad (8)\end{aligned}$$

where the viscous flux $\mathscr{F}_\Gamma^{\mathrm{v}}$ and the corresponding homogeneity tensor G_Γ at Γ are given by

$$\mathscr{F}_\Gamma^{\mathrm{v}}(\mathbf{u}_h,\nabla\mathbf{u}_h) = \mathscr{F}^{\mathrm{v}}(\mathbf{u}_\Gamma(\mathbf{u}_h),\nabla\mathbf{u}_h) = G_\Gamma(\mathbf{u}_h)\nabla\mathbf{u}_h = G(\mathbf{u}_\Gamma(\mathbf{u}_h))\nabla\mathbf{u}_h$$

Furthermore, on adiabatic boundaries $\Gamma_{\mathrm{adia}}\subset\Gamma_{\mathrm{W}}$, $\mathscr{F}_\Gamma^{\mathrm{v}}$ and G_Γ are modified such that $\mathbf{n}\cdot\nabla T=0$. For an adjoint consistent discretization of boundary terms, a numerical boundary flux function $\mathscr{H}_\Gamma$ is defined by

$$\mathscr{H}_\Gamma(\mathbf{u}_h^+,\mathbf{u}_\Gamma(\mathbf{u}_h^+),\mathbf{n}) = \mathbf{n}\cdot\mathscr{F}_\Gamma^{\mathrm{c}}(\mathbf{u}_h^+) = \mathbf{n}\cdot\mathscr{F}^{\mathrm{c}}(\mathbf{u}_\Gamma(\mathbf{u}_h^+))$$

where the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ is given according to the type of boundary condition imposed, see [15] for more detail in the case of sub- or supersonic in- and outflow boundaries as well as adiabatic or isothermal wall boundaries. Furthermore, at slip-wall boundaries in inviscid flows, we adopt the adjoint consistent treatment of the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ given by

$$\mathbf{u}_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & 1-n_1^2 & -n_1n_2 & 0\\ 0 & -n_1n_2 & 1-n_2^2 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\mathbf{u} \quad \text{on } \Gamma_{\mathrm{W}}$$

which originates from $\mathbf{u}$ by subtracting the normal velocity component of $\mathbf{u}$, i.e. $\mathbf{v}=(v_1,v_2)$ is replaced by $\mathbf{v}_\Gamma=\mathbf{v}-(\mathbf{v}\cdot\mathbf{n})\mathbf{n}$ which ensures that the normal velocity component vanishes, $\mathbf{v}_\Gamma\cdot\mathbf{n}=0$.

4. ERROR ESTIMATION AND ISOTROPIC REFINEMENT INDICATORS

In this section we first recall the *a posteriori* error estimation, cf. e.g. [16, 17] and the references cited therein, used to estimate the error in specific target quantities evaluated based on the numerical solution which may serve as a stopping criterion of an adaptive refinement algorithm when a prescribed accuracy is reached. Here, typical target quantities include the drag and lift coefficients for example. Furthermore, we recall the residual- and adjoint-based refinement indicators originally used for *isotropic* refinement in [4–6]. In this paper these indicators will be used to select the elements to be refined. Anisotropic indicators that can be used to determine the specific (anisotropic) refinement case on the selected elements will then be introduced in Section 6.

4.1. The a posteriori error estimation

In many applications the numerical flow solution is only the necessary step on the way to evaluate certain target quantities $J(\mathbf{u})$, which depend on the solution. Important target functionals in aerodynamic applications are the pressure-induced drag coefficient c_{dp}, and the viscous drag coefficient c_{df}, with

$$J_{c_{\mathrm{dp}}}(\mathbf{u}) = \frac{1}{q_\infty \bar{l}} \int_{\Gamma_{\mathrm{W}}} p\,\mathbf{n} \cdot \psi_d \,\mathrm{d}s, \quad J_{c_{\mathrm{df}}}(\mathbf{u}) = -\frac{1}{q_\infty \bar{l}} \int_{\Gamma_{\mathrm{W}}} (\underline{\tau}\mathbf{n}) \cdot \psi_d \,\mathrm{d}s \tag{9}$$

and the total drag coefficient c_d with $J_{c_d}(\mathbf{u}) = J_{c_{\mathrm{dp}}}(\mathbf{u}) + J_{c_{\mathrm{df}}}(\mathbf{u})$. Here $q_\infty = \frac{1}{2}\rho_\infty |\mathbf{v}_\infty|^2$ is the freestream dynamic pressure, $\bar{l}$ denotes a reference length and the subscripts $_\infty$ indicate freestream quantities. Furthermore, ψ_d is given by $\psi_d = (\cos(\alpha), \sin(\alpha))^\top$. Substituting ψ_d by $\psi_l = (-\sin(\alpha), \cos(\alpha))^\top$, the pressure-induced, the viscous and the total lift coefficients $J_{c_{\mathrm{lp}}}(\mathbf{u})$, $J_{c_{\mathrm{lf}}}(\mathbf{u})$ and $J_{c_l}(\mathbf{u})$, respectively, are obtained.

As has been shown in e.g. [6], the error in the target quantity can be represented as follows:

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \tag{10}$$

where $\mathbf{z}_h$ might be any discrete function in $\mathbf{V}_h^p$, and $\mathbf{z}$ is the solution of the adjoint problem with data coupling to the target functional under consideration. As the exact adjoint solution $\mathbf{z}$ is unknown, it must be approximated resulting in the following approximate error representation:

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx -\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) \tag{11}$$

where $\hat{\mathbf{z}}_h$ is the solution to the linearized/discretized adjoint problem: find $\hat{\mathbf{z}}_h \in \mathbf{V}_h^{\hat{p}}$ such that

$$\mathcal{N}'_{\mathbf{u}}[\mathbf{u}_h](\mathbf{w}_h, \hat{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \mathbf{V}_h^{\hat{p}} \tag{12}$$

where $\mathcal{N}'_{\mathbf{u}}[\mathbf{u}_h](\cdot, \mathbf{v})$ and $J'[\mathbf{u}_h](\cdot)$ denote the Frechét derivative of $\mathbf{u} \mapsto \mathcal{N}(\mathbf{u}, \mathbf{v})$ and $J(\cdot)$, respectively, evaluated at $\mathbf{u}_h$. As the approximate error representation (11) vanishes if $\hat{\mathbf{z}}_h$ is computed on $\mathbf{V}_h^p$, we solve the discrete adjoint problem on the same mesh but with a higher polynomial degree $\hat{p} > p$; usually $\hat{p} = p + 1$ is a good choice. We note that for an adjoint consistent discretization we could also use a discrete adjoint solution $\hat{\mathbf{z}}_h \in \mathbf{V}_h^p$ patchwise interpolated to $\mathbf{V}_{2h}^{p+1}$.

4.2. Adjoint-based indicators

The approximate error representation (11) can be written as the sum of all elements $\kappa \in \mathcal{T}_h$,

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx -\mathcal{N}(\mathbf{u}_h, \hat{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa \tag{13}$$

where the local error terms η_κ are the local elementwise contributions to the error measured in terms of the target functional $J(\cdot)$ and include element and face residuals, multiplied by the discrete adjoint solution, cf. [6]. The local error terms η_κ are the so-called adjoint-based indicators used for *goal-oriented* adaptive mesh refinement.

4.3. Residual-based indicators

Computing the solution to the discrete adjoint problem in order to evaluate the adjoint-based error indicators requires an additional computational effort, thus it might be desirable to derive indicators, which do not depend on the adjoint solution. Assuming that $\mathbf{z}\in[H^s(\Omega)]^4$, $2\leqslant s\leqslant p+1$, an upper bound of error in the target quantity can be derived, cf. [6],

$$|J(\mathbf{u})-J(\mathbf{u}_h)|\leqslant C\left(\sum_{\kappa\in\mathscr{T}_h}(\eta_\kappa^{\mathrm{res}})^2\right)^{1/2} \tag{14}$$

where the constant $C>0$ is independent of the mesh size h. The local residual-based indicators $\eta_\kappa^{\mathrm{res}}$, $\kappa\in\mathscr{T}_h$, are given by

$$\begin{aligned}\eta_\kappa^{\mathrm{res}} &= \|h_\kappa^s\mathbf{R}(\mathbf{u}_h)\|_{L_2(\kappa)}+\|h_\kappa^{s-1/2}(\mathscr{F}^{\mathrm{c}}(\mathbf{u}_h)\cdot\mathbf{n}_\kappa-\mathscr{H}(\mathbf{u}_h^+,\mathbf{u}_h^-,\mathbf{n}_\kappa))\|_{L_2(\partial\kappa\setminus\Gamma)}\\ &\quad+\|h_\kappa^{s-1/2}(\mathscr{F}^{\mathrm{c}}(\mathbf{u}_h)\cdot\mathbf{n}_\kappa-\mathscr{H}_\Gamma(\mathbf{u}_h^+,\mathbf{u}_\Gamma(\mathbf{u}_h^+),\mathbf{n}_\kappa))\|_{L_2(\partial\kappa\cap\Gamma)}\\ &\quad+\|h_\kappa^{s-3/2}G_{\cdot j}\underline{[\![\mathbf{u}_h]\!]}_j\|_{L_2(\partial\kappa\setminus\Gamma)}+\|h_\kappa^{s-1/2}[\![\mathscr{F}^{\mathrm{v}}(\mathbf{u}_h,\nabla\mathbf{u}_h)]\!]\|_{L_2(\partial\kappa\setminus\Gamma)}\\ &\quad+\|h_\kappa^{s-1/2}\delta\left(\mathbf{u}_h^+-\mathbf{u}_h^-\right)\|_{L_2(\partial\kappa\setminus\Gamma)}+\|h_\kappa^{s-1/2}\delta\left(\mathbf{u}_h^+-\mathbf{u}_\Gamma(\mathbf{u}_h^+)\right)\|_{L_2(\partial\kappa\cap\Gamma)}\\ &\quad+\|h_\kappa^{s-1/2}(\mathscr{F}^{\mathrm{v}}(\mathbf{u}_h^+,\nabla\mathbf{u}_h^+)-\mathscr{F}^{\mathrm{v}}_\Gamma(\mathbf{u}_h^+,\nabla\mathbf{u}_h^+))\cdot\mathbf{n}_\kappa\|_{L_2(\partial\kappa\cap\Gamma)}\\ &\quad+\|h_\kappa^{s-3/2}G_{\cdot j}(\mathbf{u}_h^+)[(\mathbf{u}_h^+-\mathbf{u}_\Gamma(\mathbf{u}_h^+))\otimes\mathbf{n}]_j\|_{L_2(\partial\kappa\cap\Gamma)}\end{aligned} \tag{15}$$

