Discontinuous Galerkin Methods for Computational Aerodynamics – 3D Adaptive Flow Simulation with the DLR PADGE Code

R. Hartmann, J. Held, T. Leicht[*], F. Prill

Deutsches Zentrum für Luft- und Raumfahrt, Institut für Aerodynamik und Strömungstechnik, 38108 Braunschweig

Abstract

Over the last few years, the discontinuous Galerkin method (DGM) has demonstrated its excellence in accurate, higher-order numerical simulations for a wide range of applications in computational physics. However, the development of practical, computationally efficient flow solvers for industrial applications is still in the focus of active research. This paper deals with solving the Navier-Stokes equations describing the motion of three-dimensional, viscous compressible fluids. We present details of the PADGE code under development at the German Aerospace Center (DLR) that is aimed at large-scale applications in aerospace engineering. The discussion covers several advanced aspects like the solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations, a curved boundary representation, anisotropic mesh adaptation for reducing output error and techniques for solving the nonlinear algebraic equations. The performance of the solver is assessed for a set of test cases.

Key words: Discontinuous Galerkin Method, Compressible Flows, Turbulent Flows, Error Estimation, Mesh Adaptation
PACS: 02.60, 47.27, 47.11.Fg, 47.85.Gj

1. Introduction

Initially developed for advection problems, the discontinuous Galerkin finite element method has recently been applied in the field of Computational Fluid Dynamics (CFD) with great success. Having a link to both the finite element method and the finite volume method, the DG approach offers the advantages and robustness of the classical Riemann fluxes, while maintaining the freedom to choose the local approximation order as well as the basis functions. Furthermore, the discretization lends itself to local mesh adaptation and efficient parallelization on modern distributed-memory computer architectures.

While first pioneering work on viscous CFD applications has been contributed by Lomtev, Quillen and Karniadakis [1] and Bassi and Rebay [2] more than a decade ago, applying the DGM to realistic engineering problems in an adaptive and parallel software framework remains a challenging task. Therefore, in the context of the European ADIGMA project [3], a coordinated European effort in advancing higher-order discretization methods in computational aerodynamics, the German Aerospace Center (DLR) has developed the PADGE code, an adaptive discontinuous Galerkin solver for 3D turbulent flow. In the following, we present the results of these activities and give a description of the major components of the flow solver.

The PADGE code is designed as a modular, object-oriented framework written in `C++` and is based on a modified version of the `deal.II` finite element library [4]. Due to its modular nature, the PADGE framework anticipates further extensions.

High-quality solutions for several test cases demonstrate the feasibility of our approach as well as the mature state of the implementation. Apart from the case of laminar flow, the presentation includes first experience with the DG discretization for turbulent cases.

2. Discontinuous Galerkin discretization

Compressible Navier-Stokes and k-ω Equations

The governing equations for the conservation of mass, momentum and energy can be written as follows:

$$\nabla\cdot\left(\mathcal{F}^c(\mathbf{u})-\mathcal{F}^v(\mathbf{u},\nabla\mathbf{u})\right)=\mathbf{0},$$

where by $\mathbf{u}=\left[\rho,\rho\mathbf{v},\rho E\right]^T$ we denote the state vector of conservative variables and $\mathcal{F}^c$, $\mathcal{F}^v$ are the convective and diffusive flux vectors, respectively. The system of equations is complemented by constitutive relations and appropriate boundary conditions for the problem, see, e. g., [5], Chapter 1, for details.

For the treatment of turbulent flows we consider the Reynolds-averaged Navier-Stokes equations with the Wilcox k-ω turbulence model equations [6, 7]:

$$\nabla\cdot\left(\mathcal{F}^c_{k,\omega}(\mathbf{u})-\mathcal{F}^v_{k,\omega}(\mathbf{u},\nabla\mathbf{u})\right)=\mathbf{P}_{k,\omega}-\mathbf{D}_{k,\omega},$$

The additional components $\mathcal{F}^c_{k,\omega}$ and $\mathcal{F}^v_{k,\omega}$ of the convective and diffusive flux vectors are determined by source terms $\mathbf{P}_{k,\omega}$, $\mathbf{D}_{k,\omega}$ representing production and destruction of the turbulence variables k and ω. The vector of conservative variables then reads $\mathbf{u}=\left[\rho,\rho\mathbf{v},\rho E,\rho k,\rho\omega\right]^T$.

