

A Comparison of Hybridized and Standard DG Methods for Target-Based hp-Adaptive Simulation of Compressible Flow

Michael Woopen^a, Aravind Balan^a,
Georg May^a and Jochen Schütz^b

Bericht Nr. 372

August 2013

Key words: discontinuous Galerkin methods, hybridization,
hp-adaptation, adjoint-based error-estimation,
compressible flow

AMS Subject Classifications: 65M60, 35M33, 65M15

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Templergraben 55, D-52056 Aachen (Germany)

^a Institute for Advanced Study in Computational Engineering Science, RWTH Aachen University, Aachen, Germany
Emails: {woopen, balan, may}@aices.rwth-aachen.de

^b Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Aachen, Germany
Email: schütz@igpm.rwth-aachen.de

Preprint submitted to Computers & Fluids, August 19, 2013

A Comparison of Hybridized and Standard DG Methods for Target-Based hp-Adaptive Simulation of Compressible Flow

Michael Woopen^a, Aravind Balan^a, Georg May^a, Jochen Schütz^b

^a*Aachen Institute for Advanced Study in Computational Engineering Science, RWTH Aachen University, Aachen, Germany*

^b*Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Aachen, Germany*

Abstract

We present a comparison between hybridized and non-hybridized discontinuous Galerkin methods in the context of target-based hp-adaptation. Using a discrete-adjoint approach, sensitivities with respect to output functionals are computed to drive the adaptation. From the error distribution given by the adjoint-based error estimator, h- or p-refinement is chosen based on the smoothness of the solution which can be quantified by properly-chosen smoothness indicators. Numerical results are shown for inviscid subsonic and transonic, and laminar viscous flow around the NACA0012 airfoil.

Keywords: discontinuous Galerkin methods, hybridization, hp-adaptation, adjoint-based error-estimation, compressible flow

Email addresses: woopen@aices.rwth-aachen.de (Michael Woopen), balan@aices.rwth-aachen.de (Aravind Balan), may@aices.rwth-aachen.de (Georg May), schuetz@igpm.rwth-aachen.de (Jochen Schütz)

1. Introduction

During the last years, discontinuous Galerkin (DG) methods (see, e.g., [5, 9, 1]) have become increasingly popular. This is indisputably due to their advantages — high-order accuracy on unstructured meshes, a variational setting, and local conservation, just to name a few.

However, the use of discontinuous function spaces is at the same time the reason for a major disadvantage: unlike in continuous Galerkin (CG) methods, degrees of freedom are not shared between elements. As a consequence, the number of unknowns is substantially higher compared to a CG discretization. Especially for implicit time discretization this imposes large memory requirements, and potentially leads to increased time-to-solution.

In order to avoid these disadvantages, a technique called hybridization may be utilized (see [11, 8, 10, 19, 18, 20]), resulting in hybridized discontinuous Galerkin (HDG) methods. Here, the globally coupled unknowns have support on the mesh skeleton, i.e. the element interfaces, only. This reduces the size of the global system and coincidentally improves the sparsity pattern.

However, aiming at industry applications, e.g. turbulent flow around a complete airplane or within an aircraft engine, hybridization alone does most likely not provide a sufficiently successful overall algorithm. In these applications one is usually interested in certain quantities only, for example lift or drag coefficients in aerospace, instead of the solution quality per se. Thus, it might be beneficial to distribute the degrees of freedom within the computational domain in such a way that the solution to the discretized problem is close to optimal with respect to the accuracy of these quantities. To achieve this goal, target-based error control methods have been developed

(see [15, 24, 25, 13, 12]). One such method is based on the adjoint solution of the original governing equations with respect to the target functional. In this method, an additional linear system of partial differential equations is solved which then gives an estimate on the spatial error distribution contributing to the error in the target functional. This estimate can be used as a criterion for local adaptation. Within the context of low order schemes, mesh refinement is used for adaptation [24, 25]. Using DG (or HDG), however, offers the additional possibility of varying the polynomial degree within each element. For smooth solutions, this is more efficient compared to mesh refinement, as it yields exponential convergence. Combining both methodologies results in so-called hp-adaptation.

In [23], we presented a discretization method for nonlinear convection-diffusion equations. The method is based on a discontinuous Galerkin discretization for convection terms, and a mixed method using $H(\text{div})$ spaces for the diffusive terms. Furthermore, hybridization is used to reduce the number of globally coupled degrees of freedom. Its adjoint consistency was shown in [22]. In [27, 2], we extended our computational framework to include HDG schemes, as well as adjoint-based h- and hp-adaptation. In the current paper, we compare our HDG method with a standard DG method in the context of hp-adaptation for stationary compressible flow, mainly with the aim to assess the efficiency of both methods.

This paper is structured as follows. We briefly cover the governing equations, namely the compressible Euler and Navier-Stokes equations, in Sec. 2. After that we introduce our discretization and describe the concept of hybridization in Sec. 3. In Sec. 4 we establish the adjoint formulation and show

how hybridization can be applied to the dual problem. Then we show its efficiency and robustness with examples from compressible flow, including inviscid subsonic and transonic, and subsonic laminar flow, in Sec. 5. Finally, we offer conclusions and outlook on future work in Sec. 6.

2. Governing Equations

We consider systems of partial differential equations

$$\nabla \cdot (f_c(w) - f_v(w, \nabla w)) = s(w, \nabla w) \quad (1)$$

with convective and diffusive fluxes

$$f_c : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times d} \quad \text{and} \quad f_v : \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^{m \times d}, \quad (2)$$

respectively, and a state-dependent source term

$$s : \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^m. \quad (3)$$

(Potentially, some of these quantities could be zero.) We denote the spatial dimension by d and the number of conservative variables by m . Boundary conditions can be applied to the conservative variables $w \in \mathbb{R}^m$ and to the diffusive flux f_v .

2.1. Two-Dimensional Euler Equations

The Euler equations are comprised of the inviscid compressible continuity, momentum and energy equations. They are given in conservative form as

$$\nabla \cdot f_c(w) = 0 \quad (4)$$

with the vector of conserved variables

$$w = (\rho, \rho u, \rho v, E)^T \quad (5)$$

where ρ is the density, u and v are the components of the velocity vector $\hat{w} := (u, v)^T$, and E the total energy. The convective flux is given by

$$f_{c,1} = (\rho u, p + \rho u^2, \rho uv, u(E + p))^T \quad (6)$$

$$f_{c,2} = (\rho v, \rho uv, p + \rho v^2, v(E + p))^T. \quad (7)$$

Pressure is related to the conservative flow variables w by the equation of state

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2) \right) \quad (8)$$

where $\gamma = c_p/c_v$ is the ratio of specific heats, generally taken as 1.4 for air.

Along wall boundaries we apply the slip boundary condition

$$U(w) = (u, v) \cdot n = 0. \quad (9)$$

We also define a boundary function which satisfies $U(w_{\partial\Omega}(w)) = 0$ as

$$w_{\partial\Omega}(w) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 - n_x^2 & -n_x n_y & 0 \\ 0 & -n_x n_y & 1 - n_y^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} w. \quad (10)$$

2.2. Two-Dimensional Navier-Stokes Equations

The Navier-Stokes equations in conservative form are given by

$$\nabla \cdot (f_c(w) - f_v(w, \nabla w)) = 0. \quad (11)$$

The convective part f_c of the Navier-Stokes equations coincides with the Euler equations. The viscous flux is given by

$$f_{v,1} = (0, \tau_{11}, \tau_{21}, \tau_{11}u + \tau_{12}v + kT_x)^T \quad (12)$$

$$f_{v,2} = (0, \tau_{12}, \tau_{22}, \tau_{21}u + \tau_{22}v + kT_y)^T. \quad (13)$$

The temperature is defined via the ideal gas law

$$T = \frac{\mu\gamma}{k \cdot \text{Pr}} \left(\frac{E}{\rho} - \frac{1}{2} (u^2 + v^2) \right) = \frac{1}{(\gamma - 1)c_v} \frac{p}{\rho} \quad (14)$$

where $\text{Pr} = \frac{\mu c_p}{k}$ is the Prandtl number, which for air at moderate conditions can be taken as a constant with a value of $\text{Pr} \approx 0.72$. k denotes the thermal conductivity coefficient. For a Newtonian fluid, the stress tensor is defined as

$$\tau = \mu \left(\nabla \hat{w} + (\nabla \hat{w})^T - \frac{2}{3} (\nabla \cdot \hat{w}) \text{Id} \right). \quad (15)$$

The variation of the molecular viscosity μ as a function of temperature is determined by Sutherland's law as

$$\mu = \frac{C_1 T^{3/2}}{T + C_2} \quad (16)$$

with $C_1 = 1.458 \times 10^{-6} \frac{\text{kg}}{\text{ms}\sqrt{\text{K}}}$ and $C_2 = 110.4 \text{ K}$.

