

# MULTITARGET ERROR ESTIMATION AND ADAPTIVITY IN AERODYNAMIC FLOW SIMULATIONS

RALF HARTMANN\*

**Abstract.** Important quantities in aerodynamic flow simulations are the aerodynamic force coefficients including the pressure induced and the viscous stress induced drag, lift and moment coefficients. In addition to the exact approximation of these quantities it is of increasing importance, in particular in the field of uncertainty quantification, to estimate the error in the computed quantities.

In recent years a *posteriori* error estimation and goal-oriented refinement approaches have been developed for the accurate and efficient computation of *single* target quantities. The current approaches are based on computing an adjoint solution related to each of the specific target quantities under consideration. In this paper we extend this approach to the accurate and efficient computation of *multiple* target quantities. Instead of computing multiple adjoint solutions, one for each target functional, the new approach is based on the solution to one discrete adjoint problem and one discrete error problem. This way only two auxiliary problems are required irrespective of the number of target functionals. The practical performance of this approach is demonstrated for a laminar compressible flow. In particular, the proposed approach is compared to the standard approach of error estimation and goal-oriented refinement as well as to residual-based refinement. The performance of the algorithms is measured in terms of computing resources required for meeting industrial as well as academic accuracy requirements on the computed force coefficients.

**Key words.** Discontinuous Galerkin discretization, error estimation, adaptivity, multiple target functionals, compressible Navier-Stokes equations

**AMS subject classifications.** 65N12,65N15,65N30

**1. Introduction.** In aerodynamic computations like compressible flows around airfoils, much emphasis is placed on the accurate approximation of specific target quantities, in particular, the aerodynamic force coefficients like the pressure induced as well as the viscous stress induced drag, lift and moment coefficients, respectively. First, the force coefficients should be computed as accurately and efficiently as possible in order to avoid wasting computing resources. Second, it is of increasing importance to have an idea on how accurate the obtained results are, in particular, to have an estimate of the error in the computed force coefficients. In fact, the discretization error is one of the main sources of uncertainty in flow computations.

While local mesh refinement is required for obtaining reasonably accurate results in applications, the goal of the adaptive refinement is either to compute the force coefficients as accurately as possible within given computing resources or to compute these quantities 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. Furthermore, in the latter case, an estimate is required on how accurate the force coefficients are approximated, i.e. an *a posteriori* error estimate that quantifies the error on the numerical solution measured in terms of the quantity of interest would be useful.

In recent years there has been considerable progress in the development of reliable *a posteriori* error estimates and goal-oriented (adjoint-based) adaptive mesh refinement [6, 20, 7, 22, 8], in particular for compressible flows [14, 15, 18, 11, 24, 23]. These approaches are based on computing an adjoint solution related to the specific target

---

\*Institute of Aerodynamics and Flow Technology, DLR (German Aerospace Center), Lilienthalplatz 7, 38108 Braunschweig, Germany; Institute of Scientific Computing, TU Braunschweig, Germany (Ralf.Hartmann@dlr.de).

quantity under consideration. However, in many applications there is not only one but there are several quantities of interest like the collection of aerodynamic force coefficients mentioned above. Current approaches of error estimation and adjoint-based adaptive mesh refinement require the computation of one adjoint solution for each of the target quantities.

In this paper we propose an extension of this approach to the efficient and accurate computation of *multiple* target quantities. Given, say  $N$  target quantities we replace the computation of  $N$  adjoint solutions as required in current approaches by the solution of *two* auxiliary problems, namely one discrete adjoint problem and one discrete error problem where the latter can also be considered as the adjoint to the adjoint problem. In particular, the solution to the discrete error equation provides the *a posteriori* error estimation of arbitrary many target quantities. Furthermore, the solution to the adjoint problem related to an appropriately defined combination of the original target functionals provides the adjoint-based refinement indicators required for goal-oriented refinement.

This approach will be demonstrated for a symmetric interior penalty discontinuous Galerkin (DG) discretization of the compressible Navier-Stokes equations [19]. Note, however, that this approach is generic in the sense that it can be applied to other finite element discretizations (and with modifications to higher order Godunov finite volume methods [5]) as well as to other partial differential equations. Furthermore, we note that this work represents an extension of earlier work [16] on the scalar inviscid Burgers equation considering point values to the case of laminar compressible flows considering aerodynamic force coefficients.

The practical performance of this approach will be demonstrated for a laminar compressible flow. In particular, the proposed approach is compared to the standard approach of error estimation and goal-oriented mesh refinement as well as to residual-based mesh refinement. The performance of the algorithms is measured in terms of computing time required for meeting industrial as well as academic accuracy requirements on the computed force coefficients.

This paper is structured as follows: In Section 2 we recall the standard approach of error estimation and goal-oriented mesh refinement for single target functionals. We extend this approach in Section 3 to the case of multiple target functionals. In particular, we introduce the discrete error equation which replaces  $N$  adjoint problems for estimating the error with respect to  $N$  target quantities. Then in Section 4 we introduce the adaptive algorithm targeted at the accurate and efficient approximation of the  $N$  target quantities. There, we define the adjoint problem related to a target functional which appropriately combines the original target functionals. In Sections 5 and 6 we introduce the compressible Navier-Stokes equations and recall its interior penalty discontinuous Galerkin discretization given in [19]. In Section 7 we then recall the primal residual form of the discretization and the target functional modifications [12] for the drag and lift coefficients required for adjoint consistency and give the respective modification for the moment coefficient. We then derive the error representation for the modified target functionals and a Type II error bound yielding the residual-based indicators. Finally in Section 8 we demonstrate the performance of the proposed algorithm in comparison to current goal-oriented and residual-based adaptive algorithms and draw some conclusions in Section 9.

**2. Error estimation and adaptive mesh refinement for single target quantities.** We begin by recalling the general approach of duality based *a posteriori* error estimation for *single* target functionals; see e.g. [7, 10, 14] among many

others. Furthermore, we give the standard algorithm, as described in e.g. [6, 14], of goal-oriented (adjoint-based) adaptive mesh refinement tailored to the accurate and efficient computation of a single target quantity.

Let us consider the nonlinear problem

$$\mathbf{N}\mathbf{u} = 0 \quad \text{in } \Omega, \quad \mathbf{B}\mathbf{u} = 0 \quad \text{on } \Gamma, \quad (2.1)$$

where  $\Omega \in \mathbb{R}^d$ ,  $d > 1$ , is an open bounded domain with boundary  $\Gamma = \partial\Omega$ .  $\mathbf{N}$  is a nonlinear differential operator and  $\mathbf{B}$  is a possibly nonlinear boundary operator on  $\Gamma$ . Let  $\mathcal{N} : \mathbf{V} \times \mathbf{V} \rightarrow \mathbb{R}$  be a semi-linear form, nonlinear in its first and linear in its second argument, such that the nonlinear problem (2.1) is discretized as follows: find  $\mathbf{u}_h \in \mathbf{V}_h$  such that

$$\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_h. \quad (2.2)$$

Furthermore, let us assume that the discretization (2.2) is *consistent*, i.e. the exact solution  $\mathbf{u} \in \mathbf{V}$  satisfies the following equation:

$$\mathcal{N}(\mathbf{u}, \mathbf{v}) = 0 \quad \forall \mathbf{v} \in \mathbf{V}. \quad (2.3)$$

Here,  $\mathbf{V}$  is some suitably chosen function space including the exact solution  $\mathbf{u} \in \mathbf{V}$  to the primal problem (2.1) and satisfying  $\mathbf{V}_h \subset \mathbf{V}$  where  $\mathbf{V}_h$  is a discrete function space on the mesh  $\mathcal{T}_h = \{\kappa\}$  consisting of elements  $\kappa$  covering the computational domain  $\Omega$ ; cf. [2, 12] for the choice of  $\mathbf{V}$  in the case of discontinuous Galerkin methods. Subtracting (2.3) from (2.2) we then obtain the Galerkin orthogonality

$$\mathcal{N}(\mathbf{u}, \mathbf{v}_h) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_h. \quad (2.4)$$

Let  $J(\cdot)$  be a nonlinear and differentiable target functional. We define the mean-value linearisation of  $J(\cdot)$  as follows

$$\bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) = J(\mathbf{u}) - J(\mathbf{u}_h) = \int_0^1 J'[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h) d\theta, \quad (2.5)$$

where  $J'[\mathbf{w}](\cdot)$  denotes the Fréchet derivative of  $J(\cdot)$  evaluated at some  $\mathbf{w}$  in  $\mathbf{V}$ .

Analogously, for  $\mathbf{v}$  in  $\mathbf{V}$ , we define the mean-value linearisation of  $\mathcal{N}(\cdot, \mathbf{v})$

$$\begin{aligned} \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{v}) &= \mathcal{N}(\mathbf{u}, \mathbf{v}) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}) \\ &= \int_0^1 \mathcal{N}'_{\mathbf{u}}[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h, \mathbf{v}) d\theta. \end{aligned} \quad (2.6)$$

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  $\mathbf{v} \in \mathbf{V}$  fixed, at some  $\mathbf{w}$  in  $\mathbf{V}$ . Let us now introduce the following adjoint problem: find  $\mathbf{z} \in \mathbf{V}$  such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}) = \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}. \quad (2.7)$$

Choosing  $\mathbf{w} = \mathbf{u} - \mathbf{u}_h$  in (2.7), recalling the linearization performed in (2.5), and exploiting the Galerkin orthogonality (2.4) we get

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

and hence the error representation formula

$$J(\mathbf{u}) - J(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h), \quad (2.8)$$

where  $\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h)$  includes the primal residuals multiplied by the difference of the adjoint solution  $\mathbf{z}$  and an arbitrary discrete function  $\mathbf{z}_h \in \mathbf{V}_h$ .

We note that the error representation formula (2.8) depends on the unknown analytical solution  $\mathbf{z}$  to the adjoint problem (2.7) which in turn depends on the unknown analytical solution  $\mathbf{u}$  to the primal problem (2.1). Thus, in order to render these quantities computable, both  $\mathbf{u}$  and  $\mathbf{z}$  must be replaced by suitable approximations. Here, the linearizations leading to  $\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$  and  $\bar{J}(\mathbf{u}, \mathbf{u}_h; \cdot)$  are performed about  $\mathbf{u}_h$  and the adjoint solution  $\mathbf{z}$  is replaced by the solution  $\tilde{\mathbf{z}}$  to the following linearized adjoint problem: Find  $\tilde{\mathbf{z}} \in \mathbf{V}$  such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}) = J'[\mathbf{u}_h](\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}, \quad (2.9)$$

which again is replaced by the solution  $\tilde{\mathbf{z}}_h \in \tilde{\mathbf{V}}_h$  to the discrete adjoint problem

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \tilde{\mathbf{V}}_h, \quad (2.10)$$

which is usually computed on the same mesh  $\mathcal{T}_h$  used for  $\mathbf{u}_h$ , but with a higher degree polynomial. Rewriting the error representation (2.8) as follows

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \\ &= \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \tilde{\mathbf{z}}) + \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) + \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_h - \mathbf{z}_h), \end{aligned} \quad (2.11)$$

we see that replacing the adjoint solution  $\mathbf{z}$  in (2.8) by the solution  $\tilde{\mathbf{z}}_h$  to the discrete adjoint problem (2.10) to obtain the following approximate error representation

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_h - \mathbf{z}_h), \quad (2.12)$$

corresponds to ignoring in (2.11) the error  $\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \tilde{\mathbf{z}})$  due to the linearization of the adjoint problem and the error  $\mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h)$  due to the approximation of the linearized adjoint problem. In fact, it can be shown (see e.g. [7]) that the linearization and the approximation errors of the adjoint problem are of higher order (quadratic) in the discretization error,  $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ , and may thus be neglected. In fact, in a series of publications, e.g. [14, 15, 18] among many others, it has been demonstrated that the approximate error representation in (2.12) is close to the true error in the target functional.