[*] Corresponding author
Email addresses: `ralf.hartmann@dlr.de` (R. Hartmann), `joachim.held@dlr.de` (J. Held), `tobias.leicht@dlr.de` (T. Leicht), `florian.prill@dlr.de` (F. Prill)

An essential modification compared to standard k-ω model implementations has been proposed by Bassi and co-workers [8]. In particular, the variable $\ln(\omega)$ is used instead of ω itself. Solving the resulting system of equations becomes considerably easier due to the more moderate behavior of $\ln(\omega)$ in the vicinity of walls. We note that this variable transformation does not change the turbulence model, however, additional requirements are imposed on the model by the limitation of the turbulent variables. While k is simply kept non-negative, a lower bound of ω is locally derived from the realizability of the Reynolds stresses.

Discretization

Like other finite element methods (FEM) the discontinuous Galerkin discretization is based on a weak formulation of the governing equations. However, in contrast to standard continuous finite element methods, the discrete trial and test functions are *discontinuous* element-wise polynomial functions. Due to the discontinuity of the trial functions the normal fluxes $\mathcal{F}^c \cdot \mathbf{n}$ across an inter-element face are replaced by a numerical flux $\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$, which connects the two different flow states $\mathbf{u}_h^+$ and $\mathbf{u}_h^-$ between the two neighboring elements.

Any consistent and conservative flux of the many Riemann fluxes known in finite volume methods, e. g. the Lax-Friedrichs or Roe flux, can be chosen to ensure consistency and conservativity of the DG discretization. In fact, the lowest-order DGM based on element-wise constant basis functions resembles a basic first-order finite volume scheme. While finite volume methods achieve second and higher-order accuracy by reconstruction techniques, discontinuous Galerkin discretizations of arbitrary high order can be obtained simply by increasing the polynomial degree of the element-wise basis functions. The resulting DG discretization is of higher order independent of the particular choice of a consistent numerical flux function.

For viscous flows, continuity between neighboring elements is weakly imposed. In fact, DG discretizations are stabilized by the addition of specific inter-element penalization terms. This is typically done through the introduction of suitable lifting operators, cf. [2, 9]. In general, the computation of the lifting operator requires the inversion of local mass matrix problems on each face present in the computational mesh, see e. g. the second scheme (BR2) proposed by Bassi & Rebay [2]. Other schemes, like the interior penalty (IP) scheme, [10, 11], extended to viscous compressible flows in [12], replace the lifting operators by explicit penalty terms that can be evaluated directly. The PADGE code offers several variants of the BR2 and IP schemes.

Both the introduction of numerical fluxes and the weakly imposed continuity constraints give rise to surface integrals over inter-element boundaries. These supplement the volume integrals over each element, which are also present in standard conforming Galerkin methods.

The stencil of the DGM considered here is minimal in the sense that each element communicates only with its direct neighbors through information on the solution and its derivative at the common interface. This stencil, which is formally compact for any discretization order allows for simple parallelization as well as flexible mesh adaptation. Both ‘hanging nodes’ in the case of local edge subdivision (h-refinement) and variations of the order of the numerical scheme between neighboring elements (p-refinement) are treated naturally by simply integrating the convective and viscous fluxes over the interfaces.

Being based on a FEM space of discontinuous functions the DGM is not only globally but also locally conservative. Because of the finite element character of the DGM an elaborate error estimation framework is available, which can be utilized to design adaptive mesh refinement algorithms, see Section 5.

The PADGE code currently supports quadrilateral and hexahedral meshes. A discretization scheme is implemented for which the discrete adjoint problem is a consistent discretization of the continuous adjoint equations, cf. [12]. The solution to the discrete adjoint problem is used for error estimation and goal-oriented mesh refinement, see Section 5. It could also be used in optimization for an efficient computation of gradient directions and for ensuring a specific level of accuracy of each of the flow solutions.

3. Higher-order boundary representation

When exploiting the ability of higher-order discretization methods to generate accurate approximations on coarse meshes, a crucial point is to provide a proper representation of curved wall boundaries. Of course, any numerical approximation, not only DG methods, is expected to show the (physical) phenomena induced by polyhedral bodies if line segments or planar faces constitute the computational boundary. In fact, one can observe ‘disturbances’ of the numerical flow solution where the wall representation exhibits kinks, see according examples in [13, 14]. Spurious entropy may be generated and then transported with the flow along the boundary.