Along wall boundaries, we apply the no-slip boundary condition, i.e.

$$(u, v) = 0. \quad (17)$$

with corresponding boundary function

$$w_{\partial\Omega}(w) = (\rho, 0, 0, E)^T \quad (18)$$

Furthermore, one has to give boundary conditions for the temperature. In the present work we use the adiabatic wall condition, i.e.

$$\nabla T \cdot n = 0 \quad (19)$$

Combining both no-slip and adiabatic wall boundary conditions, gives a condition for the viscous flux, namely

$$f_{v,\partial\Omega}(f_v) = \begin{pmatrix} 0 & \tau_{11} & \tau_{21} & 0 \\ 0 & \tau_{12} & \tau_{22} & 0 \end{pmatrix}^T. \quad (20)$$

3. Discretization

3.1. Notation

We tessellate the domain Ω into a collection of non-overlapping elements, denoted by \mathcal{T}_h , such that $\bigcup_{K \in \mathcal{T}_h} K = \Omega$. For the element edges we consider two different kinds of sets, $\partial\mathcal{T}_h$ and Γ_h , which are element-oriented and edge-oriented, respectively. The first is the collection of all element boundaries, which means that every edge appears twice. The latter, however, includes every edge just once. The reason for this distinction will become clear later. Please note that neither of these sets shall include edges lying on the domain boundary; the set of boundary edges is denoted by Γ_h^b .

We denote by $\Pi^p(D)$ the set of polynomials of degree at most p on some domain D . We will need discontinuous function spaces for the domain and the mesh skeleton:

$$V_h = \{v \in L^2(\Omega) : v|_K \in \Pi^{p_K}(K), \quad K \in \mathcal{T}_h\}^{m \times d} \quad (21)$$

$$W_h = \{w \in L^2(\Omega) : w|_K \in \Pi^{p_K}(K), \quad K \in \mathcal{T}_h\}^m \quad (22)$$

$$M_h = \{\mu \in L^2(\Gamma_h) : \mu|_e \in \Pi^{p_e}(e), \quad e \in \Gamma_h\}^m. \quad (23)$$

Thus, $v \in V_h$, $w \in W_h$ and $\mu \in M_h$ are piecewise polynomials of degree p which can be discontinuous across edges (for v , w) or vertices (for μ), respectively.

Usually, the polynomial degree between elements and interfaces does not vary. If, however, two neighboring elements have different polynomial degrees p_{K^-} and p_{K^+} , the polynomial degree on the interface shared by both elements is taken to be the higher one, i.e. $p_e = \max\{p_{K^-}, p_{K^+}\}$. This way, both optimal order of consistency in each element and stability can be ensured.

We will distinguish between element-oriented inner products and edge-oriented inner products

$$(v, w)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \int_K vw \, dx,$$

$$\langle v, w \rangle_{\partial \mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \int_{\partial K} vw \, d\sigma, \quad \langle v, w \rangle_{\Gamma_h} = \sum_{e \in \Gamma_h} \int_e vw \, d\sigma.$$

3.2. Weak Formulation

We can rewrite general convection-diffusion equations as a first-order system by introducing an additional unknown representing the gradient of the solution

$$q = \nabla w$$

$$\nabla \cdot (f_c(w) - f_v(w, q)) = s(w, q).$$

By multiplying the strong, mixed form with appropriate test functions $(\tau_h, \varphi_h) \in V_h \times W_h$ and integrating by parts, we obtain a standard DG discretization in mixed form

$$0 = (\tau_h, q_h)_{\mathcal{T}_h} + (\nabla \cdot \tau_h, w_h)_{\mathcal{T}_h} - \langle \tau_h \cdot n, \widehat{w} \rangle_{\partial \mathcal{T}_h}$$

$$- (\nabla \varphi_h, f_c(w_h) - f_v(w_h, q_h))_{\mathcal{T}_h} - (\varphi_h, s(w_h, q_h))_{\mathcal{T}_h} + \langle \varphi_h, \widehat{f}_c - \widehat{f}_v \rangle_{\partial \mathcal{T}_h}$$

$$+ \mathcal{N}_{h, \partial \Omega}^{\text{DG}}(q_h, w_h; \tau_h, \varphi_h)$$

where the numerical trace \widehat{w} and the numerical fluxes $\widehat{f}_c, \widehat{f}_v$ have to be chosen appropriately to define a stable and consistent method. Furthermore, the boundary conditions, here denoted by $\mathcal{N}_{h,\partial\Omega}^{\text{DG}}(q_h, w_h; \tau_h, \varphi_h)$, have to be discretized appropriately.

In contrast to a DG discretization, where the numerical trace \widehat{w} is defined explicitly in terms of w_h and q_h , it is treated as an additional unknown in an HDG method. This additional unknown is called λ_h and has support on the skeleton of the mesh only. In order to close the system the continuity of the numerical fluxes across edges is required in a weak sense, resulting in a third equation.

The weak formulation of the hybrid system, comprised of equations for the gradient q_h , the solution itself w_h and its trace on the mesh skeleton λ_h , is then given by:

$$\text{Find } (q_h, w_h, \lambda_h) \in X_h := (V_h, W_h, M_h) \text{ s.t. } \forall (\tau_h, \varphi_h, \mu_h) \in X_h$$

$$\begin{aligned} 0 &= \mathcal{N}_h(q_h, w_h, \lambda_h; \tau_h, \varphi_h, \mu_h) \\ &:= (\tau_h, q_h)_{\mathcal{T}_h} + (\nabla \cdot \tau_h, w_h)_{\mathcal{T}_h} - \langle \tau_h \cdot n, \lambda_h \rangle_{\partial\mathcal{T}_h} \\ &\quad - (\nabla \varphi_h, f_c(w_h) - f_v(w_h, q_h))_{\mathcal{T}_h} - (\varphi_h, s(w_h, q_h))_{\mathcal{T}_h} + \langle \varphi_h, \widehat{f}_c - \widehat{f}_v \rangle_{\partial\mathcal{T}_h} \\ &\quad + \left\langle \mu_h, \left[\widehat{f}_c - \widehat{f}_v \right] \right\rangle_{\Gamma_h} + \mathcal{N}_{h,\partial\Omega}(q_h, w_h; \tau_h, \varphi_h). \end{aligned}$$

Please note the use of $\partial\mathcal{T}_h$ in the weak formulation of the mixed form, and Γ_h in the last equation defining λ_h . This perfectly resembles the character of these equations, being element- and edge-oriented, respectively. The terms tested against τ_h and φ_h are called local solvers, meaning they do not depend on the solution within neighboring elements but only on the trace of the solution which is approximated by λ_h . The coupling between elements is

then introduced by weakly enforcing the normal continuity of the numerical fluxes across interfaces.

We choose numerical fluxes comparable to the Lax-Friedrich flux and to the LDG flux for the convective and diffusive flux, respectively, i.e.

$$\widehat{f}_c(\lambda_h, w_h) = f_c(\lambda_h) \cdot n - \alpha_c(\lambda_h - w_h) \quad (24)$$

$$\widehat{f}_v(\lambda_h, w_h, q_h) = f_v(\lambda_h, q_h) \cdot n + \alpha_v(\lambda_h - w_h) \quad (25)$$

The stabilization introduced can be given by a tensor; in our work, however, we restrict ourselves to a constant scalar $\alpha = \alpha_c + \alpha_v$ which seems to be sufficient for a wide range of test cases.

3.2.1. Boundary Conditions

In order to retrieve an adjoint-consistent scheme, special care has to be taken when discretizing the boundary conditions (see Schütz and May [22]). The boundary conditions have to be incorporated by using the boundary states $w_{\partial\Omega}$ and boundary fluxes $f_{v,\partial\Omega}$, i.e.

$$\begin{aligned} \mathcal{N}_{h,\partial\Omega}(q_h, w_h; \tau_h, \varphi_h) := & \langle \tau_h \cdot n, w_{\partial\Omega}(w_h) \rangle_{\Gamma_h^b} \\ & + \langle \varphi_h, (f_c(w_{\partial\Omega}(w_h)) - f_{v,\partial\Omega}(f_v(w_{\partial\Omega}(w_h), q_h))) \cdot n \rangle_{\Gamma_h^b}. \end{aligned}$$

We would like to emphasize that λ_h does not occur in this boundary term as it is only defined on interior edges.