where $\mathbf{R}(\mathbf{u}_h)|_\kappa=-\nabla\cdot\mathscr{F}^{\mathrm{c}}(\mathbf{u}_h)+\nabla\cdot\mathscr{F}^{\mathrm{v}}(\mathbf{u}_h,\nabla\mathbf{u}_h)$. This indicator has been derived in [6].

4.4. Selecting the elements to be refined

Based on either of the indicators given in Sections 4.2 and 4.3, the $f_{\mathrm{r}}\cdot n_{\mathrm{el}}$ elements with the largest error indicators are flagged for refinement, whereas the $f_{\mathrm{c}}\cdot n_{\mathrm{el}}$ elements with the smallest error indicators are flagged for coarsening. Here, n_{el} is the total number of (active) elements and f_{r} and f_{c} are specified fractions of elements to be refined or coarsened, respectively. This strategy allows a good control over the number of elements after the refinement step. Values of $f_{\mathrm{r}}=0.2$ and $f_{\mathrm{c}}=0.1$ are good choices for many problems, as they represent a good trade-off between being small enough to obtain an optimal mesh and being large enough to reduce the total number of refinement steps.

5. THE REFINEMENT CASES

In the following, only h-refinement will be considered, in which an element is split into several children elements by splitting all or part of the edges. Future work will incorporate the possibility to perform p-refinement, i.e. a local variation of the polynomial degree, as well as the combination of

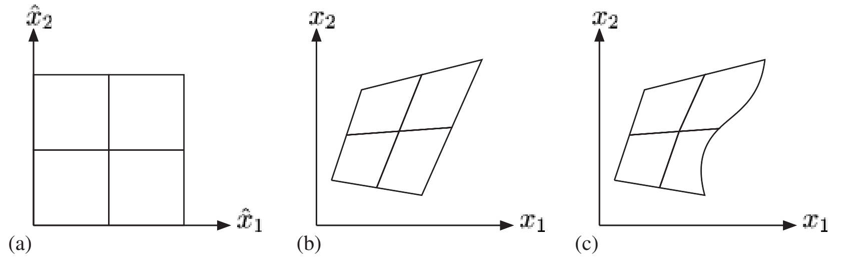

Figure 2. Isotropic refinement for quadrilateral elements: (a) reference element; (b) element in real space with straight boundaries; and (c) element in real space with a curved boundary.

both, the so-called *hp*-refinement. For all these techniques, in the case of pure refinement without local coarsening the function space of the old discretization is embedded in the newly created one, thus an interpolation of the approximate solution from the old to the new discretization is an easy task and can be done without loss of information. This ensures optimal initial conditions for the approximate solution, if an iterative scheme is used to solve the discretized equations and can thus accelerate the solution process. Furthermore, it is possible to reuse the embedded structure of the refined discretizations for multigrid algorithms.

In particular, we consider the following isotropic and anisotropic refinement cases:

Isotropic refinement: Using an isotropic refinement strategy, the edges of the reference element (unit square) are split exactly in half, resulting in four new child elements. Figure 2(a) gives a graphical representation of that process. For elements bounded only by straight edges, the same applies for the element in real-space coordinates, as shown in Figure 2(b). For elements on the boundary with a curved boundary edge, the new vertex on the boundary edge may not be exactly in the middle of the edge. Furthermore, the position of the middle vertex is not obvious and has to be defined, usually by means of a weighted mean value of both the old corner vertices and the new vertices on the edges. Figure 2(c) shows an example of the refinement of an element with curved edges. It has to be noted that generally the curved boundary is approximated by a piecewise polynomial of fixed degree. Using more pieces after the refinement, the boundary representation can change slightly. In this case the embedding of function spaces is not exact anymore. However, for the sake of simplicity this effect is not shown in Figure 2(c).

Anisotropic refinement: The anisotropic refinement considered in this work is based on splitting only a part of the edges in half. In order to result in quadrilaterals for the children as well, sets of parallel edges on the reference element have to be either split or kept unchanged. Out of the two distinct sets of parallel edges, one set is split, resulting in two new elements. As a convention, anisotropic refinement cases are named according to the axes, along which the edges are split in half, thus `cut_x` describes an element that is split along the $\hat{x}_1$-axis, whereas `cut_y` refines an element along the $\hat{x}_2$-axis. Figures 3 and 4 visualize the anisotropic refinement cases `cut_x` and `cut_y` for the reference element as well as for elements in real space. The isotropic refinement case `cut_xy` has been shown in Figure 2.

It has to be noted that the mapping between reference element and real-space element depends on the order of corner vertices. Thus, it is possible that the refinement case `cut_x` on the reference element corresponds to the case shown in Figure 3(b) or that shown in Figure 4(b), depending on the mapping. Because of that, any decision concerning a refinement case has to consider the mapping, as the actual refinement is done using the reference element.

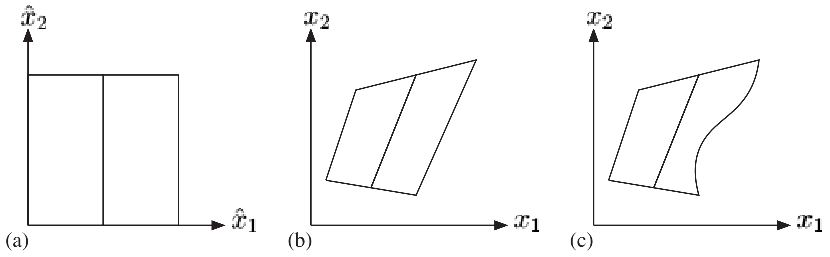

Figure 3. Anisotropic refinement (`cut_x`) for quadrilateral elements: (a) reference element; (b) element in real space with straight boundaries; and (c) element in real space with a curved boundary.

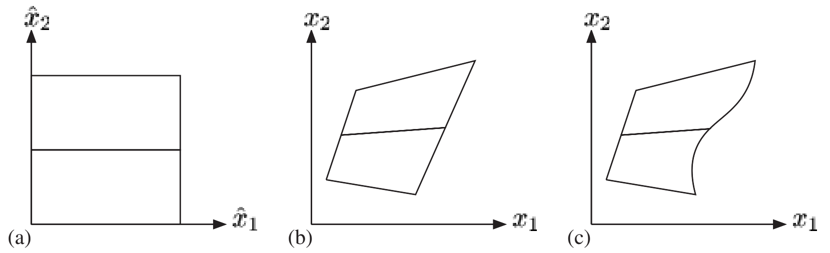

Figure 4. Anisotropic refinement (`cut_y`) for quadrilateral elements: (a) reference element; (b) element in real space with straight boundaries; and (c) element in real space with a curved boundary.

6. ANISOTROPIC REFINEMENT INDICATORS

The anisotropic indicators described in the following sections use two distinct ideas. On the one hand, jumps of the discontinuous solution can be analyzed in order to obtain knowledge of the anisotropic features of the solution; on the other hand, this information can be obtained using approximations of derivatives of the solution.

6.1. Jump indicator

The most characteristic feature of DG methods is the possible discontinuity of its discrete solutions. In fact, a discrete solution may have jumps across the faces between neighboring elements, whereas it is smooth inside each element. These jumps allow some flexibility in approximating the local properties of the solution. In smooth parts of the solution, these jumps tend to zero with successive mesh refinement as the solution is approximated with less error. Based on this observation it seems justified to assume that a large jump indicates a larger error as compared to a smaller jump. In view of an anisotropic evaluation, a large jump over a face indicates that the mesh size perpendicular to this face is too coarse to sufficiently resolve the solution. In this sense, inter-element jumps can be used to derive an anisotropic indicator that uses information which is specific to the numerical method used to solve the problem. Near discontinuities of the solution, like shocks, the jumps might not tend to zero under mesh refinement. However, in this case a large jump detects this discontinuity and suggests a refinement improving the resolution orthogonal to this feature, which is the correct behavior. Thus, the inter-element jumps can be used as an indicator in both smooth and non-smooth regions of the solution.