However, inserting additional elements close to a boundary with the sole purpose of resolving the geometry impedes the aims of higher-order methods. A more adequate approach we pursue within the PADGE code is to represent the boundary by piecewise polynomials of higher degree. We gain detailed information on the geometry, which is required by such an approximation, basically in two ways – via separate CAD data or via additional point data included in the mesh. Although the continuity of normal vectors across elements is not imposed on the polynomials, in practice, it has been found that a satisfactory representation of curved surfaces can be achieved. The normal vectors – though in general not identical – approximate each other sufficiently well as the representation of the boundary tends to the exact geometry, see Figure 1a for a schematic illustration.

Aerodynamic flows are mostly connected with high Reynolds numbers and feature very thin boundary layers. Typically, meshes with highly stretched cells are employed to resolve such flows efficiently. It is then necessary to combine a higher-order boundary representation with an adjustment of interior cells to the curvature of the boundary in order to avoid any intersection with the boundary, cf. Figure 1b. To this end, we apply a mesh transformation technique: Based on an analogy to the behavior

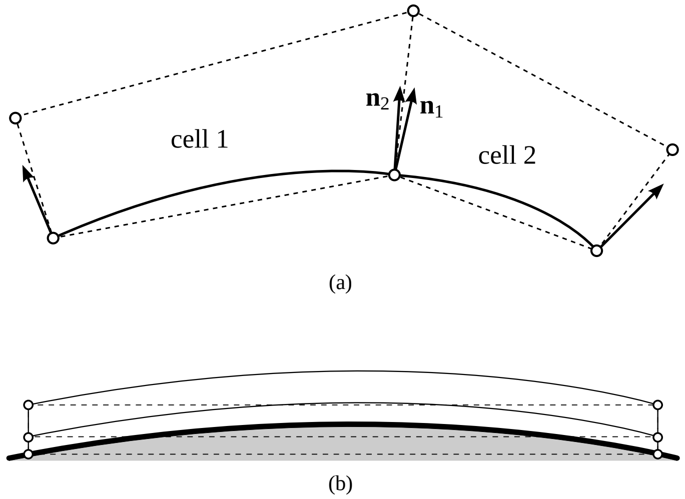


Figure 1: (a) The normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ at the common boundary of adjacent cells lie close together. (b) Interior cells might intersect with the boundary if not adjusted to its curvature.

of solid bodies under deformation an additional linear or nonlinear elasticity problem is solved, cf. [15].

It turns out to be disadvantageous when dealing with the computational mesh and CAD data separately – especially for 3D cases – that the assignment of geometrical elements to according mesh elements has to be defined afterwards, which is not a trivial task. Thus, a more favorable strategy to be pursued in the future is to incorporate extra point information during the mesh generation process and provide it along with the mesh.

4. Solution technique

For realistic applications in computational aerodynamics, the discrete problems arising from the DG discretization usually consist of several millions of unknowns. Therefore, regarding their low memory consumption and ease of implementation, Runge-Kutta time-marching schemes would be the optimal choice, especially due to the fact that mass matrices have block diagonal structure in the DGM. On the other hand, nonlinearity and anisotropies induced by the governing equations as well as by the computational mesh demand for a scalable and robust solution technique. In terms of the required CPU time, a fully implicit Newton-Krylov approach is a likely candidate for best results. Still, iterative solution methods that utilize the full Jacobian matrix remain limited to low-order discretizations due to their memory requirements.

In view of the long-term experience made with finite volume solvers, the nonlinear multigrid approach seems to be the most promising candidate to meet these challenges. In fact, the DGM lends itself to a particularly simple multilevel solution procedure: High-order numerical systems on a given computational mesh can be handled by exploiting the nested level hierarchy of spaces of varying polynomial degree. While the state vector in the original ansatz space is merely treated by a nonlinear smoothing iteration, e. g. a block Jacobi solver, the low-order discretizations are used for a defect-correction step. Recursive application of this strategy leads to a multilevel V-cycle as depicted in Figure 2. On the lowest-order discretization level, the standard Newton-GMRES approach is applied.