Finally, we note that (2.12) can be localized

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa, \quad (2.13)$$

where  $|\tilde{\eta}_\kappa|$  are local error indicators including the primal local residuals weighted with the discrete adjoint solution, sometimes denoted as dual-weighted-residual indicators [7] or as adjoint-based indicators. These local indicators can be used to drive an adaptive refinement (and coarsening) algorithm specifically tailored to the accurate and efficient approximation of the target quantity  $J(\mathbf{u})$ . 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 TOL, i.e.  $|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \text{TOL}$ , then in practice we may enforce the stopping criterion  $|\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa| \leq \text{TOL}$ . If this condition

is not satisfied on the current finite element mesh  $\mathcal{T}_h$ , then the local indicators  $\eta_\kappa$  are employed as local error indicators to guide mesh refinement and coarsening. The cycle of the goal-oriented adaptive mesh refinement [14] may be outlined as follows:

ALGORITHM 2.1 (Single-target adaptive algorithm). *Adaptive algorithm for the accurate and efficient approximation of a single target quantity  $J(\mathbf{u})$ :*

1. Construct an initial mesh  $\mathcal{T}_h$ .
2. Compute  $\mathbf{u}_h \in \mathbf{V}_h$ , see (2.2), on the current mesh  $\mathcal{T}_h$ .
3. Compute  $\tilde{\mathbf{z}}_h \in \tilde{\mathbf{V}}_h$ , see (2.10), on the same mesh employed for  $\mathbf{u}_h$ , with  $\tilde{p} > p$ .
4. Evaluate the approximate error representation  $\mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_h - \mathbf{z}_h) \approx \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa$ .
5. If  $|\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa| \leq \text{TOL}$ , 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  $|\tilde{\eta}_\kappa|$  and generate a new mesh  $\mathcal{T}_h$ ; **GOTO** 2.

Again, in several publications, e.g. [7, 10, 11, 15], the strength of this adaptive algorithm has been demonstrated.

### 3. Error estimation for multiple target quantities.

**3.1. The standard approach.** Let us now consider the extension of the above analysis to the error estimation and goal-oriented mesh refinement for multiple target quantities. Given  $N$  target functional,  $J_i(\mathbf{u}), i = 1, \dots, N$ , the standard approach for deriving an error representation formula analogous to (2.8) for each  $J_i(\cdot)$  is to introduce the following  $N$  adjoint problems: find  $\mathbf{z}_i \in \mathbf{V}$  such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}_i) = \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}, \quad (3.1)$$

for  $i = 1, \dots, N$ . Analogous to (2.8) we obtain the following error representation formula

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}_i - \mathbf{z}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_i - \mathbf{z}_h), \quad (3.2)$$

for each  $J_i(\cdot), i = 1, \dots, N$ . In practice, the dual solutions  $\mathbf{z}_i, i = 1, \dots, N$ , are unknown analytically and must be approximated numerically. After linearization and approximation: find  $\tilde{\mathbf{z}}_{i,h} \in \tilde{\mathbf{V}}_h$  such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}_{i,h}) = J'_i[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \tilde{\mathbf{V}}_h, \quad (3.3)$$

this amounts to solving  $N$  systems of linear equations with the same matrix but  $N$  different right-hand side vectors. Based on the discrete adjoint solutions,  $\tilde{\mathbf{z}}_{i,h}, i = 1, \dots, N$ , the following approximate error representation formulae and local error indicators can be evaluated

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_{i,h} - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(i)}, \quad (3.4)$$

for  $i = 1, \dots, N$ .

**3.2. A new approach.** In view of the error representation formula (3.2) an alternative approach consists of considering the following error equation: find  $\mathbf{e} \in \mathbf{V}$  such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{e}, \mathbf{w}) = \mathcal{R}(\mathbf{u}_h, \mathbf{w}), \quad \forall \mathbf{w} \in \mathbf{V} \quad (3.5)$$

whose solution is simply the discretization error  $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ . We remark that in the context of duality, (3.5) may be thought of as the adjoint of the adjoint problem

and (3.2) the adjoint/adjoint-adjoint equivalence relating (3.1) to (3.5). Again after linearization, we obtain the following discrete error equation: find  $\tilde{\mathbf{e}}_h \in \tilde{\mathbf{V}}_h$  such that

$$\mathcal{N}'[\mathbf{u}_h](\tilde{\mathbf{e}}_h, \mathbf{w}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{w}_h) \quad \forall \mathbf{w}_h \in \tilde{\mathbf{V}}_h, \quad (3.6)$$

Thereby, in practice, instead of solving  $N$  discrete adjoint problems, cf. (3.3), for  $\tilde{\mathbf{z}}_{i,h} \in \tilde{\mathbf{V}}_h$  with data  $J_i(\cdot)$  and then evaluating  $\mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}}_{i,h} - \mathbf{z}_h)$  to determine the size of the error in the target functional  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , one can simply solve the discrete error equation (3.5) for the approximate error  $\tilde{\mathbf{e}}_h \in \tilde{\mathbf{V}}_h$  and evaluate

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) = \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{e}) \approx J'_i[\mathbf{u}_h](\mathbf{e}) \approx J'_i[\mathbf{u}_h](\tilde{\mathbf{e}}_h), \quad (3.7)$$

as an approximation to  $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$  for  $i = 1, \dots, N$ . When  $N > 1$  this approach is clearly much more computationally efficient than the direct method. However, a disadvantage of this second approach is that while solving the discrete error equation (3.6) for  $\tilde{\mathbf{e}}_h$  gives information concerning the size of the error in the computed target functionals  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , it does not provide the necessary local information on each element in the computational mesh to guide adaptive mesh refinement when the desired level of accuracy has not been achieved on the current mesh. On the other hand, computing the solution  $\mathbf{z}_{i,h}$ ,  $i = 1, \dots, N$ , to the  $N$  discrete adjoint problems (3.3), the approximate error representation formulae in (3.4) provide not only information concerning the size of the error in the computed target functionals, but also local error indicators  $|\tilde{\eta}_k^{(i)}|$  which can be employed for adaptive mesh design.

**4. Adaptive refinement for multiple target quantities.** In this section we propose a strategy based on solving only two auxiliary problems (the discrete error equation (3.6) and an adjoint problem subject to appropriate data which stems from a specific combined target functional, cf. (4.5) below) which provide all the necessary information needed to both estimate the size of the error in the computed target functionals, as well as provide local error indicators that can be used to drive an adaptive mesh refinement algorithm.

Given  $N$  different target functionals  $J_i(\cdot)$ ,  $i = 1, \dots, N$ ,  $N > 1$ , we would like to compute each  $J_i(\mathbf{u}_h)$  to within a given user-defined tolerance  $\text{TOL}_i$ ,  $i = 1, \dots, N$ , respectively. More precisely, we consider the following problem: find  $J_i(\mathbf{u}_h) \in \mathbb{R}$ ,  $i = 1, \dots, N$ , such that

$$|J_i(\mathbf{u}) - J_i(\mathbf{u}_h)| \leq \text{TOL}_i, \quad \text{for } i = 1, \dots, N. \quad (4.1)$$

However, as we want to define a combined target quantity  $J_c(\cdot)$  including all original target quantities  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , we weaken the requirement (4.1), and simply insist that the sum of the *relative* errors in each of the target functionals  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , is less than TOL. In practice, since  $J_i(\mathbf{u})$ ,  $i = 1, \dots, N$ , is unknown, we approximate the sum of the relative errors by

$$\sum_{i=1}^N |J_i(\mathbf{u}) - J_i(\mathbf{u}_h)| / |J_i(\mathbf{u}_h)|, \quad (4.2)$$

see [16], assuming that  $J_i(\mathbf{u}_h) \neq 0$ , for  $i = 1, \dots, N$ . As an alternative choice we might insist that the (weighted) sum of *absolute* errors in each of the target functionals  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , is less than TOL, i.e. considering

$$\sum_{i=1}^N \alpha_i |J_i(\mathbf{u}) - J_i(\mathbf{u}_h)|. \quad (4.3)$$

where  $\alpha_i > 0$ ,  $i = 1, \dots, N$ . Here, choosing  $\alpha_i = 1$ ,  $i = 1, \dots, N$ , represents the special case of considering the (unweighted) sum of absolute errors.

Let us begin by assuming that the sign of the error in each target functional  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , is known. For example, in some applications it may be known from either theoretical considerations or numerical experimentation that under mesh refinement the computed quantity of interest  $J_i(\mathbf{u}_h)$  is always either smaller or greater than the exact value  $J_i(\mathbf{u})$ , for  $i = 1, \dots, N$ . This includes the special case of monotonically convergent target quantities. For the case that under mesh refinement the quantity  $J(\mathbf{u}_h)$  converges to  $J(\mathbf{u})$  from above, for example, then the error  $J(\mathbf{u}) - J(\mathbf{u}_h)$  is always negative; analogously, when it converges from below the error is always positive.

Employing this *a priori* knowledge concerning the convergence of the target functionals, we introduce a combined target functional

$$J_c(\mathbf{v}) = \sum_{i=1}^N \omega_i J_i(\mathbf{v}), \quad (4.4)$$

where  $\omega_i = s_i/|J_i(\mathbf{u}_h)|$  and  $\omega_i = \alpha_i s_i$  in case of considering relative and weighted absolute errors (4.2) and (4.3), respectively, and  $s_i$  denotes the expected signs of the errors  $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$ ,  $i = 1, \dots, N$ , respectively. Thereby, we may now proceed as in Section 2 to derive an error representation formula for the error in the combined target functional  $J_c(\cdot)$ . To this end, we introduce the following adjoint problem: find  $\mathbf{z}_c \in \mathbf{V}$  such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}_c) = \bar{J}_c(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}. \quad (4.5)$$

where  $\bar{J}_c(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) = \sum_{i=1}^N \omega_i \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{w})$  is the mean value linearization to  $J_c$  analogous to (2.5). Thus, we now deduce the following error representation formula

$$\begin{aligned} J_c(\mathbf{u}) - J_c(\mathbf{u}_h) &= \sum_{i=1}^N \omega_i (J_i(\mathbf{u}) - J_i(\mathbf{u}_h)) = \sum_{i=1}^N \omega_i \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) \\ &= \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) \end{aligned} \quad (4.6)$$

In general, the signs  $s_i$ ,  $i = 1, \dots, N$ , will not be known *a priori*. Thereby, we must first solve the discrete error equation (3.6) for  $\tilde{\mathbf{e}}_h$  and evaluate  $\tilde{s}_i = \text{sgn}(J'_i[\mathbf{u}_h](\tilde{\mathbf{e}}_h))$ ,  $i = 1, \dots, N$ . Then, the adjoint problem (4.5) may be solved computationally using the predicted values of  $s_i$ ,  $i = 1, \dots, N$  in  $J_c(\cdot)$ : Find  $\tilde{\mathbf{z}}_{c,h} \in \tilde{\mathbf{V}}_h$  such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}_{c,h}) = J'_c[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \tilde{\mathbf{V}}_h. \quad (4.7)$$

and the approximate error representation formula can be evaluated:

$$J_c(\mathbf{u}) - J_c(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) \approx \mathcal{R}(\mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa \quad (4.8)$$

This now provides both global information concerning the size of the error in the combined target functional  $J_c(\cdot)$ , as well as local information necessary for adaptive mesh refinement. Thus, the cycle of the adaptive algorithm can be outlined as follows:

ALGORITHM 4.1 (Multi-target adaptive algorithm). *Adaptive algorithm for the accurate and efficient approximation of multiple target quantities  $J_i(u)$ ,  $i = 1, \dots, N$ :*

1. Construct an initial mesh  $\mathcal{T}_h$ .

2. Compute  $\mathbf{u}_h \in \mathbf{V}_h$ , see (2.2), on the current mesh  $\mathcal{T}_h$ .
3. Compute  $\tilde{\mathbf{e}}_h \in \tilde{\mathbf{V}}_h$ , see (3.6), on the same mesh employed for  $\mathbf{u}_h$ , with  $\tilde{p} > p$ .
4. Evaluate  $J_i(\mathbf{u}) - J_i(\mathbf{u}_h) \approx J'_i[\mathbf{u}_h](\tilde{\mathbf{e}}_h) =: \psi_i$ ,  $i = 1, \dots, N$ .
5. If  $|\psi_i| \leq \text{TOL}_i$  for all  $i = 1, \dots, N$ , then STOP.
6. Build the target quantity  $J_c$  based on  $\tilde{s}_i = \text{sgn}(\psi_i)$ ,  $i = 1, \dots, N$ .
7. Compute  $\tilde{\mathbf{z}}_{c,h} \in \tilde{\mathbf{V}}_h$ , see (4.7), on the same mesh employed for  $\mathbf{u}_h$ , with  $\hat{p} > p$ .
8. Evaluate the approximate error representation  $\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa$ , see (4.8).
9. If  $|\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa| \leq \text{TOL}$ , where  $\text{TOL}$  is a given tolerance, then STOP.
10. Otherwise, refine and coarsen a fixed fraction of the total number of elements according to the size of  $|\tilde{\eta}_\kappa|$  and generate a new mesh  $\mathcal{T}_h$ ; GOTO 2.