In order to obtain directional information, the average jump K_i of a function ϕ over the two opposite faces f_i^j, $j=1,2$, perpendicular to one coordinate direction i on the reference element can be evaluated as

$$K_i = \frac{\sum_j |\int_{f_i^j} \lfloor\phi\rfloor \, \mathrm{d}x|}{\sum_j \mathrm{meas}(f_i^j)}, \quad i=1,2 \tag{16}$$

where $\lfloor\phi\rfloor = \phi^+ - \phi^-$ denotes the jump of a scalar function ϕ, the summations run over $j=1,2$, and $\int \cdot \, \mathrm{d}x$ indicates a line integral in two dimensions. Equation (16) provides two distinct values for each element. If there is a k, $1\leqslant k\leqslant 2$, such that the average jump in the direction k is larger than the other one in direction l by a certain threshold factor θ,

$$K_k > \theta\, K_l, \quad l = 3-k \tag{17}$$

then the element can be marked for anisotropic refinement in the corresponding direction k. If the solution function is vector valued, as is the case for the flow equations, the jump of a scalar function ϕ in Equation (16) has to be replaced by an appropriate norm of the vector of jumps, for example, the L^2-norm.

The empirical threshold factor $\theta > 1$ has to be chosen large enough to ensure that only those elements are flagged for anisotropic refinement, which are located near strong anisotropic features, otherwise the error would not be reduced sufficiently. On the other hand, however, a smaller value of θ allows more elements to be treated anisotropically, thereby leading to a reduced number of total elements. Numerical experiments showed that $\theta = 3.0$ is a good choice for a range of test problems.

An important advantage of the jump indicator is its direct applicability to higher-order DG discretizations as well as to h-, p- and hp-refinement with relatively little implementational effort.

6.2. Derivative indicators

Projection error estimates: An important contribution to the error can be approximated by the projection error. Given a projection operator Π_p, for example, the L^2-projector, projecting the solution to a space of piecewise polynomials of degree p and assuming that the solution $u|_\kappa \in H^{p+1}(\kappa)$ the projection error on rectangular, axis-parallel elements with edge length h_1 and h_2 along the x_1 and x_2 axes, respectively, can be bounded as follows:

$$\|u - \Pi_p u\|_{L_2(\kappa)} \leqslant C \sum_{i=1}^{2} h_i^{p+1} \|\partial_i^{p+1} u\|_{L_2(\kappa)} \tag{18}$$

with a problem-dependent constant C, that is independent of the element size h [18]. In order to reduce the projection error, the mesh size has to be small along the direction, in which the derivative is large, whereas it can be moderately high in directions with smaller derivatives.

For general quadrilaterals (18) has to be replaced by an expression containing the full derivative tensor: Given a $u|_\kappa \in H^{s+1}(\kappa)$, then

$$\|u - \Pi_p u\|_{L_2(\kappa)} \leqslant C \left[\int_\kappa \|\hat{D}^k u(\mathbf{x})\|_F^2 \, \mathrm{d}\mathbf{x} \right]^{1/2}, \quad 0 \leqslant k \leqslant \min(p,s)+1 \tag{19}$$

see Houston *et al.* [14], where $\hat{D}^k u$ is the kth-order derivative tensor transformed to the reference element as described below, C is a constant which depends only on the dimension and the polynomial degree and $\|\cdot\|_\mathrm{F}$ denotes the Frobenius norm, which for a tensor $A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined by

$$\|A\|_\mathrm{F} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} (A_{i_1 i_2 \ldots i_N})^2} \tag{20}$$

In the following it will be assumed that u is sufficiently smooth, i.e. that $u \in H^{p+1}(\Omega)$.

If a tensor has a dominant component, its Frobenius norm can be efficiently reduced if only this component is reduced in size. The transformation of the derivative tensor from real-space coordinates to the reference element reduces the derivative component(s) in the direction, which has a small mesh size, see below. If the (transformed) derivative is exceptionally large in one direction, reducing the mesh size in that direction offers an effective way to reduce the projection error, while reductions in the other directions are only of minor effect.

This offers a possibility of evaluating an element's aspect ratio. If the transformed derivative has components of similar magnitude in all directions, the element's aspect ratio is already optimal, further refinement should take place isotropically. If one component is dominant, however, the aspect ratio is not yet optimal and anisotropic refinement should be used in order to improve the aspect ratio.

Finite difference approximation of derivatives: For a one-dimensional polynomial of degree p the $(p+1)$th derivative vanishes. In case of a tensor product polynomial of degree p this is true for the main diagonal components of the derivative tensor. Therefore, the derivative tensor cannot be calculated locally from the piecewise polynomial approximation, but has to be approximated using additional information from the surrounding elements. On each element the pth-order derivative can be approximated using the piecewise polynomial. Setting up a finite difference scheme on an element and its neighbors, the $(p+1)$th-order derivative can be approximated. This results in a constant approximation for the derivative tensor at the element's midpoint, which is assumed to be representative for the whole element.

In the following, an algorithm is given of how the gradient is computed on an element κ based on finite difference approximations, see also `deal.II` [19]. The approximation of higher-order derivatives follows the same principle. If κ' is a neighboring element and $\mathbf{y}_{\kappa'} = \mathbf{x}_{\kappa'} - \mathbf{x}_\kappa$ describes the distance vector between the centers of the two elements, then $\phi_h(\mathbf{x}_{\kappa'}) - \phi_h(\mathbf{x}_\kappa)/\|\mathbf{y}_{\kappa'}\|$ is an approximation of the directional derivative $\nabla\phi(\mathbf{x}_\kappa) \cdot \mathbf{y}_{\kappa'}/\|\mathbf{y}_{\kappa'}\|$ of the scalar-valued function ϕ. Multiplying both terms by $\mathbf{y}_{\kappa'}/\|\mathbf{y}_{\kappa'}\|$ from the left and summing over all neighbors κ', the approximation

$$\sum_{\kappa'} \left(\frac{\mathbf{y}_{\kappa'}}{\|\mathbf{y}_{\kappa'}\|} \otimes \frac{\mathbf{y}_{\kappa'}}{\|\mathbf{y}_{\kappa'}\|} \right) \nabla\phi(\mathbf{x}_\kappa) \approx \sum_{\kappa'} \left(\frac{\mathbf{y}_{\kappa'}}{\|\mathbf{y}_{\kappa'}\|} \frac{\phi_h(\mathbf{x}_{\kappa'}) - \phi_h(\mathbf{x}_\kappa)}{\|\mathbf{y}_{\kappa'}\|} \right) \tag{21}$$

is obtained, where $\otimes$ denotes the outer product of two vectors as introduced in Section 3. If the matrix $Y = \sum_{\kappa'} (\mathbf{y}_{\kappa'}/\|\mathbf{y}_{\kappa'}\| \otimes \mathbf{y}_{\kappa'}/\|\mathbf{y}_{\kappa'}\|)$ is regular, which is the case when the vectors $\mathbf{y}_{\kappa'}$ to all neighbors span the whole space, the gradient can be approximated by

$$\nabla\phi(\mathbf{x}_\kappa) \approx Y^{-1} \sum_{\kappa'} \left(\frac{\mathbf{y}_{\kappa'}}{\|\mathbf{y}_{\kappa'}\|} \frac{\phi_h(\mathbf{x}_{\kappa'}) - \phi_h(\mathbf{x}_\kappa)}{\|\mathbf{y}_{\kappa'}\|} \right) \tag{22}$$

This way, the full gradient can be approximated using a summation of approximations to projected derivatives in several directions. Those contributions are easily computable, local quantities. Approximations to higher derivatives can be computed in an analogous way. For example, the tensor of second derivatives is approximated by the formula

$$\nabla^2\phi(\mathbf{x}_\kappa) \approx Y^{-1}\sum_{\kappa'}\left(\frac{\mathbf{y}_{\kappa'}}{\|\mathbf{y}_{\kappa'}\|}\otimes\frac{\nabla\phi_h(\mathbf{x}_{\kappa'})-\nabla\phi_h(\mathbf{x}_\kappa)}{\|\mathbf{y}_{\kappa'}\|}\right) \tag{23}$$

where the local function values $\phi_h(\mathbf{x}_{\kappa'})$ and $\phi_h(\mathbf{x}_\kappa)$ have been replaced by local values of the gradient $\nabla\phi_h(\mathbf{x}_{\kappa'})$ and $\nabla\phi_h(\mathbf{x}_\kappa)$, respectively.

It has to be noted that unlike the true tensor of second derivatives, its approximation is not necessarily symmetric. This is due to the fact that in the derivation, it is not clear whether the term $\nabla^2\phi\mathbf{y}_{\kappa'}$ or $\mathbf{y}_{\kappa'}^{\mathrm{T}}\nabla^2\phi$ should be considered as a projected second derivative. Depending on this choice, one approximation of the tensor of second derivatives or its transpose is obtained. To avoid this ambiguity, the approximation can be symmetrized.

Transformation of the derivative tensor to the reference element: Having computed the tensor of $(p+1)$th derivatives of the discrete solution on κ, $\kappa\in\mathscr{T}_h$, as described in the previous paragraph, the derivative tensor must be transformed onto the reference element $\hat{\kappa}$.