Nonlinear multi-p approaches of this type have previously been described and demonstrated in the DGM literature, see e. g. [16]. The PADGE code uses a special line-block decomposition [16, 17] as a basis for its smoothing procedure. The blocks correspond to sequences of elements (lines), which are oriented along the directions of strong coupling of the underlying flow field. This reduces the effect of local anisotropies of the mesh and the solution.

Since the level spaces are nested, the natural injection and restriction can be chosen for the intergrid transfer of the error. A hierarchical basis can be employed, which makes the injection particularly simple and memory efficient in its implementation. The nonlinear V-cycle also requires a restricted state vector, and we use the orthogonal projection as a state restriction operator. Finally, on the lowest-order level, which is usually constituted by piecewise constant or (bi-, tri-)linear functions, a Newton-Krylov iteration is employed. A nested iteration strategy is applied to obtain a proper initial approximation.

Still, the sparse, linearized coarse level problems might be infeasibly stiff for the solution with a standard ILU(0) preconditioned GMRES method. In this case, the described multi-p approach can be combined with a traditional h-multigrid method on a sequence of geometrically coarser meshes [18].

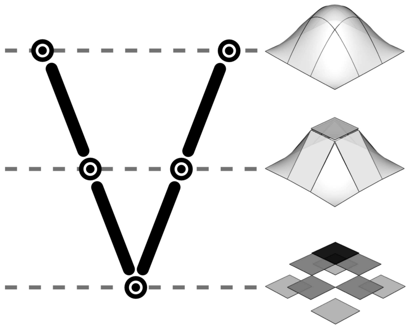


Figure 2: Schematic illustration of the p-hierarchical V-cycle. The lines indicate the prolongation and restriction between the constant, bilinear, and biquadratic polynomial spaces, while the dotted circles denote the nonlinear smoothing procedure.

5. Adaptivity

Residual-based and adjoint-based mesh refinement

Important quantities in aerodynamic flow simulations are the aerodynamic force coefficients like the 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 their error. Being a finite element method, the DGM offers a powerful theoretical framework for the derivation of error estimates. In particular, by employing a duality argument, error estimates can be obtained for estimating the error measured in terms of the aerodynamic force coefficients. The error estimate is based on local residuals of the computed flow solution multiplied by the solution to an adjoint problem, which in turn is related to the force coefficient. The error estimate can be rewritten as a sum of local *adjoint-based* indicators, also called dual-weighted-residual (DWR) indicators [19], which can be employed to drive a goal-oriented adaptive mesh refinement algorithm specifically tailored to the accurate and efficient approximation of the aerodynamic force coefficient, see e. g. [19–21] among others.

Error estimation and adjoint-based mesh refinement rely on the solution to an additional linear adjoint problem. In the numerical examples Section 6 the linear adjoint problem is solved on the same grid but with a polynomial degree increased by one. Furthermore, the PADGE code allows to use a patch recovery of the adjoint solution computed on the same mesh and with the same polynomial degree as the flow solution. In the framework of adaptive mesh refinement, the adjoint solution offers valuable information on how the discretization error on each element affects the error in the force coefficient under consideration. Successively refining those elements that contribute most to this error and possibly coarsening others yields meshes on which the computed target quantity is particularly accurate. Furthermore, based on the adjoint solution, an estimate of the discretization error in the computed quantity can be obtained. This estimate can also be used to enhance the computed quantity, which in many cases significantly increases its accuracy, see e. g. [22, 23].

In the PADGE code, this error estimation and adjoint-based mesh refinement approach is available and has been extended to treat multiple force coefficients simultaneously, see [20]. Here, an adjoint solution for a suitably combined target quantity drives the goal-oriented mesh refinement. Furthermore, the solution to a discrete linear error equation (also called the adjoint-adjoint equation) solved with an increased polynomial degree gives error estimates for an arbitrary number of target quantities.

Provided that the adjoint solution related to an arbitrary target quantity is sufficiently smooth, the corresponding error representation can be bounded from above by an error estimate that includes the primal residuals but is independent of the adjoint solution. By localizing this error estimate, so-called *residual-based* indicators can be derived. Mesh refinement based on these indicators leads to meshes which resolve all flow features irrespective the target quantity, see e. g. [21, 22]. The residual-based indicators are cheap to evaluate, because they do not depend on an auxiliary problem to be solved. They are particularly well suited for resolving the overall flow field, e. g. vortices can be resolved over long distances. However, they are in general not very efficient in approximating force coefficients. For the latter, the adjoint-based mesh refinement should be preferred.