Here, the stopping criterion in lines (5) or (9) of Algorithm 4.1 can be used corresponding to formulae (4.1) and (4.2)  $\leq \text{TOL}$  or (4.3)  $\leq \text{TOL}$ , respectively. This approach leads to the solution of only two auxiliary problems, in comparison to the  $N$  required for the standard approach.

We note that this approach has previously been developed for and applied to the discontinuous Galerkin discretization of the inviscid 1d Burgers equation in [16] considering the sum of relative errors of point values of the solution. In the following sections we apply this approach to the interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations [19] considering sums of relative and absolute errors of aerodynamic force coefficients including pressure induced and viscous drag, lift and moment coefficients.

**5. The compressible Navier-Stokes equations.** In this section we consider the two-dimensional stationary compressible Navier-Stokes equations

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u})) = 0 \quad \text{in } \Omega, \quad (5.1)$$

subject to various boundary conditions, e.g. no-slip wall boundary conditions with vanishing velocity  $\mathbf{v} = (v_1, v_2)^\top = 0$  at isothermal walls  $\Gamma_{\text{iso}}$  where  $T = T_{\text{wall}}$ , or at adiabatic walls  $\Gamma_{\text{adia}}$  where  $\mathbf{n} \cdot \nabla T = 0$ ; see [17] or [19] for more details. The vector of conservative variables  $\mathbf{u}$  and the convective fluxes  $\mathcal{F}^c(\mathbf{u})$  are as defined in

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{bmatrix}, \quad \mathbf{f}_1^c(\mathbf{u}) = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho H v_1 \end{bmatrix} \quad \text{and} \quad \mathbf{f}_2^c(\mathbf{u}) = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho H v_2 \end{bmatrix}. \quad (5.2)$$

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

$$\mathbf{f}_1^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{1j} v_j + \mathcal{K} T_{x_1} \end{bmatrix} \quad \text{and} \quad \mathbf{f}_2^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{2j} v_j + \mathcal{K} T_{x_2} \end{bmatrix}. \quad (5.3)$$

Here  $T$  denotes the temperature given by  $e = c_v T$ ,  $\mathcal{K}$  is the thermal conductivity coefficient and  $\underline{\tau}$  is the viscous stress tensor defined by  $\underline{\tau} = \mu (\nabla \mathbf{v} + (\nabla \mathbf{v})^\top - \frac{2}{3} (\nabla \cdot \mathbf{v}) I)$  where  $\mu$  is the dynamic viscosity coefficient. Writing  $G$  to denote the homogeneity tensor, with  $G_{ij}(\mathbf{u}) = \partial \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) / \partial u_{x_j}$ , for  $i, j = 1, 2$ , cf. [17], the viscous fluxes may be written in the form  $\mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) = G_{ij}(\mathbf{u}) \partial u / \partial x_j$ ,  $i = 1, 2$ , or more compactly, we may write  $\mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = G(\mathbf{u}) \nabla \mathbf{u}$ .



**6. DG discretization of the compressible Navier-Stokes equations.** In this section we recall the consistent and adjoint consistent interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations as derived in [12] and the interior penalty stabilization term as introduced in [19].

First, we begin by introducing some notation. We assume that  $\Omega$  can be subdivided into shape-regular meshes  $\mathcal{T}_h = \{\kappa\}$  consisting of quadrilateral elements  $\kappa$ . Here,  $h$  denotes the piecewise constant mesh function defined by  $h|_\kappa \equiv h_\kappa = \text{diam}(\kappa)$  for all  $\kappa \in \mathcal{T}_h$ . Let us assume that each  $\kappa \in \mathcal{T}_h$  is an image of a fixed reference element  $\hat{\kappa}$ , that is,  $\kappa = \sigma_\kappa(\hat{\kappa})$  for all  $\kappa \in \mathcal{T}_h$ . Here, we shall only consider the case when  $\hat{\kappa}$  is the open unit square in  $\mathbb{R}^2$ . Furthermore the mapping  $\sigma_\kappa$  of the reference element  $\hat{\kappa}$  to the element  $\kappa$  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  $\sigma_\kappa$  is given by a bilinear function; In order to represent curved boundaries mappings can be used that include polynomials of higher degree on boundary elements; see [10] for more details about curved elements. On the reference element  $\hat{\kappa}$  we define spaces of tensor product polynomials of degree  $p \geq 0$  as follows:

$$\mathcal{Q}_p(\hat{\kappa}) = \text{span} \{ \hat{\mathbf{x}}^\alpha : 0 \leq \alpha_i \leq p, i = 1, 2 \}, \quad (6.1)$$

where  $\alpha$  denotes a multi-index and  $\hat{\mathbf{x}}^\alpha = \prod_{i=1}^2 \hat{x}_i^{\alpha_i}$ . Finally, we introduce the finite element space  $\mathbf{V}_h^p$  consisting of discontinuous vector-valued product polynomial functions of degree  $p \geq 0$ , defined by

$$\mathbf{V}_h^p = \{ \mathbf{v}_h \in [L_2(\Omega)]^4 : \mathbf{v}_h|_\kappa \circ \sigma_\kappa \in [\mathcal{Q}_p(\hat{\kappa})]^4, \kappa \in \mathcal{T}_h \}. \quad (6.2)$$

Suppose that  $\mathbf{v}|_\kappa \in [H^1(\kappa)]^m$ ,  $m \in \mathbb{N}$ , for each  $\kappa \in \mathcal{T}_h$ . Let  $\kappa$  and  $\kappa'$  be two adjacent elements of  $\mathcal{T}_h$  and  $\mathbf{x}$  be an arbitrary point on the interior edge  $e = \partial\kappa \cap \partial\kappa'$ . By  $\mathbf{v}_\kappa^\pm$  we denote the traces of  $\mathbf{v}$  taken from within the interior of  $\kappa$  and  $\kappa'$ , respectively. Since below it will always be clear from the context which element  $\kappa$  in the subdivision  $\mathcal{T}_h$  the quantities  $\mathbf{v}_\kappa^+$  and  $\mathbf{v}_\kappa^-$  correspond to, for the sake of notational simplicity, we shall suppress the letter  $\kappa$  in the subscript and write, respectively,  $\mathbf{v}^+$  and  $\mathbf{v}^-$ , instead.

We now define average and jump operators for vector- and matrix-valued functions. To this end, let  $\kappa^+$  and  $\kappa^-$  be two adjacent elements of  $\mathcal{T}_h$  and  $\mathbf{x}$  be an arbitrary point on the interior edge  $e = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_{\mathcal{T}}$ . Moreover, let  $\mathbf{v}$  and  $\underline{\tau}$  be vector- and matrix-valued functions, respectively, that are smooth inside each element  $\kappa^\pm$ . By  $\mathbf{v}^\pm := \mathbf{v}|_{\partial\kappa^\pm}$  and  $\underline{\tau}^\pm := \underline{\tau}|_{\partial\kappa^\pm}$  we denote the traces of, respectively,  $\mathbf{v}$  and  $\underline{\tau}$  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 jump at  $\mathbf{x} \in e$  is given by  $[\![\mathbf{v}]\!] = \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}$ . On a boundary edge  $e \subset \Gamma$ , we set  $\{\mathbf{v}\} = \mathbf{v}$ ,  $\{\underline{\tau}\} = \underline{\tau}$  and  $[\![\mathbf{v}]\!] = \mathbf{v} \otimes \mathbf{n}$ . For matrices  $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m \times n}$ ,  $m, n \geq 1$ , 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  $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k w_l$ .

The discontinuous Galerkin discretization of the compressible Navier–Stokes equa-

tions (5.1) is given by: Find  $\mathbf{u}_h \in \mathbf{V}_h^p$  such that

$$\begin{aligned} \mathcal{N}(\mathbf{u}_h, \mathbf{v}) &\equiv - \int_{\Omega} \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v} \, d\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) \cdot \mathbf{v}^+ \, ds \\ &\quad + \int_{\Omega} \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v} \, d\mathbf{x} - \int_{\Gamma_x} \{\mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h)\} : \llbracket \mathbf{v} \rrbracket \, ds \\ &\quad - \int_{\Gamma_x} \{G^\top(\mathbf{u}_h) \nabla_h \mathbf{v}\} : \llbracket \mathbf{u}_h \rrbracket \, ds + \int_{\Gamma_x} \underline{\delta}(\mathbf{u}_h) : \llbracket \mathbf{v} \rrbracket \, ds \\ &\quad + \mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) = 0 \end{aligned} \quad (6.3)$$

for all  $\mathbf{v}$  in  $\mathbf{V}_h^p$ . Here, the numerical flux  $\mathcal{H}(\cdot, \cdot, \cdot)$  may be chosen to be any two-point monotone Lipschitz function which is consistent, i.e.  $\mathcal{H}(\mathbf{v}, \mathbf{v}, \mathbf{n}) = \mathcal{F}^c(\mathbf{v}) \cdot \mathbf{n}$ , and conservative, i.e.  $\mathcal{H}(\mathbf{v}, \mathbf{w}, \mathbf{n}) = -\mathcal{H}(\mathbf{w}, \mathbf{v}, -\mathbf{n})$ . The penalization term is given by

$$\underline{\delta}(\mathbf{u}_h) = \underline{\delta}^{\text{ip}}(\mathbf{u}_h) = C_{\text{IP}} \frac{p^2}{h_e} \{G(\mathbf{u}_h)\} \llbracket \mathbf{u}_h \rrbracket, \quad (6.4)$$

where

$$h_e = \min(\text{meas}(\kappa), \text{meas}(\kappa')) / \text{meas}(e) \quad (6.5)$$

represents the element dimension orthogonal to the edge  $e$  of the elements  $\kappa$  and  $\kappa'$  adjacent to  $e$ . Furthermore,  $C_{\text{IP}}$  is a positive constant, which, for reasons of stability, must be chosen sufficiently large. Finally, the boundary terms included in  $\mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v})$  are given by

$$\begin{aligned} \mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) &= \int_{\Gamma} \mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}^+) \cdot \mathbf{v}^+ \, ds + \int_{\Gamma} \underline{\delta}_\Gamma(\mathbf{u}_h^+) : \mathbf{v} \otimes \mathbf{n} \, ds, \\ &\quad - \int_{\Gamma} \mathbf{n} \cdot \mathcal{F}_\Gamma^v(\mathbf{u}_h^+, \nabla_h \mathbf{u}_h^+) \mathbf{v}^+ \, ds \\ &\quad - \int_{\Gamma} (G_\Gamma^\top(\mathbf{u}_h^+) \nabla_h \mathbf{v}_h^+) : (\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \otimes \mathbf{n} \, ds. \end{aligned} \quad (6.6)$$

where the penalization term at the boundary is given by

$$\underline{\delta}_\Gamma(\mathbf{u}_h) = \underline{\delta}_\Gamma^{\text{ip}}(\mathbf{u}_h) = C_{\text{IP}} \frac{p^2}{h_e} G_\Gamma(\mathbf{u}_h) (\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)) \otimes \mathbf{n}. \quad (6.7)$$