Given a function $\phi:\kappa\to\mathbb{R}$, the corresponding function $\hat{\phi}:\hat{\kappa}\to\mathbb{R}$ on the reference element is given by $\hat{\phi}(\hat{\mathbf{x}})=\phi(\mathbf{x})=\phi(\sigma_\kappa(\hat{\mathbf{x}}))$ where $\sigma:=\sigma_\kappa:\hat{\kappa}\to\kappa$ is mapping between reference element and element in real space as defined in Section 3. Using the notations $\hat{\partial}_k:=\partial/\partial\hat{x}_k$ and $\partial_k:=\partial/\partial x_k$ for the partial derivatives with respect to kth coordinate direction in reference coordinates on $\hat{\kappa}$ and in coordinates on κ, we obtain $\hat{\partial}_j\hat{\phi}=\partial_i\phi\, J_{ij}$, by using the chain rule, where $J_{ij}(\hat{\mathbf{x}})=\hat{\partial}_j\sigma_i(\hat{\mathbf{x}})$, $1\leqslant i,\, j\leqslant 2$, denote the components of the Jacobian matrix J of the mapping function σ. The higher-order derivatives of $\hat{\phi}$ are obtained by successive application of the chain rule. Already for the second-order derivative, this includes derivatives of the Jacobian matrix. In fact, the second-order derivative is given by

$$\hat{\partial}_k\hat{\partial}_l\hat{\phi}=\partial_i\partial_j\phi\hat{\partial}_k\sigma_i\hat{\partial}_l\sigma_j+\partial_i\phi\hat{\partial}_k\hat{\partial}_l\sigma_i=\partial_i\partial_j\phi J_{ik}J_{jl}+\partial_i\phi\hat{\partial}_l J_{ik} \tag{24}$$

We assumed that the mapping σ_κ can be decomposed into an affine part $\bar{\sigma}_\kappa$ and a higher-order part $\tilde{\sigma}_\kappa$, such that $\kappa=\sigma_\kappa(\hat{\kappa})=\tilde{\sigma}_\kappa(\bar{\sigma}_\kappa(\hat{\kappa}))$. Assuming that the higher-order part performs only small variations like bending, but no change in the dimension and the rotation of the element, the terms containing derivatives of the Jacobian matrix, i.e. second and higher derivatives of the mapping, can be neglected, leading to a simple transformation formula connecting the second derivative on the reference element and on the element in real space,

$$\hat{\partial}_k\hat{\partial}_l\hat{\phi}=\partial_i\partial_j\phi J_{ik}J_{jl} \tag{25}$$

Higher-order derivatives can be obtained using the same scheme. Using only the affine part of the mapping, the transformation follows the same structure for higher-order derivatives: each index of the derivative tensor has to be contracted with the first index of the Jacobian matrix of the mapping, for example, the transformation of the third-order derivative is given by

$$\hat{\partial}_l\hat{\partial}_m\hat{\partial}_n\hat{\phi}=\partial_i\partial_j\partial_k\phi J_{il}\, J_{jm}J_{kn} \tag{26}$$

Evaluation for anisotropic indicators: Having transformed the approximated derivative at the center of an element κ to the reference element $\hat{\kappa}$ it remains to evaluate this tensor in order to decide whether an anisotropic refinement should take place and if so which refinement case, see Section 5, shall be used.

For a second-order derivative, i.e. the Hessian matrix, anisotropic information can be easily extracted by computing the eigenvalues λ_i and the corresponding eigenvectors $\mathbf{v}_i$, $1\leqslant i\leqslant 2$. The ratio of the absolute values of the eigenvalues,

$$r=\frac{\max_{i=1,2}|\lambda_i|}{\min_{i=1,2}|\lambda_i|} \tag{27}$$

here in two dimensions, can be used as a measure of the magnitude of anisotropic properties of the solution. A value near unity corresponds to an isotropic behavior, whereas with increasing r the anisotropic features get stronger. Anisotropic refinement should only be considered for an element, if the anisotropic ratio r is larger than an empirical threshold value θ_r.

The second important information gained from the calculation of eigenvectors is the direction in which the derivative takes its maximum absolute value. If this direction is sufficiently aligned with one of the coordinate axes on the reference element, anisotropic refinement along this axis might be effective. In order to decide upon the refinement, the angle α_i between the eigenvector and the coordinate axis i is compared with a second threshold value θ_α. If the two conditions

$$r>\theta_r \quad \text{and} \quad |\alpha_i|<\theta_\alpha \tag{28}$$

are simultaneously fulfilled the element is flagged for anisotropic refinement along direction i.

For higher-order derivatives the calculation of eigenvectors and eigenvalues is not possible and the evaluation of directions in which the projected derivative takes on extremal values is expensive. Concerning the refinement, there are very limited options, namely the refinement can be done along the coordinate axes only. Thus, the exact knowledge of the characteristic direction is not necessary. It should suffice to calculate the ratio of the projected derivatives in the coordinate directions on the reference element; a similar approach has been used in [20]. As these projected derivatives are the terms on the main diagonal of the tensor, this is a trivial evaluation. This leads to a simple criterion, which is based on only one empirical threshold factor $\tilde{\theta}_r$. If there is a k, $1\leqslant k\leqslant 2$, such that the condition

$$|\hat{\partial}_k^{p+1}\hat{\phi}|>\tilde{\theta}_r|\hat{\partial}_l^{p+1}\hat{\phi}|, \quad l=3-k \tag{29}$$

is satisfied, the refinement should take place anisotropically along the $\hat{x}_k$-axis.

Using only the terms on the main diagonal of the derivative tensor seems quite natural and is a very cheap operation. It has to be kept in mind that anisotropic refinement will only be a useful addition to adaptive algorithms, if the evaluation of anisotropic indicators is inexpensive.

In order to prevent anisotropic refinement in inappropriate situations, the threshold ratios θ_r and $\tilde{\theta}_r$ have to be chosen large enough, whereas θ_α has to be chosen small enough. On the other hand, too restrictive values reduce the possible saving effect. The values 3.0 for the threshold ratios and 15° for the threshold angle seem to be adequate for several test cases.

Application to vector-valued solution functions: So far only scalar-valued solution functions ϕ have been considered. Fluid flow calculations, however, include vector-valued solution functions $\mathbf{u}$ that are composed, for example, of the density, the momentum components and the internal energy. Three different treatments of multi-component functions have been investigated:

(a) The information extracted from the derivatives of the individual vector components is 'blended' before it is evaluated. In the case of second-order derivatives, a procedure called *metric intersection* is applied to the Hessian matrices, which have to be appropriately scaled in order to represent relative changes instead of absolute ones. Concerning metric intersection, see the detailed discussion below.
(b) The refinement indicator is evaluated separately for each component. If a sufficient number of the individual indicators suggest the same refinement strategy, and the other indicators do not contradict this suggestion (i.e. the other indicators suggest isotropic refinement), this strategy is chosen, otherwise the element is refined isotropically.
(c) Instead of several independent components, one key variable is used to determine the properties. In order to deliver good estimates of the anisotropic features, the key variable should be characteristic of the whole flow field. Therefore, derived variables like the velocity scalar v or the Mach number M are possible choices. This approach avoids an empirical combination of several indicators or derivative tensors.

Metric intersection: For Hessian matrices the desired 'blending' of derivatives can be obtained by a procedure called *metric intersection*. Each metric $\mathscr{M}_i$ is identified with an ellipse *via* $\mathbf{x}^{\mathrm{T}} \mathscr{M}_i \mathbf{x} = 1$, $\mathbf{x} \in \mathbb{R}^2$. The intersection is defined as the largest ellipse lying completely inside the intersection area of the ellipses associated with $\mathscr{M}_i$. The intersection of several metrics can be approximated by the repeated intersection of two metrics, as proposed in [21]. The approximate intersection procedure in [21] gives quite poor results in some cases, however. This drawback can be avoided if the actual intersection of two metrics is determined using the procedure given in [22], which is based on the simultaneous reduction of the quadratic forms associated with the two metrics and is also used in [8, 9].

In order to regard different magnitudes and dimensions of the components of the solution vector, the Hessians are scaled with local values of the solution components, resulting in scaled derivatives that represent the relative instead of the absolute error.

7. NUMERICAL RESULTS

In the following, we test the practical performance and the effectiveness of the different anisotropic indicators for several sub-, trans- and supersonic, inviscid and viscous compressible flows. The test cases are solved using the DG(1) method which is second-order accurate. In order to demonstrate the applicability of the developed indicators to higher-order discretizations as well as the influence of the order of the numerical scheme on the performance of anisotropic mesh refinement, we also show some computations based on the DG(2) method which is third-order accurate.

The discrete equations are solved using a Newton algorithm for the nonlinear problem with a restarted version of the GMRES algorithm for the linear subproblems. GMRES with an ILU preconditioner is also used for the solution of the linear adjoint problems. The trans- and supersonic test cases use a shock capturing based on a consistent artificial viscosity term [5].

All test cases are based on the NACA0012 airfoil, which is defined by

$$y(x) = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \tag{30}$$

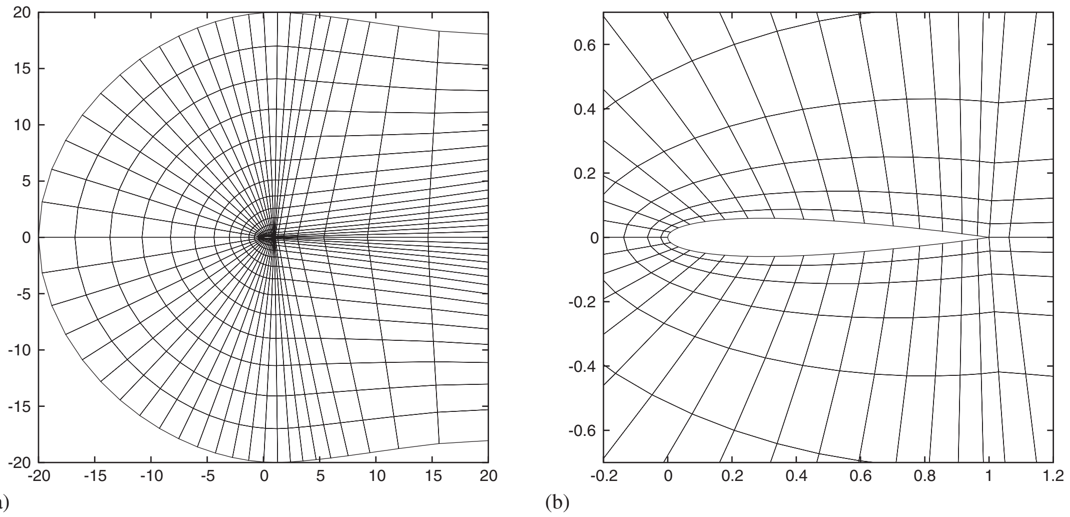

Figure 5. Initial mesh of 768 elements: (a) complete domain and (b) vicinity of the NACA0012 airfoil.