3.2.2. Shock-Capturing

In non-smooth parts of the solution, for example shocks in compressible flows, a stabilization term has to be introduced. We use the shock-capturing approach by Nguyen and Peraire [17] which is based on an artificial viscosity

term incorporating the local dilatation of the flow. It is worth mentioning, that we discretize the shock-capturing term, given by $\nabla \cdot (\epsilon(w, \nabla w) \nabla w)$, only with the volume integral contribution of the weak formulation (thus neglecting the surface integral due to the integration by parts), i.e. we augment the discretization with the term

$$\mathcal{N}_{h,sc}(w_h; \varphi_h) := -(\nabla \varphi_h, \epsilon(w_h, \nabla w_h) \nabla w_h)_{\mathcal{T}_h}. \quad (26)$$

In the viscous case, where the gradient is explicitly given, ∇w_h can be replaced by q_h yielding

$$\mathcal{N}_{h,sc}(q_h, w_h; \varphi_h) := -(\nabla \varphi_h, \epsilon(w_h, q_h) q_h)_{\mathcal{T}_h}. \quad (27)$$

Please note, that this term enters only the local part of the discretization.

3.3. Relaxation

In order to solve the nonlinear system of equations that defines the HDG method, the Newton-Raphson method is applied. Beginning with an initial guess $\mathbf{x}_h^0 := (q_h^0, w_h^0, \lambda_h^0)$, one iteratively solves the linear system

$$\mathcal{N}'_h[\mathbf{x}_h^n](\delta \mathbf{x}_h^n; \mathbf{y}_h) = -\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h) \quad \forall \mathbf{y}_h \in X_h \quad (28)$$

and updates the solution as

$$\mathbf{x}_h^{n+1} = \mathbf{x}_h^n + \delta \mathbf{x}_h^n \quad (29)$$

until the residual $\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h)$ has reached a certain threshold. Please note, that we have grouped q_h , w_h and λ_h , and the test functions into $\mathbf{x}_h := (q_h, w_h, \lambda_h)$ and $\mathbf{y}_h := (\tau_h, \varphi_h, \mu_h)$, respectively. \mathcal{N}'_h denotes the Fréchet derivative of \mathcal{N}_h with respect to \mathbf{x}_h^n .

This routine can, however, lead to stability problems if the starting value \mathbf{x}_h^0 is too far away from the solution \mathbf{x}_h . Therefore, an artificial time is introduced and a backward Euler method is applied, which yields a slight modification of the linear system given in Eq. (28), namely

$$\left(\varphi_h, \frac{1}{\Delta t^n} \delta w_h^n \right)_{\mathcal{T}_h} + \mathcal{N}'_h[\mathbf{x}_h^n](\delta \mathbf{x}_h^n; \mathbf{y}_h) = -\mathcal{N}_h(\mathbf{x}_h^n; \mathbf{y}_h). \quad (30)$$

Please note that by choosing $\Delta t^n \rightarrow \infty$, a pure Newton-Raphson method is obtained. Usually the time step is kept finite for a few initial steps to ensure stability. As soon as the residual is lower than a certain threshold, i.e. the current approximation \mathbf{x}_h^n is thought to be sufficiently close to the solution \mathbf{x}_h , we let the time step go towards infinity.

As we are interested in steady-state problems, we can use local time-stepping in order to accelerate the computation. For each element, we apply a time step based on a global CFL number, the element volume and an approximation of the flux Jacobian's spectral radius, i.e.

$$\Delta t_K^n = \text{CFL}^n \frac{|K|}{\lambda_c + 4\lambda_v}. \quad (31)$$

Here λ_c and λ_v represent approximations to the maximum eigenvalue of the convective and diffusive flux, respectively (see Mavriplis and Jameson [16]).

3.4. Hybridization

Using an appropriate polynomial expansion for δq_h , δw_h and $\delta \lambda_h$, the linearized global system is given in matrix form as

$$\begin{bmatrix} A & B & R \\ C & D & S \\ L & M & N \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \\ \delta \Lambda \end{bmatrix} = \begin{bmatrix} F \\ G \\ H \end{bmatrix} \quad (32)$$

where the vector $[\delta Q, \delta W, \delta \Lambda]^T$ contains the expansion coefficients of $\delta \mathbf{x}_h$ with respect to the chosen basis.

In order to carry on with the derivation of the hybridized method, we want to formulate that system in terms of $\delta \Lambda$ only. Therefore we split it into

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix} - \begin{bmatrix} R \\ S \end{bmatrix} \delta \Lambda \quad (33)$$

and

$$\begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} \delta Q \\ \delta W \end{bmatrix} + N \delta \Lambda = H. \quad (34)$$

Substituting Eq. (33) into Eq. (34) yields the hybridized system

$$\left(N - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix} \right) \delta \Lambda = H - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} F \\ G \end{bmatrix} \quad (35)$$

The workflow is as follows: First, the hybridized system is assembled and then being solved for $\delta \Lambda$. Subsequently, δQ and δW can be reconstructed inside the elements via Eq. (33). It is very important to note that it is not necessary to solve the large system given by Eq. (33). In fact, the matrix in Eq. (33) can be reordered to be block diagonal. Each of these blocks is associated to one element. Thus, both the assembly of the hybridized matrix in Eq. (35) and the reconstruction of δQ and δW can be done in an element-wise fashion. In order to save computational time, the solutions to Eq. (33) can be saved after the assembly of the hybridized system and reused during the reconstruction of δQ and δW .

The hybridized matrix is a $n_f \times n_f$ block matrix, where $n_f = |\Gamma_h|$ is the number of interior edges. In each block row there is one block on the

diagonal and $2d$ off-diagonal blocks in the case of simplex elements. These blocks represent the edges of the neighboring elements of one edge. Each block is dense and has $\mathcal{O}(m^2 \cdot p^{2(d-1)})$ entries. Please recall that p is the polynomial degree of the ansatz functions, d is the spatial dimension of the domain Ω and m is the number of partial differential equations ($m = 4$ for the 2-dimensional Euler or Navier-Stokes equations). This structure is very similar to that of a normal DG discretization, whereas the blocks in the latter have $\mathcal{O}(m^2 \cdot p^{2d})$ entries and thus are considerably bigger for higher polynomial order p . The size of the system matrix does not only play a big role in terms of memory consumption but also for the iterative solver. Here, a major portion of the overall workload goes into matrix-vector products which are of course faster, if the problem dimensions are smaller. In our code we use an ILU(n)-preconditioned GMRES which is available through the PETSc library [4, 3].

4. Adaptation Procedure

In the context of adjoint-based (also referred to as target- or output-based) error estimation, one is interested in quantifying the error of a specific target functional $J_h : X_h \rightarrow \mathbb{R}$, i.e.

$$e_h := J_h(\mathbf{x}) - J_h(\mathbf{x}_h), \quad (36)$$

where \mathbf{x}_h is the approximation to \mathbf{x} in X_h . This target functional can for example represent lift or drag coefficients in aerospace applications. In general, the target functional is an integrated value, where integration can be both on a volume or along the boundary. For the derivation of the adjoint-based

error estimate we expand the target functional in a Taylor series as follows

$$J_h(\mathbf{x}) - J_h(\mathbf{x}_h) = J'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_h\|^2). \quad (37)$$

We proceed in a similar manner with the error in the residual, i.e.

$$\mathcal{N}_h(\mathbf{x}; \mathbf{y}_h) - \mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) = \mathcal{N}'_h[\mathbf{x}_h](\mathbf{x} - \mathbf{x}_h; \mathbf{y}_h) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_h\|^2). \quad (38)$$

As our discretization is consistent the first term $\mathcal{N}'_h(\mathbf{x}; \mathbf{y}_h)$ vanishes.