Here, the viscous boundary flux  $\mathcal{F}_\Gamma^v$  and the corresponding homogeneity tensor  $G_\Gamma$  are defined by

$$\mathcal{F}_\Gamma^v(\mathbf{u}_h, \nabla \mathbf{u}_h) = \mathcal{F}^v(\mathbf{u}_\Gamma(\mathbf{u}_h), \nabla \mathbf{u}_h) = G_\Gamma(\mathbf{u}_h) \nabla \mathbf{u}_h = G(\mathbf{u}_\Gamma(\mathbf{u}_h)) \nabla \mathbf{u}_h. \quad (6.8)$$

Furthermore, on adiabatic boundaries  $\Gamma_{\text{adia}} \subset \Gamma_W$ ,  $\mathcal{F}_\Gamma^v$  and  $G_\Gamma$  are modified such that  $\mathbf{n} \cdot \nabla T = 0$ . Finally, we define

$$\mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) = \mathbf{n} \cdot \mathcal{F}_\Gamma^c(\mathbf{u}_h^+) = \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_\Gamma(\mathbf{u}_h^+)), \quad (6.9)$$

where the boundary function  $\mathbf{u}_\Gamma(\cdot)$  is given by  $\mathbf{u}_\Gamma(\mathbf{w}) = (w_1, 0, 0, w_4)^\top$  on  $\Gamma_{\text{adia}}$ , and by  $\mathbf{u}_\Gamma(\mathbf{w}) = (w_1, 0, 0, w_1 c_v T_{\text{wall}})^\top$  on  $\Gamma_{\text{iso}}$ ; see [17] or [19] for the treatment of other boundary conditions. We note that the boundary function  $\mathbf{u}_\Gamma(\cdot)$  is consistent, i.e. on all boundary parts,  $\mathbf{u}_\Gamma(\cdot)$  is chosen such that the exact solution  $\mathbf{u}$  to (5.1) satisfies  $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$ . As a consequence also  $\underline{\delta}_\Gamma(\cdot)$  as defined in (6.7) is consistent. In fact, the exact solution  $\mathbf{u}$  to (5.1) satisfies  $\underline{\delta}_\Gamma(\mathbf{u}) = 0$ . Finally, we note that the viscous and convective boundary fluxes in (6.8) and (6.9) are chosen so that the discretization of boundary terms in combination with specific target quantities is adjoint consistent. We refer to [12] for a detailed analysis on the adjoint consistency property in this case.

### 7. Consistency, adjoint consistency and a posteriori error estimates.

Following [19] we using integration by parts in (6.3) and obtain the primal residual form given by: find  $\mathbf{u}_h \in \mathbf{V}_h^p$  such that

$$\begin{aligned} \mathcal{R}(\mathbf{u}_h, \mathbf{v}) \equiv & \int_{\Omega} \mathbf{R}(\mathbf{u}_h) \cdot \mathbf{v} \, dx + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathbf{r}(\mathbf{u}_h) \cdot \mathbf{v}^+ + \underline{\rho}(\mathbf{u}_h) : \nabla \mathbf{v}^+ \, ds \\ & + \int_{\Gamma} \mathbf{r}_{\Gamma}(\mathbf{u}_h) \cdot \mathbf{v}^+ + \underline{\rho}_{\Gamma}(\mathbf{u}_h) : \nabla \mathbf{v}^+ \, ds = 0 \quad \forall \mathbf{v} \in \mathbf{V}_h^p, \end{aligned} \quad (7.1)$$

where the primal element, interior face and boundary residuals are given by

$$\begin{aligned} \mathbf{R}(\mathbf{u}_h) &= -\nabla \cdot \mathcal{F}^c(\mathbf{u}_h) + \nabla \cdot \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) && \text{in } \kappa, \kappa \in \mathcal{T}_h, \\ \mathbf{r}(\mathbf{u}_h) &= \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) - \frac{1}{2} \llbracket \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) \rrbracket - \mathbf{n} \cdot \underline{\delta}(\mathbf{u}_h), \\ \underline{\rho}(\mathbf{u}_h) &= \frac{1}{2} \left( G(\mathbf{u}_h) \llbracket \mathbf{u}_h \rrbracket \right)^\top && \text{on } \partial\kappa \setminus \Gamma, \kappa \in \mathcal{T}_h, \\ \mathbf{r}_{\Gamma}(\mathbf{u}_h) &= \mathbf{n} \cdot \left( \mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{F}_{\Gamma}^c(\mathbf{u}_h^+) - \mathcal{F}^v(\mathbf{u}_h^+, \nabla \mathbf{u}_h^+) + \mathcal{F}_{\Gamma}^v(\mathbf{u}_h^+, \nabla \mathbf{u}_h^+) \right) - \mathbf{n} \cdot \underline{\delta}_{\Gamma}(\mathbf{u}_h), \\ \underline{\rho}_{\Gamma}(\mathbf{u}_h) &= (G_{\Gamma}^{\top}(\mathbf{u}_h^+) : (\mathbf{u}_h^+ - \mathbf{u}_{\Gamma}(\mathbf{u}_h^+)) \otimes \mathbf{n})^\top && \text{on } \Gamma. \end{aligned} \quad (7.2)$$

As shown in [19] the exact solution  $\mathbf{u}$  to (5.1) satisfies

$$\mathbf{R}(\mathbf{u}) = 0, \quad \mathbf{r}(\mathbf{u}) = 0, \quad \underline{\rho}(\mathbf{u}) = 0, \quad \mathbf{r}_{\Gamma}(\mathbf{u}) = 0, \quad \underline{\rho}_{\Gamma}(\mathbf{u}) = 0.$$

Thereby, the discretization (6.3) is consistent.

In the following, we consider the aerodynamic force coefficients of a body immersed in a viscous fluid with inlet flow at the angle of attack  $\alpha$ . In particular, we consider the pressure induced and viscous drag coefficients,  $c_{dp}$  and  $c_{df}$ , respectively,

$$J_{c_{dp}}(\mathbf{u}) = \frac{1}{C_{\infty}} \int_{\Gamma_W} p \mathbf{n} \cdot \boldsymbol{\psi}_d \, ds, \quad J_{c_{df}}(\mathbf{u}) = -\frac{1}{C_{\infty}} \int_{\Gamma_W} (\underline{\tau} \mathbf{n}) \cdot \boldsymbol{\psi}_d \, ds, \quad (7.3)$$

the pressure induced and viscous lift coefficients,  $c_{lp}$  and  $c_{lf}$ , respectively,

$$J_{c_{lp}}(\mathbf{u}) = \frac{1}{C_{\infty}} \int_{\Gamma_W} p \mathbf{n} \cdot \boldsymbol{\psi}_l \, ds, \quad J_{c_{lf}}(\mathbf{u}) = -\frac{1}{C_{\infty}} \int_{\Gamma_W} (\underline{\tau} \mathbf{n}) \cdot \boldsymbol{\psi}_l \, ds, \quad (7.4)$$

and the pressure induced and viscous moment coefficients,  $c_{mp}$  and  $c_{mf}$ , respectively,

$$\begin{aligned} J_{c_{mp}}(\mathbf{u}) &= \frac{1}{C_{\infty}} \int_{\Gamma_W} (\mathbf{x} - \mathbf{x}_{\text{ref}}) \times p \mathbf{n} \, ds, \\ J_{c_{mf}}(\mathbf{u}) &= -\frac{1}{c_{\text{ref}} C_{\infty}} \int_{\Gamma_W} (\mathbf{x} - \mathbf{x}_{\text{ref}}) \times \underline{\tau} \mathbf{n} \, ds, \end{aligned} \quad (7.5)$$

as well as the total drag, lift and moment coefficients,  $c_d = c_{dp} + c_{df}$ ,  $c_l = c_{lp} + c_{lf}$  and  $c_m = c_{mp} + c_{mf}$ , respectively, given by

$$J_{c_d}(\mathbf{u}) = J_{c_{dp}}(\mathbf{u}) + J_{c_{df}}(\mathbf{u}), \quad (7.6)$$

$$J_{c_l}(\mathbf{u}) = J_{c_{lp}}(\mathbf{u}) + J_{c_{lf}}(\mathbf{u}), \quad (7.7)$$

$$J_{c_m}(\mathbf{u}) = J_{c_{mp}}(\mathbf{u}) + J_{c_{mf}}(\mathbf{u}). \quad (7.8)$$

Here,  $\boldsymbol{\psi}_d = (\cos(\alpha), \sin(\alpha))^\top$  and  $\boldsymbol{\psi}_l = (-\sin(\alpha), \cos(\alpha))^\top$  for the drag and lift coefficients, respectively. Furthermore,  $C_\infty = \frac{1}{2}\gamma p_\infty M_\infty^2 \bar{l} = \frac{1}{2}\gamma \frac{|\mathbf{v}_\infty|^2}{c_\infty^2} p_\infty \bar{l} = \frac{1}{2}\rho_\infty |\mathbf{v}_\infty|^2 \bar{l}$ , where  $M_\infty$  denotes the Mach number at free-stream conditions,  $c_\infty$  is the free-stream speed of sound defined by  $c_\infty^2 = \gamma p_\infty / \rho_\infty$ , where  $p_\infty$  and  $\rho_\infty$  denote the freestream pressure and density, respectively, and  $\bar{l}$  denotes the reference (chord) length of the body. Finally,  $\mathbf{x}_{\text{ref}}$  is the moment reference point, usually  $\mathbf{x}_{\text{ref}} = (0.25, 0)$  for a profile of unit length, and  $c_{\text{ref}}$  is a dimensionless reference chord length usually equal to  $\bar{l}$ . Here, we adopt the notation  $\mathbf{a} \times \mathbf{b} = a_1 b_2 - a_2 b_1$  for  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ .

In [12] it has been shown that the discretization (6.3) together with the total drag or lift coefficients,  $c_d$  and  $c_l$ , is adjoint consistent, see also [2, 9, 23], provided the target functionals,  $J_{c_d}(\mathbf{u})$  and  $J_{c_l}(\mathbf{u})$ , respectively, are modified as follows

$$\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_\Gamma(\mathbf{u}_h)) + \int_{\Gamma_W} \underline{\delta}_\Gamma(\mathbf{u}_h) : \mathbf{z}_\Gamma \otimes \mathbf{n} \, ds, \quad (7.9)$$

where the second and third component of  $\mathbf{z}_\Gamma = \frac{1}{C_\infty}(0, \psi_1, \psi_2, 0)^\top$  represent the corresponding constant boundary values of the continuous adjoint solution  $\mathbf{z}$  related to the total drag and lift coefficient; cf. [12] for more details. Similarly, it can be shown that the discretization (6.3) together with the total moment coefficient  $c_m$  is adjoint consistent provided the target functional  $J_{c_m}(\mathbf{u})$  is modified like in (7.9) with  $\mathbf{z}_\Gamma(\mathbf{x}) = \frac{1}{c_{\text{ref}} C_\infty}(0, -d_2(\mathbf{x}), d_1(\mathbf{x}), 0)^\top$  and  $\mathbf{d}(\mathbf{x}) = \mathbf{x} - \mathbf{x}_{\text{ref}}$  on  $\Gamma_W$ .