As this equation yields an airfoil with slightly more than unit length, a rescaling is used to ensure that $y(0)=y(1)=0$. The initial mesh for the adaptive process is a C-type mesh containing 768 elements. This mesh as well as a zoom of the mesh in the vicinity of the airfoil is shown in Figure 5. For some test cases, a globally refined version of this mesh is used, which contains 3072 elements.

Test case A: *viscous subsonic flow*. A subsonic viscous laminar flow around a NACA0012 airfoil at Mach number $M=0.5$, Reynolds number $Re=5000$ and zero angle of attack is computed based on the compressible Navier–Stokes equations. As the airfoil is symmetric, the lift coefficients vanish for zero angle of attack. Figure 6 shows the Mach isolines of the resulting symmetric flow. The target functional under consideration is the pressure-induced drag coefficient c_{dp}, which is evaluated to approximately 0.0222875 by fine grid computations.

Test case B: *inviscid transonic flow*. The transonic inviscid flow around a NACA0012 airfoil at Mach number $M=0.8$ and an angle of attack of $\alpha=1.25^\circ$ is computed based on the compressible Euler equations. Figure 7 shows the Mach isolines of this flow. The subsonic flow is accelerated to supersonic velocities near the surface of the airfoil. On the second half of the upper airfoil surface the velocity is reduced spontaneously by a shock, influencing both drag and lift. This shock on the upper side of the airfoil as well as the smaller one at the lower side can clearly be seen by the agglomeration of Mach isolines. The target functional under consideration is the pressure-induced drag coefficient c_{dp}, which is evaluated to approximately 0.0219 by fine grid computations.

Test case C: *viscous supersonic flow*. The supersonic viscous laminar flow around a NACA0012 airfoil at Mach number $M=1.2$, Reynolds number $Re=1000$ and zero angle of attack is computed based on the compressible Navier–Stokes equations. A detached bow shock in front of the airfoil decelerates the flow to subsonic velocity in the vicinity of the airfoil. However, the main flow accelerates to supersonic velocities again, whereas the flow near the airfoil and in the wake stays subsonic. Figure 8 shows the Mach isolines for this test case. The target functional under

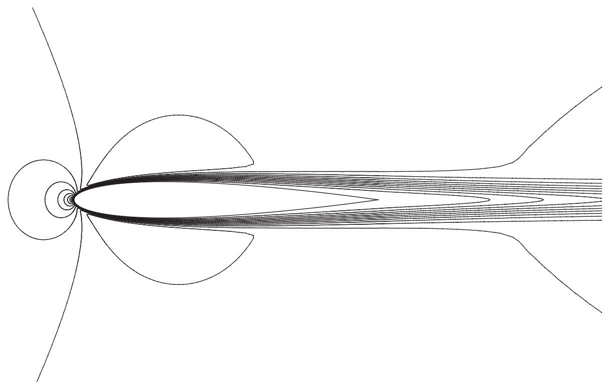

Figure 6. Test case A: viscous flow at $M=0.5$, $\alpha=0^{\circ}$ and $Re=5000$: Mach isolines.

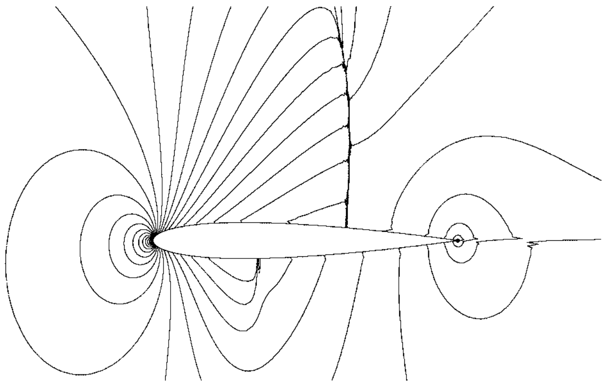

Figure 7. Test case B: inviscid flow at $M=0.8$ and $\alpha=1.25^{\circ}$: Mach isolines.

consideration is the viscous drag coefficient c_{df}, which is evaluated to approximately 0.1079 by fine grid computations.

Several parameters and threshold values have been introduced but kept unspecified in the definition of the anisotropic indicators in Sections 6.1 and 6.2. In the following two Sections 7.1 and 7.2 we give some numerical results that helped to find appropriate values.

7.1. Specification of the jump indicator

The evaluation of the jump indicator includes an empirical threshold value θ, see (17), which describes the desired ratio of the largest average jump in one direction and the average jump in the remaining direction. Several examples that we omit for brevity show that a value of about $\theta \approx 3.0$ gives the best results. Higher values result in too few anisotropic elements, whereas smaller values lead to anisotropic refinement in inappropriate places, resulting in a reduced accuracy.

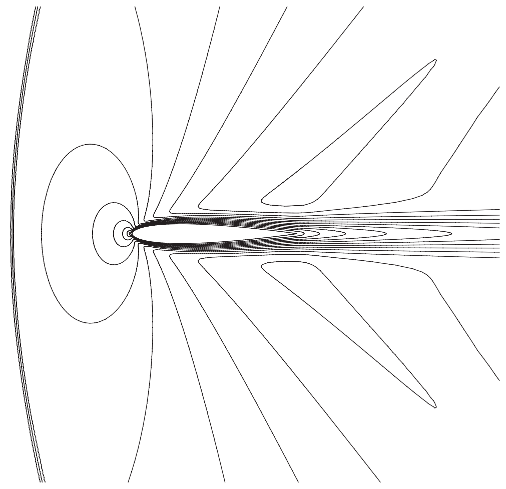
Figure 8. Test case C: viscous flow at $M = 1.2$, $\alpha = 0°$ and $Re = 1000$: Mach isolines.

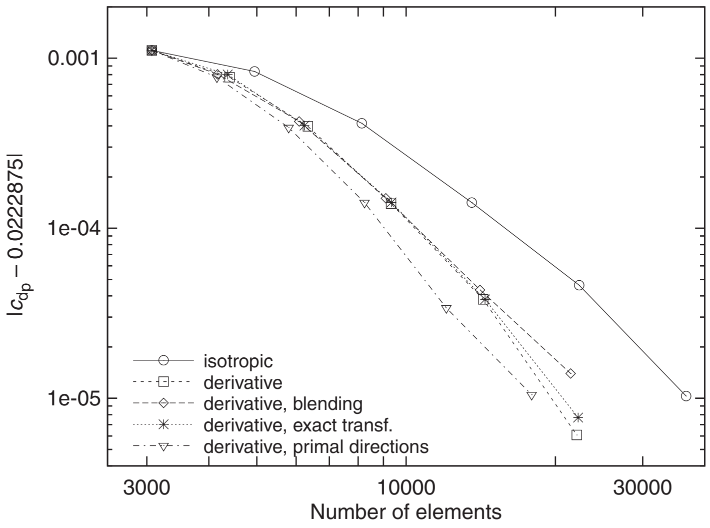

Figure 9. Test case A: convergence of the error in c_{dp} for the DG(1) method using adjoint-based error indicators combined with different anisotropic derivative indicators.

7.2. Specification of the derivative indicators

The evaluation of the derivative indicators described in Section 6.2 offers a wide range of alternatives. The main results for the comparison of different derivative indicators will be given for test case A using the DG(1) method and the adjoint-based error indicators.

In Figure 9, we collect the convergence history of the error in the target functional based on different derivative indicators. The reference case is based on evaluating the characteristic direction of the derivative (in this case, the eigenvector belonging to the largest absolute eigenvalue of the Hessian matrix) and comparing it to a threshold angle $\theta_\alpha = 15°$, see (28), as well as comparing the anisotropic ratio as defined in (27) to a second threshold value $\theta_r = 3.0$. This is done individually for all conservative variables. If at least two of these evaluations suggest the same anisotropic refinement and the others do not suggest a different direction, i.e. they suggest isotropic refinement, then the element is flagged for anisotropic refinement in this particular direction. The resulting convergence curve clearly shows that the accuracy reached after a given number of refinement steps is very similar to that of the corresponding isotropic refinement, but uses a reduced number of elements, in the example this reduction is about 40% after the last refinement step.

As described above, the derivative indicator uses a linearized transformation from real space coordinates to the reference element. The assumption, that higher-order terms including derivatives of the Jacobian matrix of the transformation can be neglected seems justified. Using the exact transformation given by (24), the overall convergence changes only slightly, as shown in Figure 9. This justifies the use of the linearized transformation, which enables a simple and unified approach for all derivative orders.

The concept of blending is introduced in Section 6.2 as a possible means of dealing with vector-valued solution functions. The general idea is to use a combination of the individual derivative components which is dominated by the strongest anisotropic features and thus allows to refine an element, if at least one variable shows strong anisotropic features whereas the others do not. As Figure 9 indicates, the expected gain concerning the number of elements can be observed in the numerical examples, the blended derivatives lead to an almost identical convergence behavior compared with the reference case above, only the value on the finest mesh shows a deviation from this trend. In the end, the similar results justify the use of the presented procedure based on individual indicator evaluation, which is, in contrast to metric intersection, extensible to higher-order discretizations.