Substituting Eq. (38) into Eq. (37) and neglecting the quadratic terms yields

$$e_h \approx \eta := -\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h) \quad (39)$$

where \mathbf{z}_h is defined by the so-called adjoint equation

$$\mathcal{N}'_h[\mathbf{x}_h](\mathbf{y}_h; \mathbf{z}_h) = J'_h[\mathbf{x}_h](\mathbf{y}_h) \quad \forall \mathbf{y}_h \in \tilde{X}_h. \quad (40)$$

The adjoint solution $\mathbf{z}_h = (\tilde{q}_h, \tilde{w}_h, \tilde{\lambda}_h) \in \tilde{X}_h$ represents the link between variations in the residual and in the target functional.

The global error estimate η can then be restricted to a single element to yield a local indicator to drive an adaptation procedure, i.e.

$$\eta_K := |\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h)|_K \quad (41)$$

so that $|\eta| \leq \sum_{K \in \mathcal{T}_h} \eta_K$ holds.

Please note, that the functionals \mathcal{N}_h and J_h and their jacobians have to be evaluated in a somewhat richer space than X_h , namely $\tilde{X}_h \supset X_h$. Otherwise the weighted residual $\mathcal{N}_h(\mathbf{x}_h; \mathbf{z}_h)$ would be identical zero as

$$\mathcal{N}_h(\mathbf{x}_h; \mathbf{y}_h) = 0 \quad \forall \mathbf{y}_h \in X_h. \quad (42)$$

This can be achieved by either mesh refinement or a higher polynomial degree of the ansatz functions. In our setting, especially when using a hierarchical basis, the latter is advantageous with respect to implementational effort and efficiency.

4.1. Hybridization

In matrix form, the adjoint system (see Eq. (40)) reads as follows

$$\begin{bmatrix} A & B & R \\ C & D & S \\ L & M & N \end{bmatrix}^T \begin{bmatrix} \tilde{Q} \\ \tilde{W} \\ \tilde{\Lambda} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ \tilde{G} \\ \tilde{H} \end{bmatrix} \quad (43)$$

Please note, that in our formulation $\tilde{H} = 0$ as λ_h is not defined on the boundary and thus the target functional depends only on w_h and q_h .

As the overall structure of the adjoint equation is similar to the primal system (see Eq. (35)), one can also apply static condensation to the adjoint system which then yields its hybridized form:

$$\left(N - \begin{bmatrix} L & M \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix} \right)^T \tilde{\Lambda} = - \begin{bmatrix} R^T & S^T \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-T} \begin{bmatrix} \tilde{F} \\ \tilde{G} \end{bmatrix}$$

It is interesting to note that the hybridized adjoint system matrix is also the transpose of the hybridized primal system matrix. This is very beneficial for the implementation as the routines for the assembly of this matrix are already available.

The adjoint solution within each element can then be computed with the aid of the adjoint local system, given by

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^T \begin{bmatrix} \tilde{Q} \\ \tilde{W} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ \tilde{G} \end{bmatrix} - \begin{bmatrix} L & M \end{bmatrix}^T \tilde{\Lambda} \quad (44)$$

where the matrix is also the transpose of the primal local matrix (see Eq. (33)).

4.2. Marking Elements for Refinement

After having obtained a localized error estimate, we have to choose a set of elements to be refined. This can be done in many different ways. Often the so-called fixed-fraction approach is chosen [12]. Here, a user-defined fraction of elements which contribute the most to the overall error are marked. The only parameter necessary is $\theta \in (0, 1)$ which determines the size of this set relative to the total number of elements ($\theta = 1$ corresponds to uniform refinement and $\theta = 0$ to no refinement at all).

4.3. Choosing between h - and p -Adaptation

The final step in the adaptation procedure is the decision between mesh refinement and order enrichment. There exist several ways to make this decision. Ceze and Fidkowski [7] solve local adjoint problems for both options and then decide which one is more efficient with respect to degrees of freedom or the non-zero entries in the system matrix. We, however, adopt the strategy by Wang and Mavriplis [26]. They used a smoothness sensor devised by Persson and Peraire [21] for an artificial viscosity approach. This sensor exploits the fact that the decay of the expansion coefficients of the solution is closely linked to its smoothness. For smoother solutions, decay is faster. This can be exploited to check the regularity of the solution. On each elements, the smoothness sensor is defined as

$$S_K := \frac{(w - \tilde{w}, w - \tilde{w})_K}{(w, w)_K} \quad (45)$$

where \tilde{w} is the projection of w to the next smaller polynomial space. Hence, $w - \tilde{w}$ represents the higher order components of the solution (see Fig. 1).

As we use an orthogonal basis this projection is very cheap. First promising results for this approach are given in [2].

5. Results

In the following we compare our in-house HDG and DG solvers in terms of degrees of freedom and runtime. The DG discretization is based on the Lax-Friedrich and the BR2 [6] fluxes for convective and viscous terms, respectively. Boundary conditions and target functionals are evaluated in an adjoint-consistent manner [14]. Both solvers share the same computational framework, so we believe that our comparison is meaningful.

We apply both solvers to compressible flow problems, including inviscid subsonic and transonic, and subsonic laminar flow. In all cases we show results for pure mesh-adaptation ($p = 1 \dots 4$) and hp-adaptation ($p = 2 \dots 5$).

However, before we turn our attention to the adaptive computations, we want to compare runtimes for both methods on a fixed mesh for several polynomial orders. This way, we can a priori learn which improvement can be expected. In Fig. 2 runtimes for both HDG and DG can be seen for a Euler and a Navier-Stokes computation on a mesh with 2560 elements. We used polynomial orders from $p = 0$ to $p = 6$. Usually, there are more faces than elements. Hence, for $p = 0$ and $p = 1$ DG is faster than HDG. However, already for $p = 2$ HDG catches up. At $p = 6$ there is a ratio of 2 for the Euler test case and 1.8 for the Navier-Stokes test case.

5.1. Subsonic Inviscid Flow over the NACA 0012 Airfoil

In the first test case, we consider subsonic inviscid flow over the NACA 0012 airfoil which is defined by

$$y = \pm 0.6 (0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (46)$$

with $x \in [0, 1]$. Using this definition, the airfoil would have a finite trailing edge thickness of .252 %. In order to obtain a sharp trailing edge we modify the x^4 coefficient, i.e.

$$y = \pm 0.6 (0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4). \quad (47)$$

The flow is characterized by a free stream Mach number of $\text{Ma}_\infty = 0.5$ and an angle of attack of $\alpha = 2^\circ$. In Fig. 3 the baseline mesh for the Euler test cases (subsonic and transonic) can be seen. It consists of 719 elements and its far field is over a 1000 chords away.

Admissible target functionals defined on the boundary for the Euler equations are given by the weighted pressure along wall boundaries, i.e.

$$J(w) = \int_{\partial\Omega} \psi \cdot (pn) \, d\sigma \quad (48)$$

where n is the outward pointing normal. By using $\psi = \frac{1}{C_\infty} (\cos \alpha, \sin \alpha)^T$ or $\psi = \frac{1}{C_\infty} (-\sin \alpha, \cos \alpha)^T$ along wall boundaries and otherwise 0, the functional represents the pressure drag coefficient c_D or the pressure lift coefficient c_L , respectively. C_∞ is a normalized reference value defined by $C_\infty = \frac{1}{2}\gamma\text{Ma}_\infty^2 p_\infty l$. Here, l is the chord length of the airfoil.

For the purely h-adaptive runs, $\theta = 0.02$ showed the best performance. In case of hp-adaptation, we used $\theta = 0.1$. In order to compute the error

p	1	2	3	4	hp
t_{DG}/t_{HDG}	0.885	1.364	1.267	1.417	1.582

Table 1: Runtime ratios for a fixed error level ($\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$)

in the drag coefficient, a reference value was obtained on a hp-adapted mesh with approximately $2.6 \cdot 10^5$ degrees of freedom.

In Fig. 4a, a purely h-adapted mesh can be seen. The most refined regions are the leading and trailing edge. The first is of importance as the flow experiences high gradients towards the stagnation point. Refinement of the latter is necessary as the flow is singular at this point due to the sharp trailing edge and the slip-wall boundary conditions. As soon as the error in these two regions is sufficiently low, other elements close to the airfoil get refined as well. For the hp-adapted mesh (see Fig. 4b) the leading and trailing edge are refined as well. All other regions, however, undergo mostly p-enrichment.