Finally, we recall from [12] that considering one of the pressure induced or viscous force coefficients,  $c_{\star p}$  and  $c_{\star f}$ , respectively, where  $\star$  stands for  $d, l$  or  $m$ , there is no modification of  $J_{c_{\star p}}(\mathbf{u})$  or  $J_{c_{\star f}}(\mathbf{u})$  such that the discretization (6.3) is adjoint consistent. Nevertheless, we modify the viscous force coefficients  $J_{c_{\star f}}(\mathbf{u})$  like in (7.9) and the pressure induced force coefficients  $J_{c_{\star p}}(\mathbf{u})$  as follows

$$\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_\Gamma(\mathbf{u}_h)), \quad (7.10)$$

in order to ensure that we recover the adjoint consistent modification of the total force coefficients  $J_{c_\star}(\mathbf{u})$ , see (7.9), simply by adding the modified pressure induced and viscous force coefficients,  $\tilde{J}_{c_\star}(\mathbf{u}) = \tilde{J}_{c_{\star p}}(\mathbf{u}) + \tilde{J}_{c_{\star f}}(\mathbf{u})$ , for  $\star \in \{d, l, m\}$ . We note, that the functional modifications given in (7.9) and (7.10) are consistent modifications of the target functionals. In fact, due to the consistency of the boundary function  $\mathbf{u}_\Gamma(\cdot)$  and the penalization term  $\underline{\delta}_\Gamma(\cdot)$  the exact solution  $\mathbf{u}$  to (5.1) satisfies  $\tilde{J}(\mathbf{u}) = J(\mathbf{u})$ .

We now can derive following *a posteriori* error estimate of the discontinuous Galerkin discretization (6.3) for the error measured in terms of one of the modified aerodynamic force coefficients defined above.

LEMMA 7.1 (Error representation formula). *Let  $\mathbf{u}$  and  $\mathbf{u}_h$  denote the solutions of (5.1) and (6.3), respectively, and suppose that the adjoint problem (2.7), with  $J$  being replaced by  $\tilde{J}$ , is well-posed. Then, the following error representation holds:*

$$J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h) = \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa, \quad (7.11)$$

where the local error contributions  $\eta_\kappa$ ,  $\kappa \in \mathcal{T}_h$ , are given by

$$\begin{aligned} \eta_\kappa = & \int_\kappa \mathbf{R}(\mathbf{u}_h) \cdot (\mathbf{z} - \mathbf{z}_h) \, d\mathbf{x} + \int_{\partial\kappa \setminus \Gamma} \mathbf{r}(\mathbf{u}_h) \cdot (\mathbf{z} - \mathbf{z}_h)^+ + \underline{\rho}(\mathbf{u}_h) : \nabla(\mathbf{z} - \mathbf{z}_h)^+ \, ds \\ & + \int_{\kappa \cap \Gamma} \mathbf{r}_\Gamma(\mathbf{u}_h) \cdot (\mathbf{z} - \mathbf{z}_h)^+ + \underline{\rho}_\Gamma(\mathbf{u}_h) : \nabla(\mathbf{z} - \mathbf{z}_h)^+ \, ds. \end{aligned} \quad (7.12)$$

Here,  $\mathbf{z}$  is the solution to the adjoint problem (2.7) with  $J$  being replaced by  $\tilde{J}$ .  $\mathbf{z}_h$  may be any discrete function in  $\mathbf{V}_h^p$ , and  $\mathbf{R}(\mathbf{u}_h)$ ,  $\mathbf{r}(\mathbf{u}_h)$ ,  $\underline{\rho}(\mathbf{u})$ ,  $\mathbf{r}_\Gamma(\mathbf{u}_h)$  and  $\underline{\rho}_\Gamma(\mathbf{u}_h)$  are the primal element, interior face and boundary residuals, respectively, defined in (7.2).

*Proof.* Due to the consistency of the functional modification we have  $J(\mathbf{u}) = \tilde{J}(\mathbf{u})$ . According to the derivation along the lines in Section 2 we have

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

and after integration by parts on each element  $\kappa \in \mathcal{T}_h$  like in (7.1) we obtain (7.11).  $\square$

We are now ready to prove the following error bound, referred to as the Type II error bound in the literature, cf. [21]. This error bound gives rise to the residual-based indicators  $|\eta_\kappa^{(\text{res})}|$ , see (7.14) below.

**COROLLARY 7.2** (Type II error bound). *Given that the assumptions of Lemma 7.1 hold, suppose that  $\mathbf{z} \in [H^s(\Omega)]^4$ ,  $2 \leq s \leq p+1$ , and that we have found a constant  $C_{\text{stab}}$  such that  $\|\mathbf{z}\|_{H^s(\Omega)} \leq C_{\text{stab}}$ . Then, the following Type II a posteriori error bound holds:*

$$|J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h)| \leq C \left( \sum_{\kappa \in \mathcal{T}_h} \left( \eta_\kappa^{(\text{res})} \right)^2 \right)^{1/2}, \quad (7.13)$$

where the residual-based indicators  $\eta_\kappa^{(\text{res})}$ ,  $\kappa \in \mathcal{T}_h$ , are given by

$$\eta_\kappa^{(\text{res})} = h_\kappa^{t+1} \|\mathbf{R}(\mathbf{u}_h)\|_\kappa + h_\kappa^{t+1/2} \|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|_{\partial\kappa} + h_\kappa^{t-1/2} \|\underline{\rho}_{\partial\kappa}(\mathbf{u}_h)\|_{\partial\kappa}, \quad (7.14)$$

where  $t = \min(s, p)$ . Here,  $\mathbf{r}_{\partial\kappa} = \mathbf{r}$  on  $\partial\kappa \setminus \Gamma$  and  $\mathbf{r}_{\partial\kappa} = \mathbf{r}_\Gamma$  on  $\Gamma$ , i.e.

$$\|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|_{\partial\kappa}^2 = \|\mathbf{r}(\mathbf{u}_h)\|_{\partial\kappa \setminus \Gamma}^2 + \|\mathbf{r}_\Gamma(\mathbf{u}_h)\|_\Gamma^2,$$

and analogously for  $\underline{\rho}_{\partial\kappa}$ ,

$$\|\underline{\rho}_{\partial\kappa}(\mathbf{u}_h)\|_{\partial\kappa}^2 = \|\underline{\rho}(\mathbf{u}_h)\|_{\partial\kappa \setminus \Gamma}^2 + \|\underline{\rho}_\Gamma(\mathbf{u}_h)\|_\Gamma^2,$$

where  $\mathbf{R}(\mathbf{u}_h)$ ,  $\mathbf{r}(\mathbf{u}_h)$ ,  $\underline{\rho}(\mathbf{u})$ ,  $\mathbf{r}_\Gamma(\mathbf{u}_h)$  and  $\underline{\rho}_\Gamma(\mathbf{u}_h)$  are the primal element, interior face and boundary residuals, respectively, defined in (7.2).

*Proof.* Selecting  $\mathbf{z}_h = P_h \mathbf{z} \in \mathbf{V}_h$  in (7.11), applying the Cauchy-Schwarz inequality together with standard approximation estimates for  $\mathbf{z} - P_h \mathbf{z}$  (see [18] for more details) we obtain (7.13).  $\square$

Note that Corollary 7.2 is the extension of Corollary 4.5 in [18] for the adjoint inconsistent discretization of the compressible Navier-Stokes equations given in [18] to the adjoint consistent discretization [19] considered in this article.

As described in Section 2 the adjoint-based indicators  $|\eta_\kappa|$ , see (7.12), can be used for goal-oriented refinement. The residual-based indicators  $|\eta_\kappa^{(\text{res})}|$ , see (7.14) include primal residuals but do not depend on an adjoint solution and are thus much cheaper to evaluate than adjoint-based indicators. In the following section we compare the proposed multi-target adaptive algorithm to current adjoint-based algorithms as well as to a residual-based algorithm.

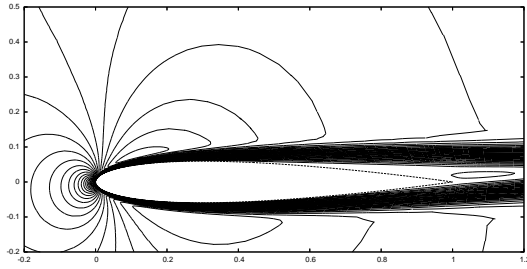


FIG. 8.1. ADIGMA MTC-3 test case: Mach number isolines. The laminar compressible flow at  $M = 0.5$ ,  $\alpha = 2^\circ$ ,  $Re = 5000$  is a subsonic flow with a laminar separation at the trailing edge.

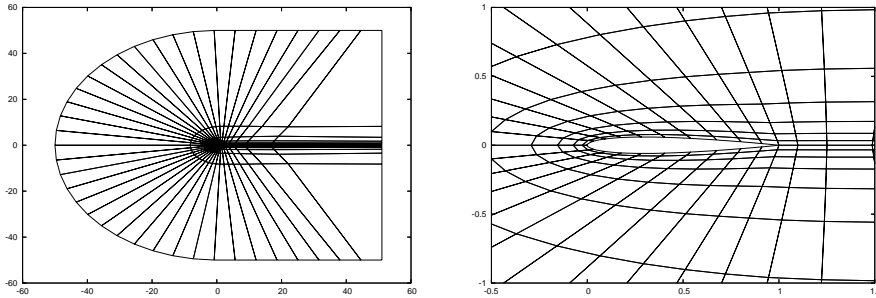


FIG. 8.2. Coarse mesh with 400 elements: (left) full and (right) detailed view.

**8. Numerical results.** In this section we present several numerical results demonstrating the performance of the *a posteriori* error estimation and goal-oriented mesh refinement for the accurate and efficient approximation of multiple force coefficients. To this end, we consider the MTC-3 test case defined in the European project ADIGMA [1]: A laminar compressible flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack  $\alpha = 2^\circ$ , and Reynolds number  $Re = 5000$  with adiabatic no-slip wall boundary condition imposed on the airfoil geometry. This is a steady subsonic flow with a large laminar separation at the trailing edge, see Figure 8.1. The adaptive algorithms performed in the following will be based on the coarse mesh of 400 quadrilateral elements shown in Figure 8.2.

In this test case the most relevant aerodynamic force coefficients, namely the pressure induced and viscous drag coefficients,  $c_{dp}$  and  $c_{df}$ , respectively, the total lift coefficient  $c_l$  and the total moment coefficient  $c_m$  shall be computed up to a predefined error tolerance  $TOL$ . In the EU project ADIGMA the following industrial accuracy requirements have been defined for this test case:

$$\begin{aligned} |J_{c_{dp}}(\mathbf{u}) - J_{c_{dp}}(\mathbf{u}_h)| &\leq TOL_{c_{dp}} = 5 \cdot 10^{-4}, \\ |J_{c_{df}}(\mathbf{u}) - J_{c_{df}}(\mathbf{u}_h)| &\leq TOL_{c_{df}} = 5 \cdot 10^{-4}, \\ |J_{c_l}(\mathbf{u}) - J_{c_l}(\mathbf{u}_h)| &\leq TOL_{c_l} = 5 \cdot 10^{-3}, \\ |J_{c_m}(\mathbf{u}) - J_{c_m}(\mathbf{u}_h)| &\leq TOL_{c_m} = 5 \cdot 10^{-4}. \end{aligned} \quad (8.1)$$

Additionally, for academic purposes we define the following accuracy requirements:

$$|J_\star(\mathbf{u}) - J_\star(\mathbf{u}_h)| \leq \frac{1}{5} TOL_\star, \quad \text{for } \star \in \{c_{dp}, c_{df}, c_l, c_m\}, \quad (8.2)$$

where  $TOL_\star$  stands for the tolerances defined in (8.1). Thereby, the academic accu-

racy requirements are stronger in the sense that the tolerances for each of the force coefficients is a fifth of the tolerances in the industrial accuracy requirements.

We remark that in view of the discretization being adjoint consistent for specific force coefficients only, see Section 7, it would be preferable to approximate the total drag coefficient  $c_d$  rather than separately its pressure induced and viscous parts,  $c_{dp}$  and  $c_{df}$ , respectively. Nevertheless, for e.g. wing design at industry some force coefficients are important to be accurately approximated separating pressure induced and viscous parts like is requested for the drag coefficient in this case.