The last indicator used in Figure 9 is based on evaluating the derivative only in the direction of the coordinate axes on the reference element (primal directions) and comparing the resulting ratio with a threshold value $\tilde{\theta}_r$, see (29). No threshold angle θ_α is used. Using both threshold values θ_r and θ_α from above, a corresponding threshold value can be calculated for the primal directions, if the derivative described by these threshold values is projected to the coordinate axes of the reference element. For a second-order derivative, this results in

$$\tilde{\theta}_r = \frac{\theta_r \cos^2(\theta_\alpha) + \sin^2(\theta_\alpha)}{\cos^2(\theta_\alpha) + \theta_r \sin^2(\theta_\alpha)} \tag{31}$$

However, this value is not unique to the combination of threshold ratio and threshold angle. Using $\theta_r = 3.0$ and $\theta_\alpha = 15°$ as above, $\tilde{\theta}_r \approx 2.53$ is obtained. In view of (31) this value is obtained also for a lower threshold ratio and lower threshold angles and for a higher threshold ratio and higher threshold angles. This effect leads to a number of elements being flagged for anisotropic refinement, which were not flagged using the more involved indicator based on both threshold ratio and angle. Obviously, this results in a slightly improved convergence behavior. However, on the finest mesh the achieved accuracy is slightly worse than for all other methods. Still, the differences are small and thus justify the use of this simplified criterion, which is directly applicable to higher-order derivatives as well.

Using different threshold values the results vary, as can be expected. Increasing the threshold angle θ_α from 15° to 20° results in a steeper descent of the convergence curve after the first refinement steps. However, in the later refinement steps the error changes the sign, afterwards the absolute value of the error starts growing again. This behavior indicates that the refinement is not really appropriate for the problem, i.e. the threshold values are not strict enough. Similar results are obtained using a key variable like the Mach number M. In early refinement steps, the error is reduced more quickly than with the indicators using all conservative variables, but later on the convergence behavior gets worse, the error changes its sign and grows in absolute values. Again, this indicates that the chosen indicator does not catch all aspects of the physical behavior of the flow under consideration, i.e. the flow field and its anisotropic features cannot be represented with sufficient accuracy by a single key variable.

For each of the test cases defined above, we now compare the convergence of target functional values between isotropic refinement and anisotropic refinement governed by both the specified jump and derivative indicators. Furthermore, we present some of the adaptively refined meshes and some of the results based on higher-order DG discretizations.

7.3. Results for test case A

Figure 10 shows the convergence behavior for test case A, using the DG(1) method, where a jump indicator with $\theta = 3.0$ and a derivative indicator based on the conservative variables with $\theta_r = 3.0$ and $\theta_\alpha = 15°$ is used.

Although the accuracy reached after a given number of refinement steps is very similar for both versions of anisotropic refinement, the number of elements is smaller for the derivative indicator, i.e. the jump indicator reduces the use of resources less effectively. However, both versions demonstrate an improved behavior as compared with isotropic refinement. As the time needed for the evaluation of the indicators can be neglected in comparison with the time spent on the solution process, the saving of elements directly translates to a gain in computational time. Furthermore, a smaller number of elements also reduces the memory consumption.

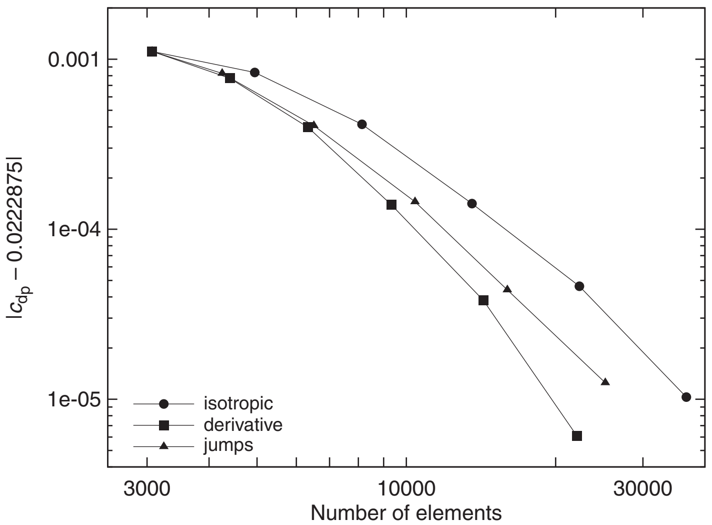

Figure 10. Test case A: convergence of the error in c_{dp} for the DG(1) method using adjoint-based error indicators combined with anisotropic jump and derivative indicators.

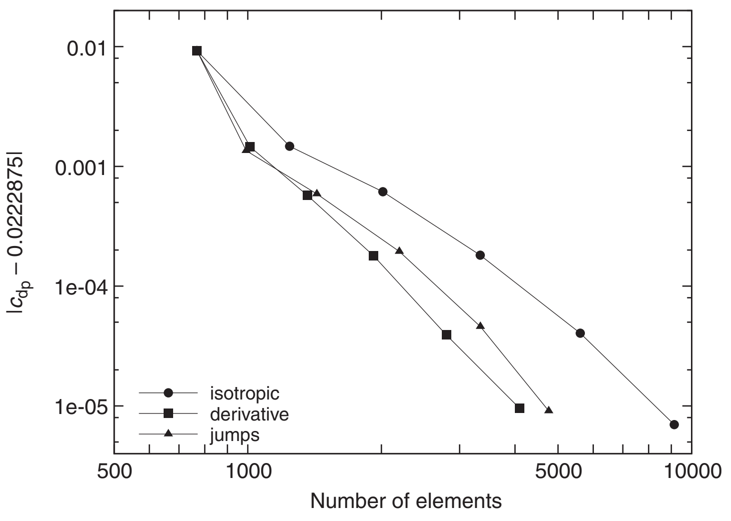

Figure 11. Test case A: convergence of the error in c_{dp} for the DG(2) method using adjoint-based error indicators combined with anisotropic jump and derivative indicators.

Figure 11 shows the corresponding comparison for the DG(2) method. Here, the jump indicator is evaluated using $\theta = 3.0$, whereas $\tilde{\theta}_r = 3.0$ is used for the derivative indicator. Both indicators perform similarly well, the convergence of the derivative indicator is again slightly better than that of the jump indicator. Compared with the DG(1) method, a similar or even larger fraction of elements can be saved on the finest mesh.

Examples of the adapted meshes in the vicinity of the airfoil after three adjoint-based refinement steps for the DG(2) method are shown in Figure 12. The comparison demonstrates the good resolution of the boundary layer in all cases. Near the leading edge, isotropic refinement is used even if anisotropic refinement is enabled. Owing to the strong curvature of the airfoil, the flow changes rapidly along the airfoil as well as in the orthogonal direction. However, further downstream the curvature reduces and anisotropic refinement is appropriate. In this region isotropic refinement produces an unnecessary amount of elements. The comparison between the adapted meshes based on the jump and the derivative indicator shows that the resulting refinement strategy is quite similar in both cases.

7.4. Results for test case B

As inviscid flows do not exhibit strong boundary layer effects, the occurrence of anisotropic features is mainly limited to shocks. The solution for the transonic test case B has a large shock on the upper side of the airfoil and a smaller one at the lower side. It can be expected that anisotropic refinement is advantageous in the region of these shocks. The most restrictive assumption in using the derivative indicators introduced in Section 6.2 is the smoothness of the solution u. For using the $(p+1)$th-order derivative it has to be assumed, that $u \in H^{s+1}(\Omega)$, with $s \geqslant p$. While this is a reasonable assumption in boundary layers, it will generally not be fulfilled in other regions of the flow, in particular at shocks.

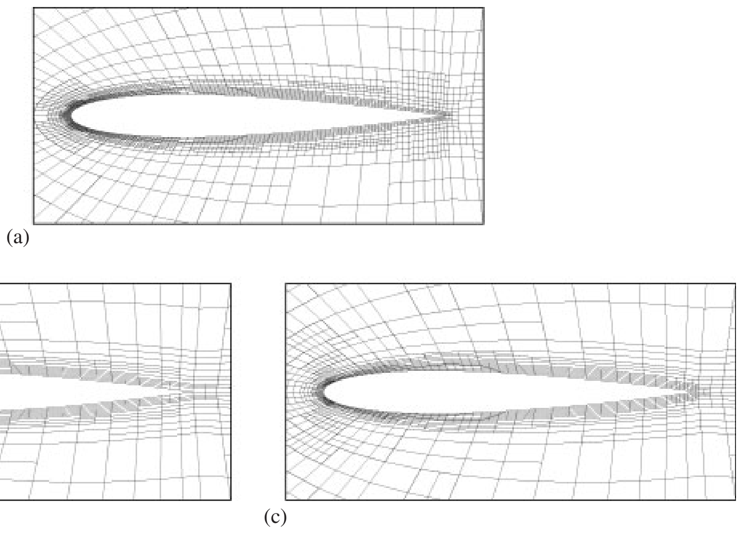

Figure 12. Test case A: adapted mesh after three refinement steps using adjoint-based error indicators for the DG(2) method: (a) isotropic refinement; (b) anisotropic refinement based on derivative indicators; and (c) anisotropic refinement based on jump indicators.