In terms of degrees of freedom, both HDG and DG show similar results (please note, that $\text{ndof} = \text{ndof}_w$ as this is a good measure for the resolution). For all computations it takes some adaptations until the critical regions, leading and trailing edge, are resolved. From this point on, one can see the benefit of a higher order discretization: the error drops significantly faster with respect to degrees of freedom and computational time (see Fig. 5 and 6).

In Tbl. 1, we give the runtime ratios for a fixed error level (we always choose the minimum level attained). Here, we can see that HDG is already faster from $p = 2$ on. This is due to the fact, that the adjoint is approximated with $p = 3$ so that the computation of the dual solution is faster using HDG.

5.2. Transonic Inviscid Flow over the NACA 0012 Airfoil

Next, we turn our attention to transonic flow which develops more complex features (e.g. compression shocks) compared to the subsonic regime. The flow is characterized by a free stream Mach number of $\text{Ma}_\infty = 0.8$ and an angle of attack of $\alpha = 1.25^\circ$.

As this flow has more features than the subsonic test case before, we chose higher parameters for adaptation, namely $\theta = 0.05$ for both mesh-adaptation and hp-adaptation. The reference value for the drag coefficient was obtained on a hp-adapted mesh with approximately $2.3 \cdot 10^5$ degrees of freedom.

In Fig. 7a a purely h-adapted mesh can be seen. The adjoint sensor detects all regions of relevance for the drag: the upper shock, the leading and trailing edge, and the lower weak shock. Further refinement is added upstream of the shock, where the adjoint has steep gradients and thus needs higher resolution. In the case of hp-adaptation, the mesh-refinement is stronger confined to the shock region and the trailing edge. The other features undergo p-enrichment.

As expected, both methods show a similar accuracy for a given number of degrees of freedom. The computations with $p = 2 \dots 4$ outperform $p = 1$ but are comparable to each other. Hp-adaptation shows very good results which is due to the accurate prediction of the solution smoothness (see Fig. 8 and 9).

Again, HDG is faster than DG from $p = 2$ on (see Tbl. 2). The hp-adaptive run is nearly 2.5 times faster.

p	1	2	3	4	hp
t_{DG}/t_{HDG}	0.57	1.309	1.669	1.708	2.492

Table 2: Runtime ratios for a fixed error level ($\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$)

5.3. Subsonic Laminar Flow over the NACA 0012 Airfoil

Finally, we consider viscous flow in the subsonic regime. The free stream Mach number is $\text{Ma}_\infty = 0.5$, the angle of attack $\alpha = 1^\circ$ and the Reynolds number $\text{Re} = 5000$. Due to the latter, a thin boundary layer develops around the airfoil.

The baseline mesh for the Navier-Stokes test case is more refined around the airfoil such that the boundary layer is correctly captured (see Fig. 10). It consists of 1781 elements and its far field is over a 1000 chords away.

Admissible target functionals defined on the boundary for the Navier-Stokes equations are given by the weighted boundary flux along wall boundaries, i.e.

$$J(w, \nabla w) = \int_{\partial\Omega} \psi \cdot (pn - \tau n) \, d\sigma \quad (49)$$

where n is the outward pointing normal. Here ψ is non-zero only on wall boundaries. By using $\psi = \frac{1}{c_\infty} (\cos \alpha, \sin \alpha)^T$ or $\psi = \frac{1}{c_\infty} (-\sin \alpha, \cos \alpha)^T$ along wall boundaries and otherwise 0, the functional represents the viscous drag coefficient c_D or the viscous lift coefficient c_L , respectively.

For this test case we use $\theta = 0.05$ during mesh-adaptation and $\theta = 0.1$ during hp-adaptation. The reference value for the drag coefficient was obtained on a hp-adapted mesh with approximately $2.5 \cdot 10^5$ degrees of freedom.

Both the h-adapted mesh (see Fig. 11a) and the hp-adapted mesh (see Fig. 11b) undergo refinement within the boundary layer and the wake region.

p	1	2	3	4	hp
t_{DG}/t_{HDG}	2.069	1.458	1.204	1.541	2.453

Table 3: Runtime ratios for a fixed error level ($\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$)

The mesh refinement for the hp-adaptive run is however more restricted to the leading edge region where the boundary layer develops. Further downstream, p-enrichment is used as soon as the necessary mesh-resolution is reached.

In terms of accuracy versus degrees of freedom HDG does a slightly better job than DG. This is most probably due to the better approximation of the gradient. The overall behavior is however the same. The higher the polynomial degree the more accurate and efficient the computations are for both HDG and DG (see Fig. 12 and 13). The difference between $p = 3$, $p = 4$ and hp is not as big, though. This might lead to the conclusion that isotropic mesh refinement is not longer efficiently applicable in cases involving strong gradients. Hence, the efficiency of the adaptation procedure is rather limited by the mesh refinement strategy.

Concerning the timings, we can see a similar trend as in the previous test cases (see Tbl. 3). For $p = 1$, however, HDG is twice as fast which is again caused by the approximation of the gradient (for a $p = 1$ DG discretization the gradient is an element-wise constant). For higher polynomial degree, the runtime ratios drop to reasonable numbers. The hp-adaptive HDG computation is 2.5 times as fast compared to the DG run.

6. Conclusion and Outlook

We presented an adjoint-based hp-adaptation methodology and compared it for hybridized and non-hybridized discontinuous Galerkin methods. Using hp-adaptation proved to be superior to pure h-adaptation if discontinuous or singular flow features were involved. In all cases, a higher polynomial degree turned out to be beneficial.

We showed that one can expect HDG to be faster than DG from $p = 2$ on. For viscous test cases, HDG yields more accurate lift and drag coefficients for the same number of degrees of freedom due to the better approximation of the gradient.

We plan to extend our computational framework to three dimensional problems. Then, adaptivity will play an even more crucial role, as the problem size increases drastically compared to the two dimensional case.

Furthermore, compressible flows are often dominated by anisotropic features, such as shocks or very thin boundary layers. Thus, taking this anisotropy into account during adaptation is crucial.

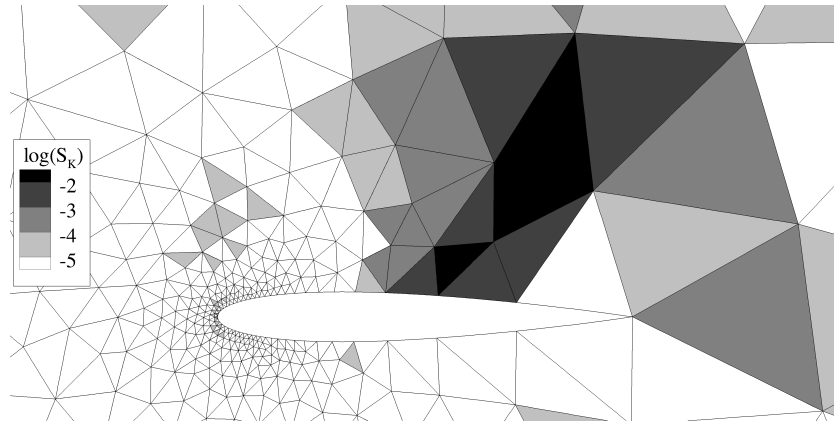
- [1] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of Discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39:1749–1779, 2002.
- [2] A. Balan, M. Woopen, and G. May. Adjoint-Based Hp-Adaptation for a Class of High-Order Hybridized Finite Element Schemes for Compressible Flows. *AIAA Paper*, 2938, 2013.
- [3] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang.

- PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
- [4] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc Web page, 2013. <http://www.mcs.anl.gov/petsc>.
 - [5] F. Bassi and S. Rebay. A High-Order Accurate Discontinuous Finite-Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 131:267–279, 1997.
 - [6] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin Solution of the Reynolds-Averaged Navier-Stokes and $k-\omega$ Turbulence Model Equations. *Computers & Fluids*, 34(4):507–540, 2005.
 - [7] M. Ceze and K.J. Fidkowski. Output-Driven Anisotropic Mesh Adaptation for Viscous Flows using Discrete Choice Optimization. *AIAA Paper*, 170, 2010.
 - [8] B. Cockburn and J. Gopalakrishnan. A Characterization of Hybridized Mixed Methods for Second Order Elliptic Problems. *SIAM Journal on Numerical Analysis*, 42(1):283–301, 2004.
 - [9] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for Conservation Laws II: General framework. *Mathematics of Computation*, 52:411–435, 1988.
 - [10] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for

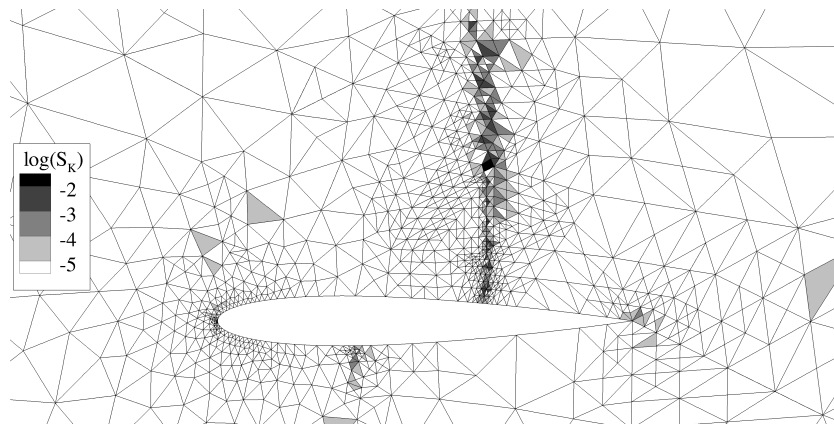
- Second Order Elliptic Problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, 2009.
- [11] B. Fraeijs de Veubeke. *Displacement and Equilibrium Models in the Finite Element Method*, chapter 9, pages 145–197. Wiley New York, 1965.
- [12] K.J. Fidkowski and D.L. Darmofal. Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics. *AIAA Journal*, 49(4):673–694, 2011.
- [13] R. Hartmann. Adaptive Discontinuous Galerkin Methods with Shock-Capturing for the Compressible Navier–Stokes Equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.
- [14] R. Hartmann. Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations. *SIAM Journal on Numerical Analysis*, 45(6):2671–2696, 2007.
- [15] R. Hartmann and P. Houston. Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations. *Journal of Computational Physics*, 183(2):508–532, 2002.
- [16] D.J. Mavriplis and A. Jameson. Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes. *AIAA Journal*, 28(8):1415–1425, 1990.
- [17] N.C. Nguyen and J. Peraire. An Adaptive Shock-Capturing HDG Method for Compressible Flows. *AIAA Paper*, 3060, 2011.

- [18] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable Discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228:8841–8855, 2009.
- [19] N.C. Nguyen, J. Peraire, and B. Cockburn. An Implicit High-Order Hybridizable Discontinuous Galerkin Method for Linear Convection-Diffusion Equations. *Journal of Computational Physics*, 228(9):3232–3254, 2009.
- [20] J. Peraire, N.C. Nguyen, and B. Cockburn. A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations. *AIAA Paper*, 363, 2010.
- [21] P.O. Persson and J. Peraire. Sub-cell Shock Capturing for Discontinuous Galerkin Methods. *AIAA Paper*, 112, 2006.
- [22] J. Schütz and G. May. An Adjoint Consistency Analysis for a Class of Hybrid Mixed Methods. *IMA Journal of Numerical Analysis*, 2013.
- [23] J. Schütz and G. May. A Hybrid Mixed Method for the Compressible Navier–Stokes Equations. *Journal of Computational Physics*, 240:58–75, 2013.
- [24] D.A. Venditti and D.L. Darmofal. Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows. *Journal of Computational Physics*, 176(1):40–69, 2002.

- [25] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22–46, 2003.
- [26] L. Wang and D. J. Mavriplis. Adjoint-based h-p adaptive discontinuous galerkin methods for the 2d compressible euler equations. *Journal of Computational Physics*, 228:7643–7661, 2009.
- [27] M. Woopen, A. Balan, G. May, and J. Schütz. Adjoint-Based Mesh Adaptation for a Class of High-Order Hybridized Finite-Element Schemes for Convection-Diffusion Problems. SIAM CSE13, SIAM Conference on Computational Science and Engineering, 2013.



(a) Coarse mesh



(b) Adapted mesh

Figure 1: Smoothness sensor for a transonic test case

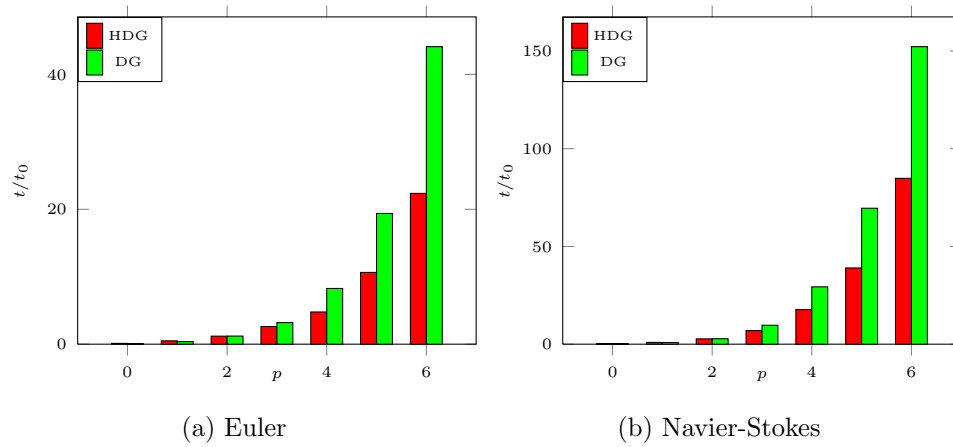


Figure 2: Runtime comparison of the hybridized and non-hybridized DG method for a fixed mesh and varying polynomial degree

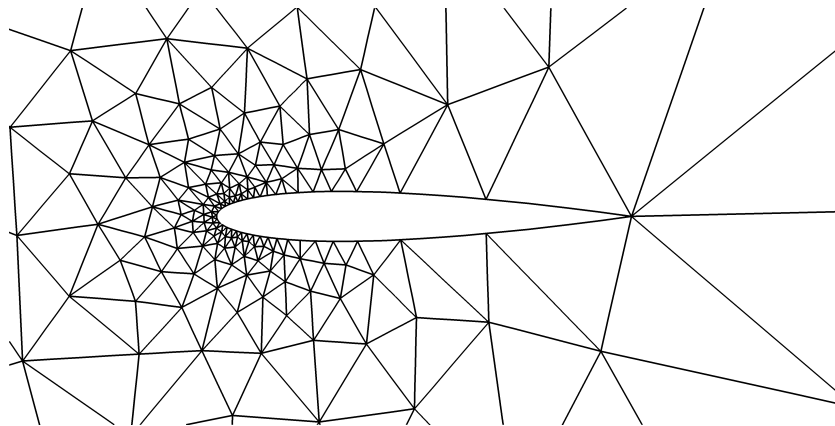
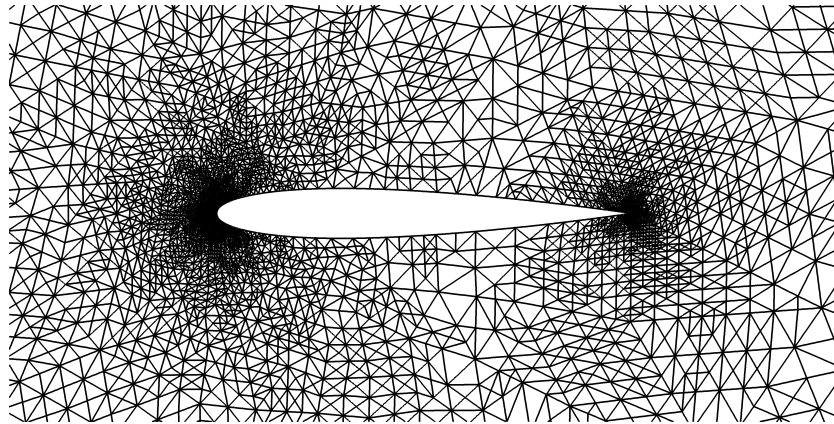
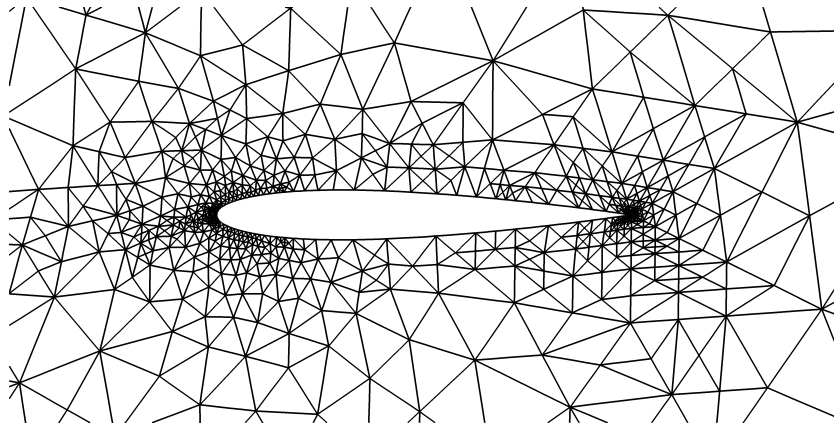