Finally we note, that very fine grid computations, also with higher polynomial degrees, have been performed in order to obtain the following reference values (‘true’ values):

$$\begin{aligned} J_{c_{dp}}(\mathbf{u}) &= c_{dp}^{\text{ref}} = 0.0238006, & J_{c_{df}}(\mathbf{u}) &= c_{df}^{\text{ref}} = 0.0322805, \\ J_{c_l}(\mathbf{u}) &= c_l^{\text{ref}} = 0.037468, & J_{c_m}(\mathbf{u}) &= c_m^{\text{ref}} = -0.01662. \end{aligned} \quad (8.3)$$

These reference values will be used to compare with the force coefficients being evaluated on coarser meshes and using lower polynomial degrees in the following numerical examples. Also the accuracy of a posteriori error estimates will be investigated based on the reference values in (8.3).

In all subsequent computations we choose the penalization constant to be  $C_{IP} = 20$  in (6.4). The solutions  $\mathbf{u}_h$  to the nonlinear primal discretization (6.3) are computed in  $\mathbf{V}_h^p$  with  $p = 1$ , i.e. the flow solutions are approximated using piecewise bilinear functions. By reducing the nonlinear residual over 6 orders of magnitude on each mesh it is ensured that the resulting flow solutions are sufficiently converged such that iterative error contributions are negligible and errors observed with respect to force coefficients are due to the discretization only. Like in e.g. [15, 18] the solutions  $\tilde{\mathbf{z}}_h$  to the linear discrete adjoint problems (3.3) and (4.7) are computed in  $\tilde{\mathbf{V}}_h = \mathbf{V}_h^{\tilde{p}}$  with  $\tilde{p} = p+1$ . Also the solutions  $\tilde{\mathbf{e}}_h$  to the discrete error equations (3.6) are computed in  $\tilde{\mathbf{V}}_h = \mathbf{V}_h^{\tilde{p}}$ .

In the following, we investigate the performance of the standard adaptive algorithm described in Section 2 and 3.1 in comparison to the proposed algorithm described in Sections 3.2 and 4.

**8.1. The standard approach.** Given  $N$  target quantities,  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , the standard approach of error estimation and goal-oriented adaptive mesh refinement consists of a multiple application of the single-target adaptive algorithm, i.e. the cycle of adaptive mesh refinement as given in Algorithm 2.1 is employed for each of the target quantities separately. This includes the solution of one discrete adjoint problem (3.3) for each of the target functionals,  $J_i(\cdot)$ ,  $i = 1, \dots, N$ , and the evaluation of the approximate error representation formulae (3.4) for  $i = 1, \dots, N$ .

We note that this amounts to solving  $N$  systems of linear equations with the same matrix but  $N$  different right-hand side vectors. Although additional adjoint solutions may possibly be obtained cheaper by using e.g. an LU factorization of the matrix we refrain from this due to the memory requirements and use an iterative solvers instead. Furthermore, a multiple application of Algorithm 2.1 leads to  $N$  separate sequences of adaptively refined meshes where each sequence is based on the same coarse grid but the subsequently refined meshes might differ from sequence to sequence. In fact, each of the  $N$  sequences of adaptively refined meshes is particularly tailored to the accurate approximation of one of the  $N$  target quantities under consideration. As a consequence each of the adjoint problems must be solved on different meshes.

TABLE 8.1

Single-target adaptive algorithm for the numerical approximation of  $J_1(\mathbf{u}) = \tilde{J}_{c_{dp}}(\mathbf{u})$ .

elem.	DoF	$J_1(\mathbf{u}) - J_1(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(1)}$	$\theta_1$
400	6400	1.040e-03	-1.404e-03	-1.35
652	10432	3.347e-03	2.959e-03	0.88
1090	17440	4.105e-04	5.712e-04	1.39
1801	28816	-2.019e-04	-1.093e-04	0.54
3034	48544	-2.284e-04	-1.893e-04	0.83
5056	80896	-1.468e-04	-1.373e-04	0.94
8515	136240	-7.400e-05	-7.141e-05	0.96
14374	229984	-3.884e-05	-3.912e-05	1.01
24265	388240	-1.678e-05	-1.698e-05	1.01

TABLE 8.2

Single-target adaptive algorithm for the numerical approximation of  $J_2(\mathbf{u}) = \tilde{J}_{c_{df}}(\mathbf{u})$ .

elem.	DoF	$J_2(\mathbf{u}) - J_2(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(2)}$	$\theta_2$
400	6400	1.075e-02	1.525e-02	1.42
655	10480	-2.976e-03	-2.592e-03	0.87
1093	17488	-1.418e-03	-1.418e-03	1.00
1804	28864	-3.977e-04	-4.325e-04	1.09
2980	47680	-9.425e-05	-1.110e-04	1.18
5101	81616	-3.930e-05	-4.344e-05	1.11
8413	134608	-2.236e-05	-2.271e-05	1.02
14041	224656	-1.601e-05	-1.631e-05	1.02
23629	378064	-1.221e-05	-1.218e-05	1.00

In Tables 8.1, 8.2, 8.3 and 8.4 we demonstrate the performance of the standard approach for the numerical approximation of the pressure induced drag, the viscous drag, the total lift and the total moment coefficient,

$$J_1(\mathbf{u}) = \tilde{J}_{c_{dp}}(\mathbf{u}), \quad J_2(\mathbf{u}) = \tilde{J}_{c_{df}}(\mathbf{u}), \quad J_3(\mathbf{u}) = \tilde{J}_{c_1}(\mathbf{u}), \quad J_4(\mathbf{u}) = \tilde{J}_{c_m}(\mathbf{u}), \quad (8.4)$$

respectively. In each case,  $i = 1, \dots, 4$ , we show the number of elements and degrees of freedom (DoF) in  $\mathbf{V}_h^1$ , the true error in the functional  $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$ , the approximate error representation formula  $\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(i)}$ , and the corresponding effectivity index  $\theta_i = \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(i)} / (J_i(\mathbf{u}) - J_i(\mathbf{u}_h))$ . We see that on all meshes but the initial coarse mesh the quality of the computed error representation formulae  $\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(i)}$  is relatively good, in the sense that  $\theta_i$  is close to one; however, as the mesh is refined, we observe that the effectivity indices  $\theta_i$  improve by slowly tending towards unity.

This confirms the behaviour of the a posteriori error estimation as already demonstrated in previous publications; cf. e.g. [15, 18]. We note however, that for each target quantity one adjoint problem needs to be solved, see the  $z_1$  components of the adjoint solutions related to the  $c_{dp}$ ,  $c_{df}$ ,  $c_1$  and  $c_m$  values in Figure 8.3.

The four different adjoint solutions account for four different sensitivities of how local residuals contribute to the error in the respective target functionals under consideration. Based on this, four different sequences of meshes are created. The resulting locally refined meshes are particularly tailored to the accurate and efficient computation of the respective target quantity under consideration, for brevity we omit more



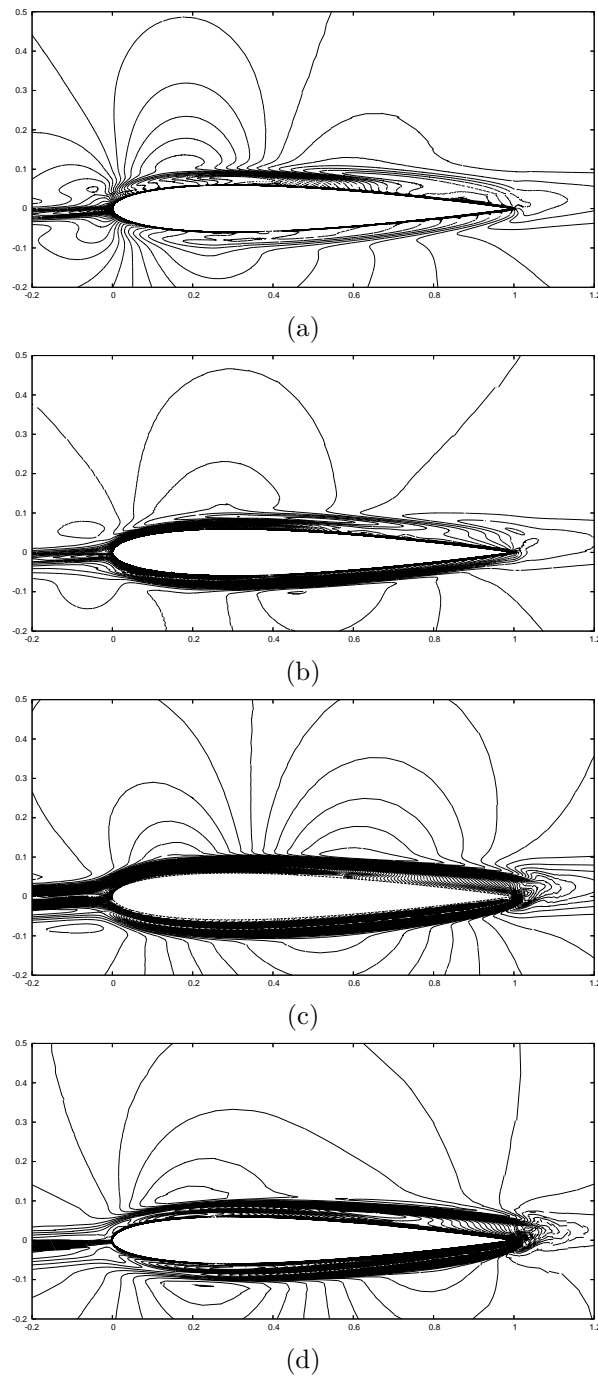


FIG. 8.3. The  $z_1$  components of the adjoint solution corresponding to the (a)  $c_{dp}$ , (b)  $c_{df}$ , (c)  $c_l$  and (d)  $c_m$  force coefficients, see Section 7.

TABLE 8.3

Single-target adaptive algorithm for the numerical approximation of  $J_3(\mathbf{u}) = \tilde{J}_{c_1}(\mathbf{u})$ .

elem.	DoF	$J_3(\mathbf{u}) - J_3(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(3)}$	$\theta_3$
400	6400	-1.173e-01	-5.869e-02	0.50
658	10528	6.730e-03	6.836e-03	1.02
1108	17728	-1.110e-03	-1.165e-03	1.05
1861	29776	-1.604e-03	-1.808e-03	1.13
3127	50032	-1.066e-03	-1.019e-03	0.96
5224	83584	-4.971e-04	-4.969e-04	1.00

TABLE 8.4

Single-target adaptive algorithm for the numerical approximation of  $J_4(\mathbf{u}) = \tilde{J}_{c_m}(\mathbf{u})$ .

elem.	DoF	$J_4(\mathbf{u}) - J_4(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa^{(4)}$	$\theta_4$
400	6400	-2.654e-03	-3.836e-03	1.45
670	10720	2.209e-03	2.055e-03	0.93
1138	18208	2.044e-04	1.647e-04	0.81
1912	30592	1.787e-05	1.910e-05	1.07
3295	52720	-1.704e-05	-1.693e-05	0.99

details and refer to similar computations in e.g. [18, 11]. The four sequences of meshes, however, amount to about four times the number of flow solutions and adjoint solutions to be computed as compared to the case of considering one target functional only. This computational overhead increases as the number of target functionals  $N$  is increased rendering this approach inefficient for sufficiently large  $N$ .

**8.2. The new approach for multiple target functionals.** In this section we now employ the approach of *a posteriori* error estimation and goal-oriented refinement for multiple target quantities as proposed in Sections 3.2 and 4. Given  $N$  target quantities this approach of error estimation does not require  $N$  adjoint solutions. Instead, as described in Section 3.2, the solutions to  $N$  adjoint problems are replaced by the solution to *one* discrete error equation. Additionally, based on a combined target functional including all original target functionals, see Section 4, only one adjoint solution is required for obtaining adjoint-based indicators to be used in goal-oriented mesh refinement. In summary, this approach allows the error estimation and adjoint-based refinement based on two auxiliary problems, namely the discrete error equation and the discrete adjoint problem, irrespective of the number of target quantities.