As the derivative indicator is based on this smoothness assumption, its performance is likely to deteriorate. In fact, the derivative indicator based on the conservative variables used for test case A does not lead to any improvement of the convergence behavior. However, using the Mach number M as key variable, anisotropic refinement governed by a derivative indicator can be used effectively. Although the jump indicator does not suffer from unsatisfied assumptions, anisotropic refinement is still restricted to those parts of the domain, where the anisotropic features are sufficiently aligned with the mesh.

Finally, we note that concerning a possible extension to *hp*-refinement it can be assumed that the polynomial degree p will be small in the vicinity of shocks. Thus the applicability of derivative indicators will not be as problematic as it might seem, even near shocks.

Figure 13 shows a comparison of the convergence behavior of the DG(1) method for isotropic refinement, for the derivative indicator based on the Mach number and for the jump indicator. Given the drawbacks of the applicability to inviscid and transonic flows, both anisotropic indicators perform quite well and lead to a reduction of about 40% in the number of elements on the finest mesh in case of adjoint-based error indicators. Although this reduction is higher than expected, it has to be kept in mind that the number of refinement steps is also higher for this test case as compared with test case A. The accuracy reached after a given refinement step is very similar for all presented methods and the jump indicator performs slightly better. The only exception to these two observations is the finest mesh on which the jump indicator leads to a reduced accuracy.

Concerning residual-based error indicators the convergence of the target functional error is slightly worse, as might be expected from the results of test case A. For test case B, anisotropic refinement is not as effective for residual-based indicators as for adjoint-based indicators, although an improvement can still be achieved. Again, the jump indicator performs slightly better than the derivative indicator.

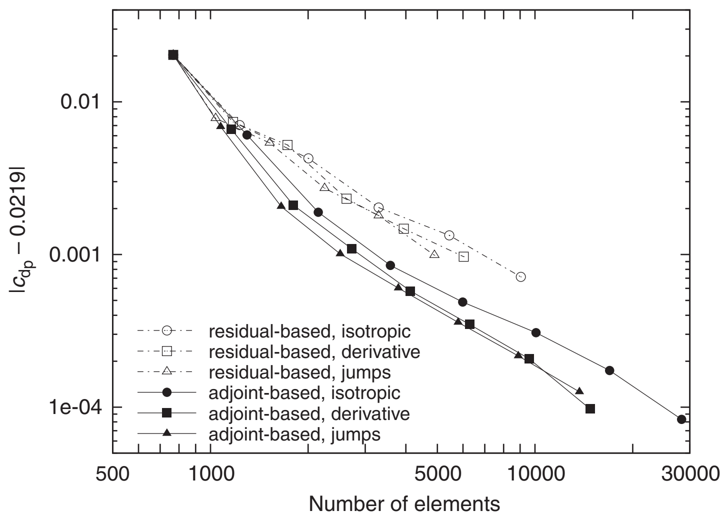

Figure 13. Test case B: convergence of the error in c_{dp} for the DG(1) method using residual and adjoint-based error indicators combined with anisotropic jump and derivative indicators.

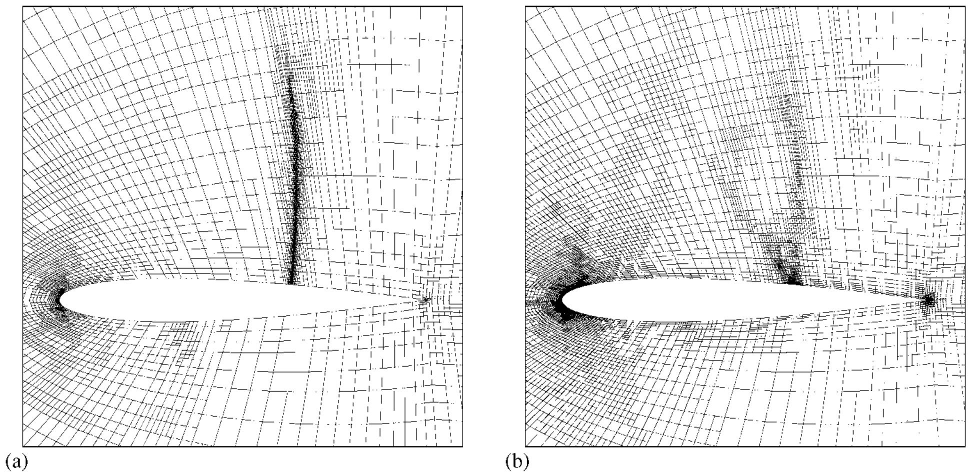

Figure 14. Test case B: adapted anisotropic meshes after six refinement steps for the DG(1) method using a derivative indicator based on the Mach number: (a) residual-based error indicators and (b) adjoint-based error indicators.

Examples of adapted meshes are given in Figure 14, which shows the anisotropically refined meshes obtained after six adaptive refinement steps using the derivative indicator. We observe a clear difference between the meshes obtained with residual-based and obtained with adjoint-based error indicators. Although the first strategy refines mainly the shock on the upper airfoil surface and the leading edge, the latter reduces the refinement of the shock in favor of a stronger refinement of the leading and trailing edge and an additional refinement along a line starting from the point, where the shock touches the airfoil. This line represents a characteristic of the solution to the adjoint problem. This refinement underlines the importance of the exact position of the shock. Clearly, this is more important than a better refinement of the shock, as the mesh obtained with this error indicator yields a target functional error, that is reduced by a factor of approximately 2.5 compared with the corresponding mesh using residual-based error indicators and a comparable number of elements. As the shock is reasonably aligned with the mesh, at least in the upper part, anisotropic elements can be used effectively in both cases.

7.5. Results for test case C

The supersonic viscous flow considered in test case C features both a detached bow shock in front of the airfoil and boundary layers at the surface of the airfoil. Examples of the adapted meshes are given in Figure 15, which shows the meshes after five anisotropic refinement steps with the derivative indicator, both for residual- and adjoint-based error indicators. Like in test case A, an anisotropic derivative indicator based on the conservative variables is used.

Using residual-based error indicators the refinement concentrates on the complete bow shock. As this shock is not well aligned with the initial mesh, anisotropic refinement cannot be used efficiently, although some anisotropic elements appear, mainly induced by mesh smoothing resulting from hanging nodes of the multiply refined elements near the shock. Additional refinement takes place in the wake, where anisotropic refinement is quite effective. However, the boundary layer is hardly refined up to this refinement step. This situation changes drastically, if adjoint-based error indicators are used, which leads to a refinement mainly in the boundary layer. The bow shock and the wake are only refined in the vicinity of the airfoil, where the flow strongly influences the target functional value. Refining the shock and wake far away from the airfoil seems not to be as efficient as refining the boundary layer. As the latter is well aligned with the mesh, anisotropic refinement is quite effective. The aspect ratio of the boundary layer elements might be expected to be higher. However, the Reynolds number is low ($Re = 1000$) which results in quite a weak boundary layer. Furthermore, the aspect ratio of some elements can still increase after additional refinement steps.

Figure 16 shows a comparison for the DG(1) method between isotropic refinement and anisotropic refinement based on the jump indicator as well as on the derivative indicator. Both, residual- and adjoint-based error indicators are used. The advantage of adjoint-based error indicators, which can already be seen from the adapted meshes, is reflected in the convergence plot. For the first few refinement steps, the target functional error does not decrease, if residual-based indicators are used. The refinement of the complete bow shock does not contribute noticeably to an improved target functional value. Therefore, the error does not decrease until the vicinity of the airfoil is refined after some additional steps of the adaptive algorithm. As anisotropic elements are not effective in the bow shock due to little alignment with the initial mesh, no significant improvement can be achieved using anisotropic refinement.

Considering the convergence achieved with adjoint-based error indicators, using an anisotropic refinement strategy a considerable number of elements can be saved on the finest mesh, over 40% of the elements in the case of the derivative indicator and even more than 50% in the case of the jump indicator. Furthermore, the achieved accuracy is very similar for isotropic refinement and anisotropic refinement based on the derivative indicator, whereas the accuracy reached after a given refinement step even increases for the jump indicator.