Figure 3: Baseline mesh with 719 elements for inviscid computations



(a) Pure h-adaptation ($p = 2$)



(b) hp-adaptation ($p = 2 \dots 5$)

Figure 4: Adapted meshes for the subsonic Euler test case ($\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$)

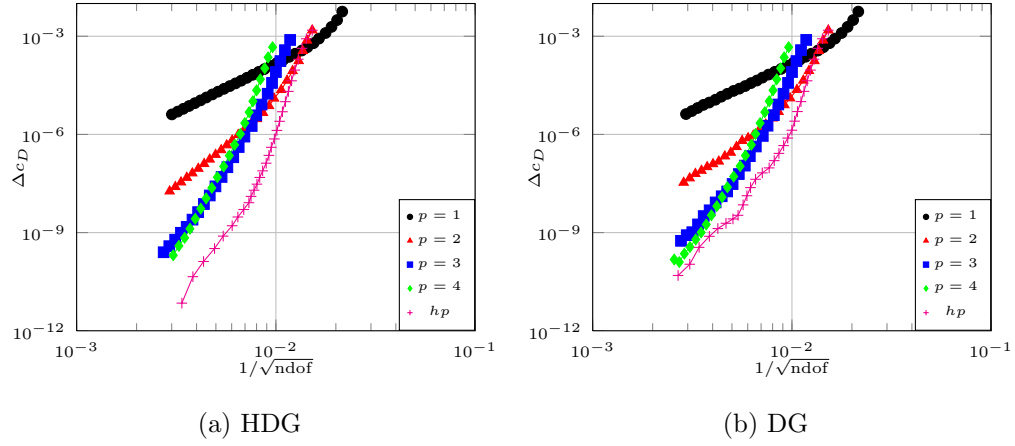


Figure 5: Drag convergence with respect to degrees of freedom ($\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$)

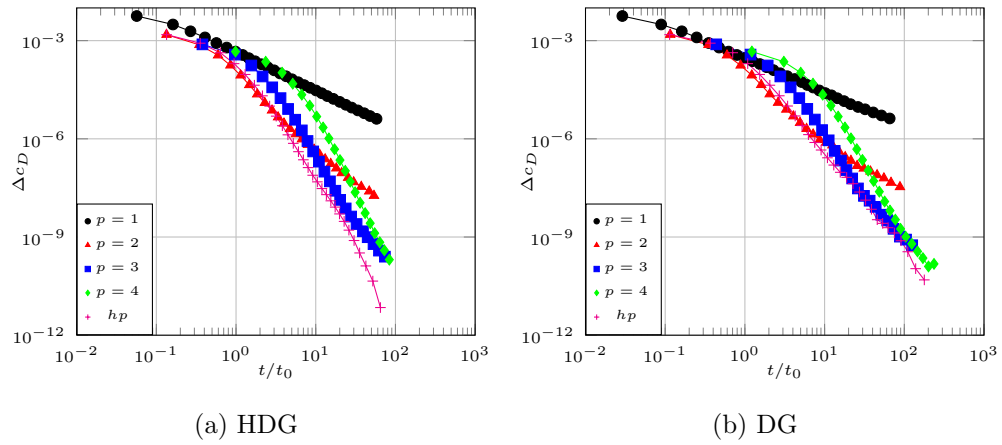
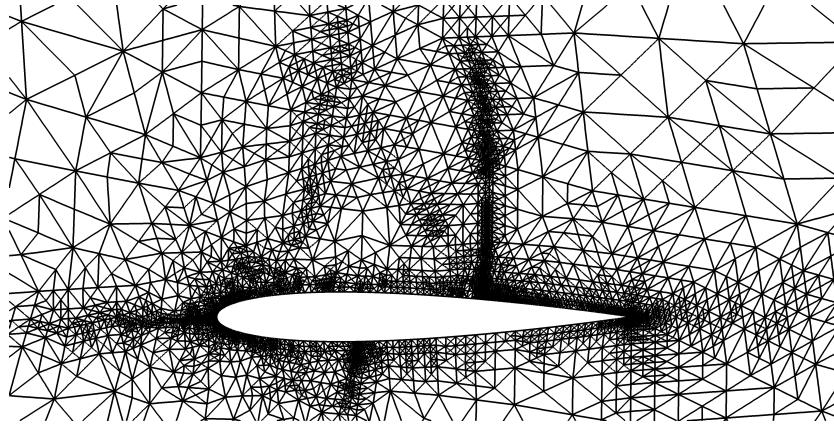
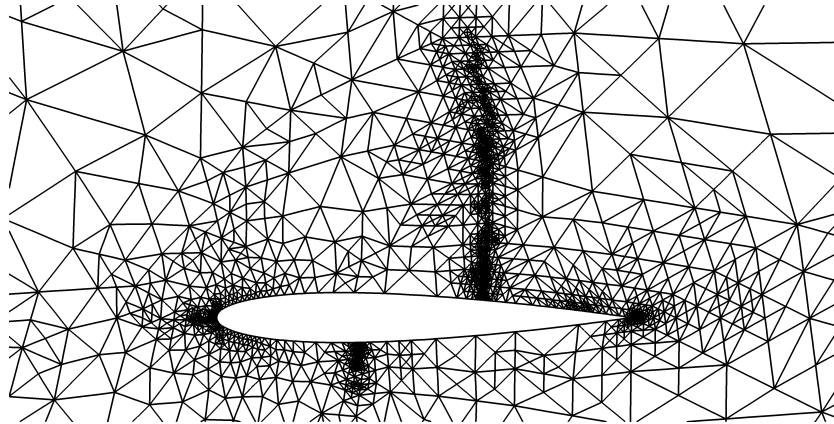


Figure 6: Drag convergence with respect to time ($\text{Ma}_\infty = 0.5$, $\alpha = 2^\circ$)



(a) Pure h-adaptation ($p = 2$)



(b) hp-adaptation ($p = 2 \dots 5$)

Figure 7: Adapted meshes for the transonic Euler test case ($\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$)

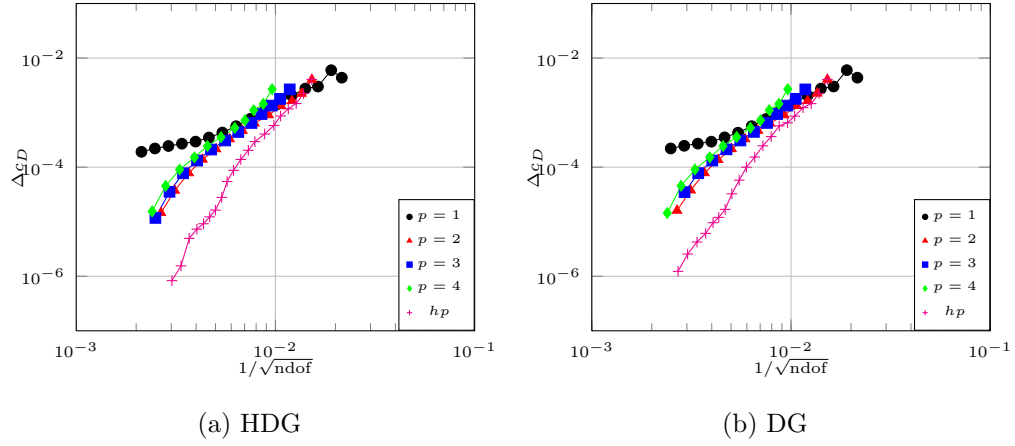


Figure 8: Drag convergence with respect to degrees of freedom ($\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$)