Here, we consider the same test case as above. Again, the goal is the accurate and efficient approximation of the pressure induced and the viscous drag, the total lift and the total moment coefficient, see (8.4), including providing error estimates for each of the computed quantities.

First we adopt the strategy of reducing the sum of the relative errors in the four target quantities, i.e. we choose the combined target functional in the adjoint problem (4.5) like in (4.4) with  $\omega_i = s_i/|J_i(\mathbf{u}_h)|$ ,  $i = 1, \dots, 4$ . This results in one sequence of adaptively refined meshes tailored to the accurate approximation of all four quantities. In Table 8.5 we collect the data of the adaptive algorithm. Here, we show the number of elements on the sequence of adaptively refined meshes. Furthermore, there are four columns, one for each of the target quantities  $c_{dp}$ ,  $c_{df}$ ,  $c_l$  and  $c_m$ . Each column is

TABLE 8.5

Multi-target adaptive algorithm for the numerical approximation of  $c_{dp}$ ,  $c_{df}$ ,  $c_l$  and  $c_m$  targeted at the reduction of the sum of relative errors. The error estimation is based on the discrete error equation (3.6) and the estimate (3.7).

elem.	error in $c_{dp}$		error in $c_{df}$		error in $c_l$		error in $c_m$	
	exact	estim.	exact	estim.	exact	estim.	exact	estim.
400	1.0e-03	-2.8e-03	1.1e-02	1.7e-02	-1.2e-01	-6.6e-02	-2.7e-03	-4.3e-03
649	1.1e-03	1.2e-03	-3.0e-03	-2.9e-03	6.0e-03	3.7e-03	2.4e-03	2.0e-03
1114	-2.7e-04	-6.7e-05	-1.4e-03	-1.9e-03	-1.1e-03	-1.1e-03	3.8e-04	3.3e-04
1879	-4.2e-04	-3.3e-04	-6.2e-04	-7.5e-04	-6.6e-04	-1.0e-03	-4.5e-05	-9.0e-05
3163	-2.0e-04	-1.7e-04	-4.6e-04	-5.2e-04	-5.4e-04	-6.4e-04	-3.0e-05	-2.7e-05
5248	-1.4e-04	-1.2e-04	-2.3e-04	-2.6e-04	-3.9e-04	-5.7e-04	-8.8e-05	-9.3e-05

subdivided into two subcolumns where the left ones include the exact errors  $J_i(u) - J_i(u_h)$ ,  $i = 1, \dots, 4$ , and the right ones include the corresponding *a posteriori* error estimates  $J'_i[\mathbf{u}_h](\tilde{\mathbf{e}}_h)$ ,  $i = 1, \dots, 4$ , see (3.7), based on the solution  $\tilde{\mathbf{e}}_h \in \tilde{\mathbf{V}}_h$  to the discrete error equation (3.6). Here, we see that on all meshes except of the coarsest mesh the estimated errors are quite close to the exact errors. In particular, the signs  $\tilde{s}_i = \text{sgn}(J'_i[\mathbf{u}_h](\tilde{\mathbf{e}}_h))$ ,  $i = 1, \dots, 4$ , of the error estimates coincide with the signs  $s_i = \text{sgn}(J_i(\mathbf{u}) - J_i(\mathbf{u}_h))$ ,  $i = 1, \dots, 4$ , of the respective exact errors. We recall that these signs are required in the definition of the combined target functional in (4.4) and are approximated by  $\tilde{s}_i$  as described in Section 4. We note, that here the difference between exact errors and error estimates are larger than the respective differences in the Tables 8.1-8.4. This is due to the fact that in Table 8.5 the estimates are based on (3.7) which includes two approximation: first the linearization of  $J_i(\cdot)$  about the discrete function  $\mathbf{u}_h$  and second the replacement of the exact error  $\mathbf{e}$  by the solution  $\tilde{\mathbf{e}}_h \in \tilde{\mathbf{V}}_h$  to the discrete error equation. In contrast, the error estimates in the Tables 8.1-8.4 include only one approximation, namely the replacement of the exact adjoint solution  $\mathbf{z}$  by the discrete adjoint solution  $\mathbf{z}_h \in \tilde{\mathbf{V}}_h$ . Nevertheless, the estimates in Table 8.5 are sufficiently close to the exact errors to serve as reasonable indication of the size of the error in each target quantity. We recall that this error estimation is based not on four (or in general  $N$ ) adjoint solutions like in Section 8.1 but based on one solution to a discrete error equation only.

After having investigated the accuracy of the *a posteriori* error estimation of the target quantities, we now concentrate on the accuracy of the evaluated target quantities achieved based on the two adaptive algorithms in this and the last subsection. Scanning through the Tables 8.1-8.4 we see that the industrial accuracy requirements (8.1) for  $c_{dp}$  (resp.  $c_{df}$ ,  $c_l$  and  $c_m$ ) are reached after 2 (resp. 3, 2 and 2) refinement steps. In contrast to that, in Table 8.5 we see that the accuracy requirements are reached after 2 (resp. 4, 2 and 2) refinement steps. We notice that there is a slight increase in the number of refinement steps for the adaptive approach based on the combined target functional in Table 8.5 in comparison to the single-target adaptive approaches applied to each of the four of the target functionals separately, see Tables 8.1-8.4. Like in single-target and multi-target optimization algorithms, this is due to the fact, that the single-target adapted meshes are optimized for the respective single target quantities whereas the multi-target adapted mesh is optimized for the combined target functional which results in a kind of compromise between the single-target adapted meshes that cannot be as accurate for the individual target functionals as the respective single-target adapted meshes.

TABLE 8.6

*Multi-target adaptive algorithm for the numerical approximation of  $c_{dp}$ ,  $c_{df}$ ,  $c_1$  and  $c_m$  targeted at the reduction of the weighted sum of absolute errors. The error estimation is based on the solution to the discrete error equation (3.6) and the estimate (3.7).*

elem.	error in $c_{dp}$		error in $c_{df}$		error in $c_1$		error in $c_m$	
	exact	estim.	exact	estim.	exact	estim.	exact	estim.
400	1.0e-03	-2.8e-03	1.1e-02	1.7e-02	-1.2e-01	-6.6e-02	-2.7e-03	-4.3e-03
652	1.4e-03	1.4e-03	-3.0e-03	-2.9e-03	6.4e-03	4.1e-03	2.4e-03	2.0e-03
1138	-2.4e-04	-5.0e-05	-1.5e-03	-1.9e-03	-1.1e-03	-1.1e-03	4.3e-04	3.8e-04
1894	-4.7e-04	-3.2e-04	-2.9e-04	-4.9e-04	-5.1e-04	-8.4e-04	-5.5e-05	-6.2e-05
3112	-4.9e-05	2.6e-05	-4.0e-04	-5.0e-04	-5.6e-05	-2.6e-04	5.5e-05	6.3e-05
5131	-1.9e-04	-1.6e-04	-8.3e-05	-1.1e-04	-8.2e-04	-9.2e-04	-2.1e-05	-1.4e-05
8539	-1.0e-04	-8.1e-05	-2.2e-05	-4.9e-05	-1.1e-04	-3.3e-04	-2.4e-05	-1.8e-05

However, the efficiency of the adaptive mesh refinement can be improved: Recalling that the accuracy requirements in (8.1) are given in terms of absolute errors where the tolerances of  $c_{dp}$ ,  $c_{df}$  and  $c_m$  are 1/10 times the tolerance of  $c_1$ , we see that choosing the combined target functional  $J_c(\cdot)$  to correspond to the weighted sum of absolute errors might be more appropriate for the problem at hand than the combined target functional corresponding to the sum of relative errors as used in Table 8.5. In fact, considering the weighted sum of absolute errors, i.e.  $J_c(\cdot)$  is given by (4.4) with  $\omega_i = \alpha_i s_i$ ,  $i = 1, \dots, 4$ , and adjusting the weighting factors

$$\alpha_1 = 1, \quad \alpha_2 = 1, \quad \alpha_3 = 0.1, \quad \alpha_4 = 1, \quad (8.5)$$

the influence of each target functional on the combined target functional corresponds to the specific accuracy requirements given in (8.1).

Analogous to the adaptive algorithm in Table 8.5 targeted at reducing the sum of relative errors, we now collect the corresponding data in Table 8.6 for the adaptive algorithm targeted at reducing the weighted sum of absolute errors. We see that the behaviour of the error estimation is similar to that described for Table 8.5. We recall that the latter two tables include the error estimates for the original force coefficients based on the solutions to the discrete error equations, see Figure 8.4. Additionally, for the combined target functional  $J_c(u_h)$  representing the weighted sum of absolute errors, we now collect the error estimates in Table 8.7. Here, we show the number of elements and degrees of freedom (DoF) in  $\mathbf{V}_h^1$ , the true error in the combined functional  $J_c(\mathbf{u}) - J_c(\mathbf{u}_h)$ , the approximate error representation formula  $\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa$ , see (4.8), and the corresponding effectivity index  $\theta_c = \sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa / (J_c(\mathbf{u}) - J_c(\mathbf{u}_h))$ . We see that on all meshes even including the initial coarse mesh the quality of the computed error representation formulae  $\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa$  is extremely good in the sense that  $\theta_c$  is close to one. We recall that these error estimates are based on the discrete adjoint solution (4.7) related to the combined target functional. Corresponding to the weighted sum (4.4) of the original target functionals the adjoint solution  $\mathbf{z}_c$ , see Figure 8.5, represents a linear combination of the adjoint solutions,  $\mathbf{z}_i$ ,  $i = 1, \dots, 4$ , (depicted in Figure 8.3) which are related to the original target functionals  $J_i(\mathbf{u})$ ,  $i = 1, \dots, 4$ .

Considering again the accuracy of the computed target quantities we see in Table 8.6 that the industrial accuracy requirements (8.1) for  $c_{dp}$  (resp.  $c_{df}$ ,  $c_1$  and  $c_m$ ) are reached after 2 (resp. 3, 2 and 2) refinement steps which is equal to the number of refinement steps required in the respective single-target adaptive algorithms in Tables

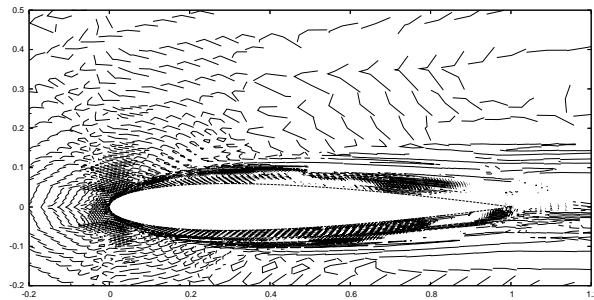


FIG. 8.4. The first component of the solution  $\tilde{\mathbf{e}}_h$  to the discrete error equation, see (3.6), on the mesh of 8539 elements, cf. Table 8.6 and Figure 8.7(b).

TABLE 8.7

Multi-target adaptive algorithm for the numerical approximation of  $c_{dp}$ ,  $c_{df}$ ,  $c_l$  and  $c_m$  targeted at the reduction of the weighted sum of absolute errors. The error estimation is based on the solution, see Figure 8.5 of the discrete adjoint problem (4.7) and the estimate (4.8).

elem.	DoF	$J_c(\mathbf{u}) - J_c(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \tilde{\eta}_\kappa$	$\theta_c$
400	6400	2.618e-02	2.636e-02	1.01
652	10432	7.378e-03	6.809e-03	0.92
1138	18208	2.258e-03	2.074e-03	0.92
1894	30304	8.582e-04	8.508e-04	0.99
3112	49792	5.087e-04	5.622e-04	1.11
5131	82096	3.753e-04	3.706e-04	0.99
8539	136624	1.622e-04	1.621e-04	1.00

8.1-8.4. However, we recall that the adaptive algorithms in Tables 8.1-8.4 include the solutions to four (or in general  $N$ ) adjoint solutions whereas the algorithm in Table 8.6 requires the solution to two auxiliary problems (the discrete error problem and the discrete adjoint problem) irrespective of the number of target functionals. This difference can also be seen in terms of computing times. In fact, the four separate single-target adaptive algorithms in Tables 8.1-8.4 add up to 147.5s to reach the industrial requirements whereas the multi-target adaptive algorithm, Table 8.6, requires 80.8s only. Note, that this difference increases when considering  $N$  target quantities for  $N > 4$ . In fact, the computing time in the single-target adaptive algorithms can be expected to increase linearly with the number  $N$  of target quantities, whereas the computing time of the multi-target adaptive algorithm requires always two auxiliary problems to be solved and is thus independent of  $N$ .