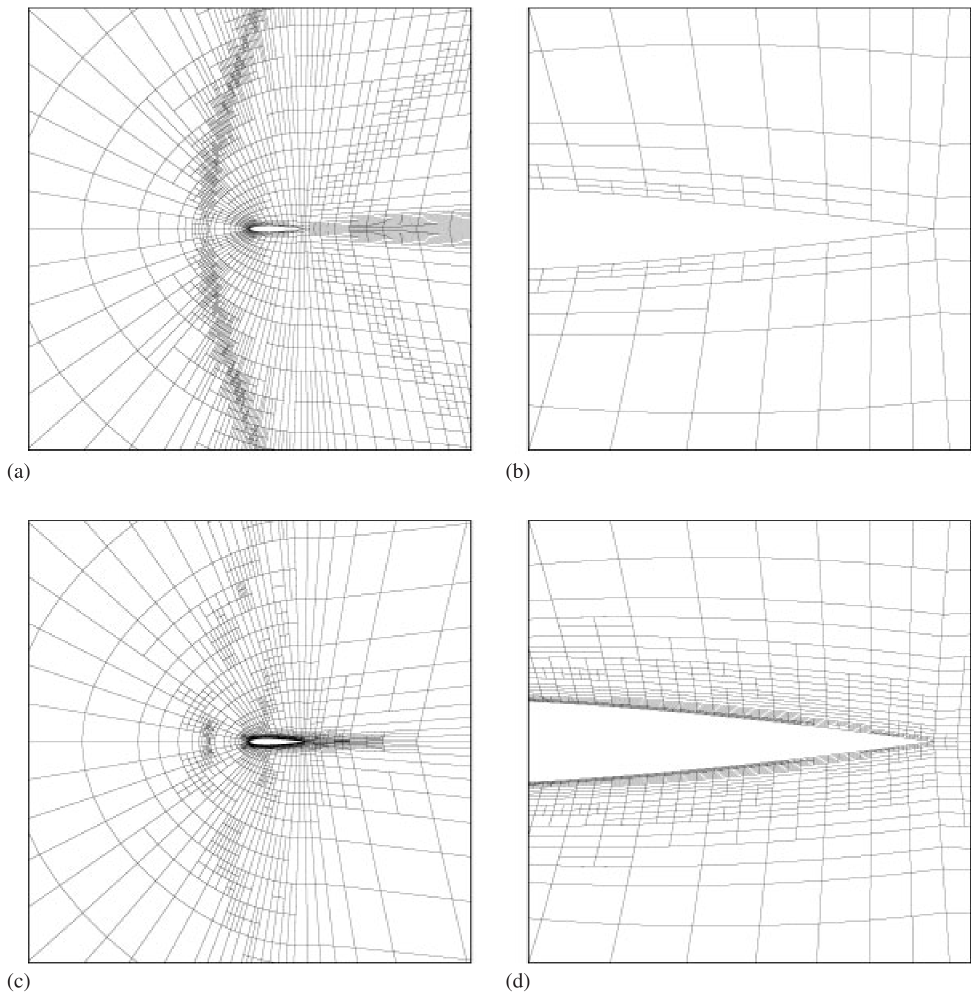

Figure 15. Test case C: adapted anisotropic meshes after five refinement steps for the DG(1) method using a derivative indicator based on the conservative variables: (a) and (b) distant view and zoom for the residual-based indicators and (c) and (d) distant view and zoom for the adjoint-based indicators.

8. CONCLUSION

The numerical examples presented demonstrate that adjoint-based error indicators are clearly advantageous over residual-based indicators. Solutions on adjoint-based adapted meshes had a significantly reduced error in the target quantities as compared with residual-based adapted meshes. This behavior becomes particularly clear in test cases A and C, and justifies the additional effort of solving the adjoint problem. Similar results have been shown in [4–6].

Concerning anisotropic indicators, the results show that both types of the developed indicators can be used successfully to govern an anisotropic refinement process. The reduction in the number

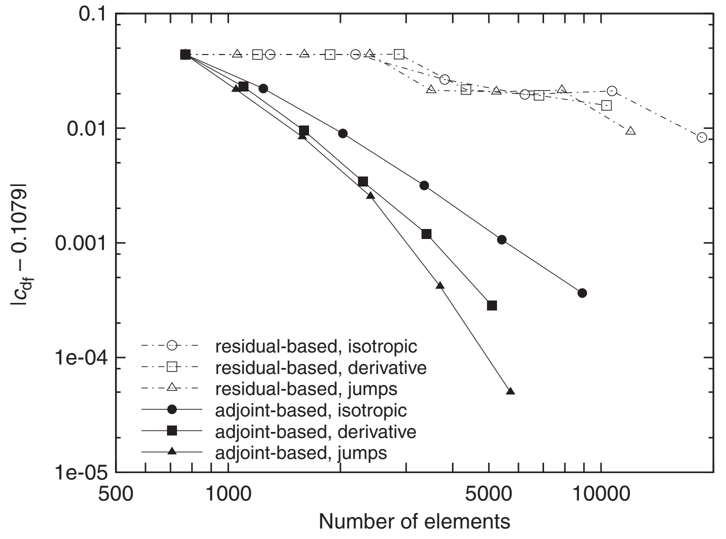

Figure 16. Test case C: convergence of the error in c_{df} for the DG(1) method using residual- and adjoint-based error indicators combined with anisotropic jump and derivative indicators.

of elements achieved is about 40% in most cases. A derivative indicator based on the individual evaluation of the conservative variables gives the best results for a subsonic viscous flow, whereas the jump indicator is advantageous for trans- and supersonic flows. For inviscid flows, anisotropic features can be found mainly in shocks, thus the jump indicator seems advantageous, but a derivative indicator based on the Mach number can be used, too.

Both types of anisotropic indicators can be applied also to higher-order DG discretizations. For the derivative indicator this includes the approximation of higher-order derivatives of the solution components. The evaluation of indicators for each component individually yields comparable results to the usual approach of using a blending technique based on metric intersection and has the advantage of being trivially extensible to higher-order discretizations. Numerical examples show that the simplification of using only the linear part of the mapping for the transformation of derivatives from real space to reference element coordinates is justified.

Anisotropic refinement is most effective in boundary layers. This is caused by the alignment of the initial mesh with the characteristic directions of the boundary layer. Near shocks the mesh is in general not well aligned resulting in an isotropically dominated refinement.

In summary, it has been shown that the developed anisotropic indicators are effective for sub-, trans- and supersonic inviscid and laminar flows. As the indicators come computationally almost for free, the reduction of 40% in the number of elements and degrees of freedom in comparison with isotropic refinement while reaching the same accuracy directly translates into a corresponding reduction of computing resources required. Future research will be devoted to compare the overall performance of the anisotropic indicators developed in this article to the use of locally recomputed primal and adjoint solutions within a fully *a posteriori* error estimation approach as suggested in [14]. Further research will also include the extension of the anisotropic refinement algorithm to the simulation of three-dimensional aerodynamic flows. Apart from the implementational effort this will require the extension of the presented anisotropic indicators, as in 3d a single value will not be sufficient to describe the ratio of anisotropic features. In order to reduce complexity the effort might be concentrated on the simpler jump indicator, which has the additional advantage of being directly applicable to *hp*-refinement situations as well.

ACKNOWLEDGEMENTS

All computations have been performed with the DG flow solver `PADGE` (Parallel Adaptive Discontinuous Galerkin Environment) [23] currently developed at DLR, Braunschweig, which is based on the `C++` Finite Element library `deal.II` [19, 24]. This work has been supported by the President's Initiative and Networking Fund of the Helmholtz Association of German Research Centres.

REFERENCES

1. Le Saint P, Raviart P. On a finite element method for solving the neutron transport equation. In *Mathematical Aspects of Finite Elements in Partial Differential Equations*, de Boor C (ed.). Academic Press: New York, 1974; 89–123.
2. Chanvent G, Salzano G. A finite element method for the 1-D water flooding problem with gravity. *Journal of Computational Physics* 1982; **45**:307–344.
3. Cockburn B, Karniadakis G, Shu C-W. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods*, Cockburn B, Karniadakis G, Shu C-W (eds), vol. 11. Springer: Berlin, 1999; 3–50.
4. Hartmann R, Houston P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics* 2002; **183**:508–532.
5. Hartmann R. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2006; **51**(9–10):1131–1156.
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 Modelling* 2006; **3**(2):141–162.
7. Formaggia L, Micheletti S, Perotto S. Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the Stokes problems. *Applied Numerical Mathematics* 2004; **51**:511–533.
8. Frey PJ, Alauzet F. Anisotropic mesh adaption for transient flow simulations. *Twelfth International Meshing Roundtable*, Sandia National Laboratories, 2003; 335–348.
9. Frey PJ, Alauzet F. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:5068–5082.
10. Huang W. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics* 2005; **204**:633–665.
11. Sahni O, Müller J, Jansen KE, Shepard MS, Taylor CA. Efficient anisotropic adaptive discretization of the cardiovascular system. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5634–5655.
12. Venditti DA, Darmofal DL. Anisotropic grid adaption for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics* 2003; **187**:22–46.
13. Sun S, Wheeler MF. Anisotropic and dynamic mesh adaption for discontinuous Galerkin methods applied to reactive transport. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3382–3405.
14. Houston P, Georgoulis EH, Hall E. Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes. In *International Conference on Boundary and Interior Layers*, *BAIL2006*, Lube G, Rapin G (eds). Göttingen, Germany, 24–28 July 2006.
15. Hartmann R, Houston P. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: method formulation. *International Journal of Numerical Analysis and Modelling* 2006; **3**(1):1–20.
16. Becker R, Rannacher R. An optimal control approach to a-posteriori error estimation in finite element methods. *Acta Numerica* 2001; **10**:1–102.
17. Süli E, Houston P. Adaptive finite element approximation of hyperbolic problems. In *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Barth T, Deconinck H (eds). Lecture Notes in Computational Science and Engineering, vol. 25. Springer: Berlin, 2002; 269–344.
18. Georgoulis EH. Discontinuous Galerkin methods on shape-regular and anisotropic meshes. *Ph.D. Thesis*, University of Oxford, 2003.

19. Bangerth W, Hartmann R, Kanschat G. `deal.II` *Differential Equations Analysis Library*. Technical Reference. http://www.dealii.org/, 5.2 edition, 2005.
20. Aftosmis MJ, Kroll N. A quadrilateral based second-order TVD method for unstructured adaptive methods. *AIAA 91-0124*, 1991.
21. Castro-Díaz MJ, Hecht F, Mohammadi B, Pironneau O. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids* 1997; **25**:475–491.
22. Alauzet F, Frey PJ. Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. Partie I: aspects théoriques. *RR-4759*, INRIA Rocquencourt.
23. Hartmann R *et al*. `PADGE`, *Technical Reference*. DLR Braunschweig, 2007.
24. Bangerth W, Hartmann R, Kanschat G. deal.II—A general purpose object oriented finite element library. *ACM Transactions on Mathematical Software* 2007; **33**(4).
25. Leicht T. Anisotropic mesh refinement for discontinuous Galerkin methods in aerodynamic flow simulations. *Diploma Thesis*, Dresden University of Technology, 2006.