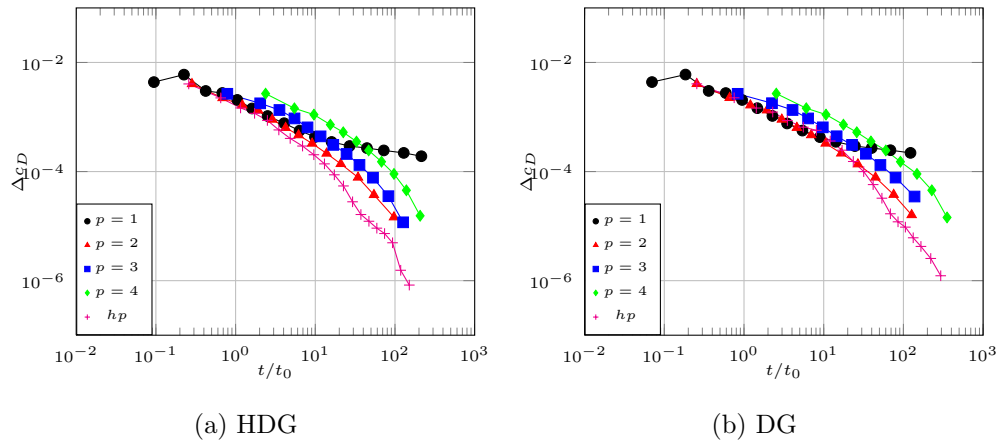


Figure 9: Drag convergence with respect to time ($\text{Ma}_\infty = 0.8$, $\alpha = 1.25^\circ$)

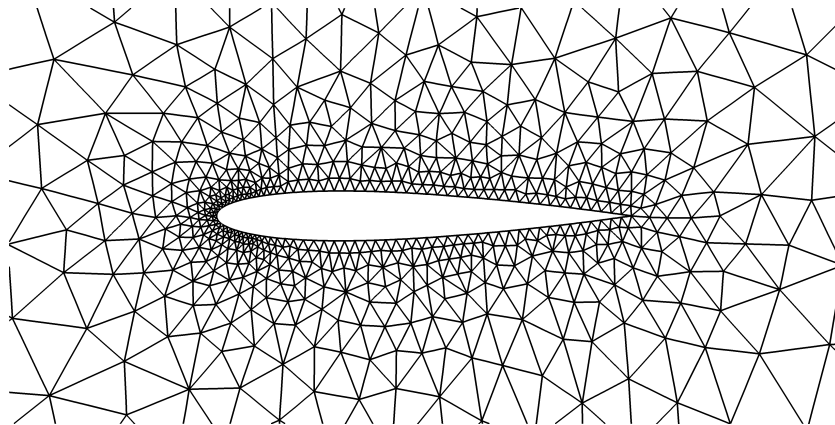
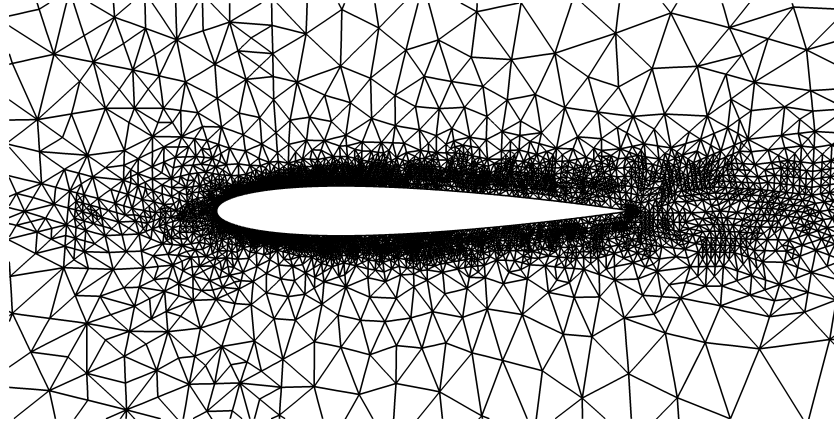
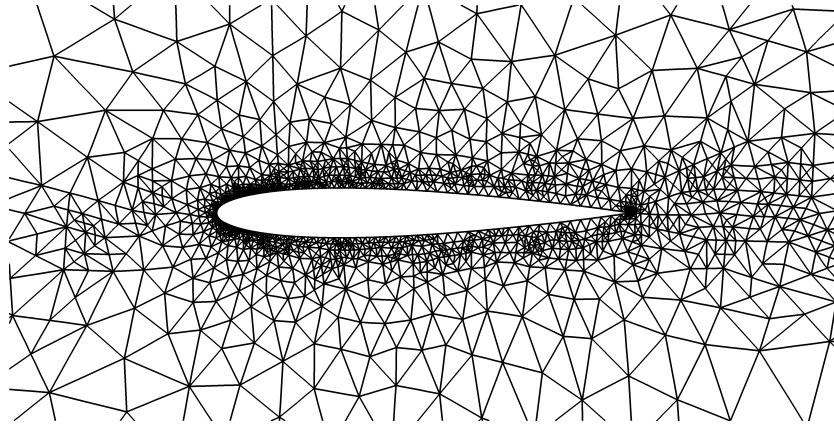


Figure 10: Baseline mesh with 1781 elements for viscous computations



(a) Pure h-adaptation ($p = 2$)



(b) hp-adaptation ($p = 2 \dots 5$)

Figure 11: Adapted meshes for the Navier-Stokes test case ($\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$)

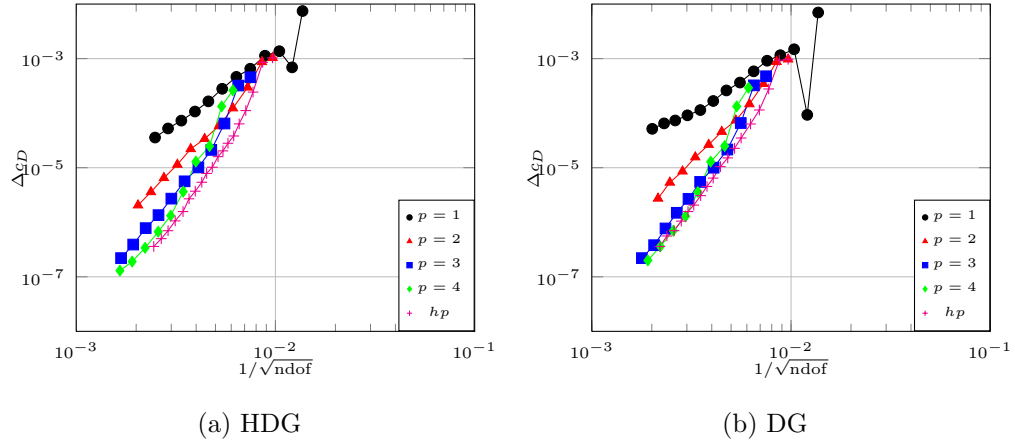


Figure 12: Drag convergence with respect to degrees of freedom ($\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$)

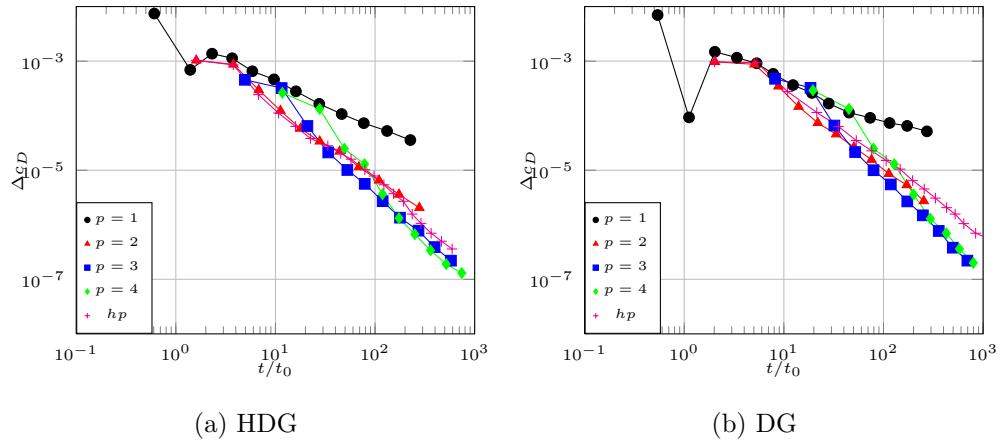


Figure 13: Drag convergence with respect to time ($\text{Ma}_\infty = 0.5$, $\alpha = 1^\circ$, $\text{Re} = 5000$)