Finally, we compare the goal-oriented (adjoint-based) adaptive algorithms discussed so far with a residual-based adaptive algorithm. Here, by residual-based indicators we denote the indicators obtained from the so-called Type II error bound of the discretization error, see Corollary 7.2. The indicators  $|\eta_\kappa^{(\text{res})}|$ ,  $\kappa \in \mathcal{T}_h$ , include primal residuals but no adjoint solution. In fact, these indicators are not targeted at the exact approximation of specific target quantities but at resolving all flow features. Not depending on the adjoint solution the residual-based indicators are significantly faster to evaluate than the adjoint-based indicators. Nevertheless, as demonstrated in a sequence of earlier publications, [7, 10, 11, 15] among others, the sequences of meshes created using adjoint-based indicators are in general significantly more efficient and require much less computing resources for accurately approximating the

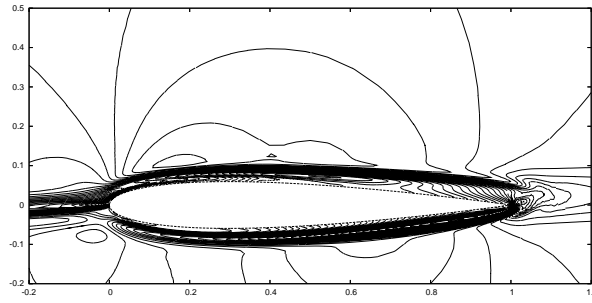


FIG. 8.5. The  $z_1$  components of the adjoint solution  $\mathbf{z}_c$  corresponding to the combined target functional  $J_c(\mathbf{u})$  which is related to the weighted sum of absolute errors.

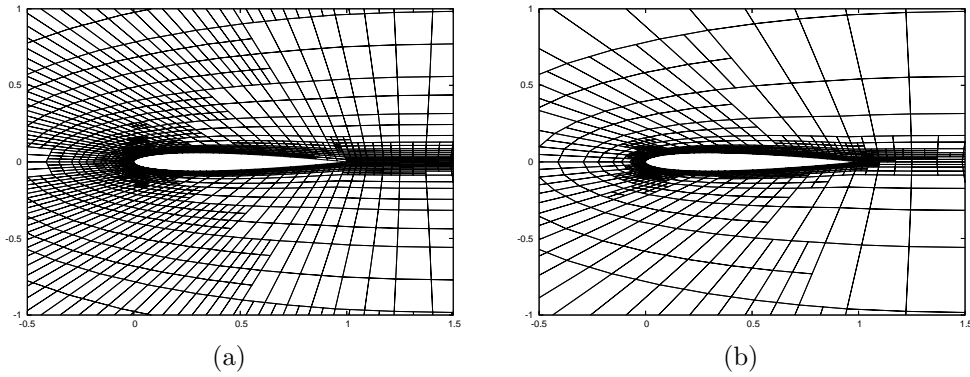


FIG. 8.6. The industrial accuracy requirements (8.1) are met on (a) the residual-based adapted mesh of 8896 elements in 149.4s and on (b) the multi-target adapted mesh of 1894 elements in 80.8s.

target quantities under consideration than the meshes created using the residual-based indicators.

A similar behaviour we observe now for the adjoint-based adaptive algorithm for multiple target functionals proposed in this article in comparison to the residual-based algorithm. In fact, whereas the multi-target adaptive algorithm meets the industrial accuracy requirements (8.1) after 3 refinement steps on 1894 elements and in 80.8s, the residual-based adaptive algorithm meets the requirements after 6 refinement steps on 8896 elements in 149.4s, see the meshes in Figure 8.6. Note, however, that in the latter case no error estimates are available. In summary, we see that, even by including the computation of error estimates in each target quantity and in the combined target functional (weighted sum of relative errors) the multi-target adaptive algorithm is significantly faster than the residual-based algorithm without providing error estimates.

This difference becomes even more significant when the stronger accuracy requirements (8.2) are imposed. Scanning through Table 8.6 we see that these requirements for  $c_{dp}$  (resp.  $c_{df}$ ,  $c_l$  and  $c_m$ ) are reached after 6 (resp. 5, 3 and 3) refinement steps. In summary, using the multi-target algorithm the academic accuracy requirements are met after 6 refinement steps on 8539 elements in 664.63s whereas using the residual-based algorithm the requirements are met after 10 refinement steps on 67660 elements in 2691.2s, see the meshes in Figure 8.7.

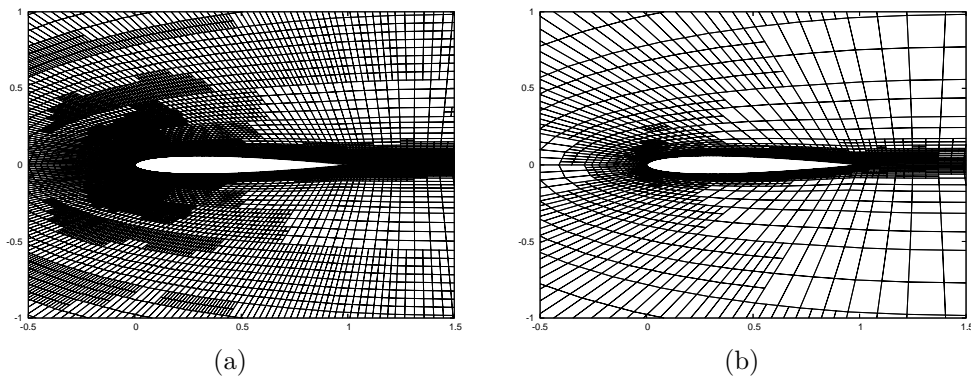


FIG. 8.7. The academic accuracy requirements (8.2) are met on (a) the residual-based adapted mesh of 67660 elements in 2691.2s and on (b) the multi-target adapted mesh of 8539 elements in 664.63s.

**9. Conclusion.** In this article we generalized the *a posteriori* error estimation and goal-oriented (adjoint-based) adaptive mesh refinement for single target quantities to multiple target quantities. Given, say  $N$  target quantities the new approach of error estimation requires not the solution to  $N$  adjoint problems as required by current approaches but the solution to *one* discrete error equation only. Additionally, we defined an adjoint problem connected to a target functional which represents a suitable combination of the original target functionals. Based on a discrete solution to this problem adjoint-based indicators are evaluated leading to an adaptive mesh refinement algorithm specifically tailored to the accurate and efficient approximation of all target quantities under consideration. This way, the *a posteriori* error estimation and goal-oriented adaptive algorithm for multiple target quantities requires not the solutions to multiple adjoint problems but the solution to two auxiliary problems only (the discrete error equation and the discrete adjoint problem) irrespective of the number of target quantities considered.

We presented numerical results applying the new adaptive algorithm to a laminar compressible flow around an airfoil with the goal to approximate several aerodynamic force coefficients up to specific accuracy tolerances defined by the industry in the European project ADIGMA. We demonstrated that based on an appropriately chosen combined target functional (corresponding to the weighted sum of absolute errors) used in the multi-target adaptive algorithm we met the prescribed accuracy requirements in the same number of local refinement steps but with significantly less computing time than current adaptive approaches which apply the single-target adaptive algorithms to each target quantity separately. This difference increases for a larger number  $N$  of target quantities as the current approach requires  $N$  adjoint solutions whereas in the proposed multi-adaptive algorithm only two auxiliary problems must be solved irrespective of the number of target quantities.

Although not being as accurate as the error estimation based on several adjoint solutions, the error estimation based on the solution to one discrete error equation still gave reasonably good error estimates with respect to the force coefficients. Additionally, the error estimation for the combined target functional representing the weighted sum of absolute errors was extremely good. Finally, the efficiency of the proposed adaptive algorithm has been compared with residual-based mesh refinement which does not require the solution of auxiliary problems but does not provide error

estimates either. We showed that the industrial as well as stronger academic accuracy requirements on the force coefficients are met after significantly less refinement steps and computing time based on the multi-target algorithm than based on the residual-based refinement algorithm.

**Acknowledgments.** The author acknowledges the partial financial support of both the President's Initiative and Networking Fund of the Helmholtz Association of German Research Centres and the European project ADIGMA [1]. Computations have been performed using the DG flow solver PADGE [13] based on the deal.II library [3, 4]. Finally, the author likes to thank the Aircraft Research Association for providing the initial coarse mesh employed in the numerical results section.

## REFERENCES

- [1] ADIGMA. Adaptive higher-order variational methods for aerodynamic applications in industry. A specific targeted research project of the sixth European community framework program. See <http://www.dlr.de/as/desktopdefault.aspx/tabid-2035>.
- [2] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
- [3] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – A general purpose object oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4), Aug. 2007. Available also as technical report ISC-06-02-MATH, Texas A&M University, 2006.
- [4] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II *Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org/>, 6.0 edition, Sept 2007. First edition 1999.
- [5] T. Barth and M. Larson. A-posteriori error estimation for higher order godunov finite volume methods on unstructured meshes. In R. Herbin and D. Kröner, editors, *Finite Volumes for Complex Applications III*, pages 41–63. Hermes Science Pub., 2002. (\*).
- [6] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.
- [7] R. Becker and R. Rannacher. An optimal control approach to error estimation and mesh adaptation in finite element methods. *Acta Numerica*, 10:1–102, 2001.
- [8] M. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.
- [9] K. Harriman, D. Gavaghan, and E. Süli. The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method. Technical report, Oxford University Computing Laboratory, 2004.
- [10] R. Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations*. PhD thesis, University of Heidelberg, 2002.
- [11] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations. *Int. J. Numer. Meth. Fluids*, 51(9–10):1131–1156, 2006.
- [12] R. Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6):2671–2696, 2007.
- [13] R. Hartmann, J. Held, T. Leicht, and F. Prill. *PADGE, Parallel Adaptive Discontinuous Galerkin Environment, Technical reference*. DLR, Braunschweig, 2008. In preparation.
- [14] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 24:979–1004, 2002.
- [15] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183:508–532, 2002.
- [16] R. Hartmann and P. Houston. Goal-oriented a posteriori error estimation for multiple target functionals. In T. Y. Hou and E. Tadmor, editors, *Hyperbolic problems: theory, numerics, applications*, pages 579–588. Springer, 2003.
- [17] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: Method formulation. *Int. J. Num. Anal. Model.*, 3(1):1–20, 2006.
- [18] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal-oriented a posteriori error estimation. *Int. J. Num. Anal. Model.*, 3(2):141–162, 2006.
- [19] R. Hartmann and P. Houston. An optimal order interior penalty discontinuous Galerkin dis-



- cretization of the compressible Navier–Stokes equations. *J. Comput. Physics*, 2007. Submitted.
- [20] P. Houston, R. Rannacher, and E. Süli. A posteriori error analysis for stabilised finite element approximations of transport problems. *Comput. Meth. Appl. Mech. Engrg.*, 190(11-12):1483–1508, 2000.
  - [21] P. Houston and E. Süli. *hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM J. Sci. Comp.*, 23(4):1225–1251, 2001.
  - [22] P. Houston and E. Süli. Stabilized *hp*-finite element approximation of partial differential equations with nonnegative characteristic form. *Computing*, 66(2):99–119, 2001.
  - [23] J. Lu. *An a posteriori Error Control Framework for Adaptive Precision Optimization using Discontinuous Galerkin Finite Element Method*. PhD thesis, M.I.T., 2005.
  - [24] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comp. Phys.*, 187:22–46, 2003.