Local mesh refinement algorithm

Employing either residual-based or adjoint-based indicators, an estimate of the contribution to the error can be obtained for each element. A new mesh is then constructed by selecting a given fraction of the elements with the largest error indicators and splitting their edges, forming new child elements. This way, mesh resolution is added only locally where needed, i. e. where errors are large.

Anisotropic mesh refinement

As aerodynamic flows exhibit strong anisotropic features like shocks and boundary layers, it is not always efficient to split all of an element's edges when performing local refinement. Selectively splitting edges orthogonal to anisotropic layers and thus forming only two or four children from one hexahedral element instead of the eight children of an isotropic splitting might be more efficient. After selecting the elements of the mesh to be refined as described above, we analyze the inter-element jumps present in the discrete DG solution and associate large jumps with insufficient resolution along the same direction. Thus, whenever the jumps evaluated over two opposite faces are small compared to other jumps on the same element, we do not split the edges orthogonal to those faces, see [22, 24].

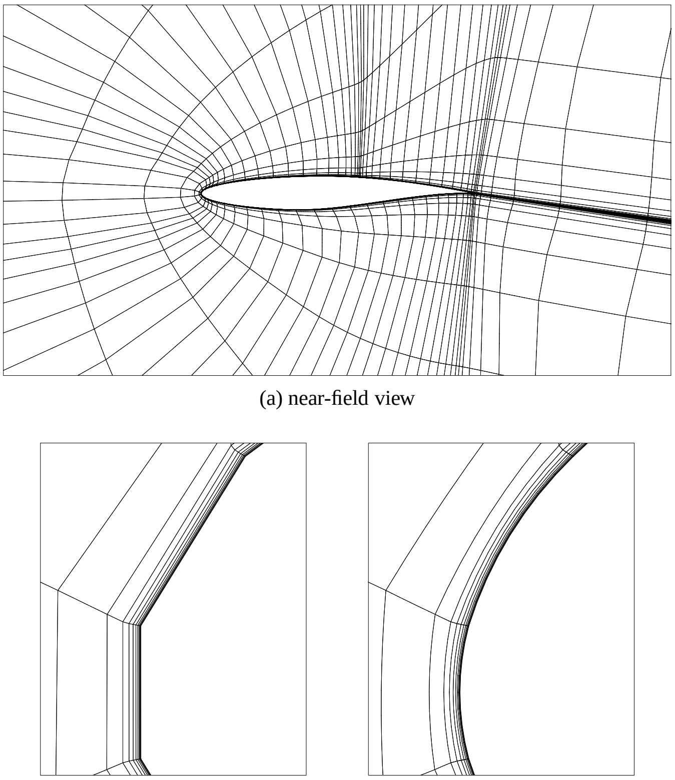
In the literature, anisotropic adaptation is mostly applied in the context of linear continuous finite elements or finite volume schemes on simplical meshes. These approaches are usually based on interpolation error estimates, see [23, 25–27] among others. For higher-order DG discretizations this concept has also been considered in [24], but it suffers from the underlying smoothness assumption on the solution of the flow problem. Furthermore, due to the inherent recovery procedure to obtain higher-order derivatives, it is difficult to extend this approach to hp-discretizations with locally varying approximation order, which are, however, naturally treated by the jump indicator used in the PADGE code.

6. Numerical examples

RAE 2822

In order to demonstrate the capability to handle meshes with highly stretched elements along curved boundaries, the laminar flow over the RAE 2822 airfoil at a Mach of number $M = 0.5$, Reynolds number of $Re = 10\,000$ and an angle of attack of $\alpha = 1^\circ$ is computed. The block-structured mesh is the coarse level of a mesh originally designed for high Reynolds number turbulent flow computations with a second-order finite volume code and exhibits a maximum cell aspect ratio of 3 000. For this case a piecewise quartic polynomial basis and the Roe numerical flux function is used and the nonlinear system is solved using a backward Euler method combined with an ILU preconditioned GMRES. The mesh and a detailed view of the leading-edge region are shown in Figure 3. The curved mesh representation has been generated using a piecewise quadratic approximation based on a CAD description of the airfoil. Obviously, due to the coarseness of the mesh, the piecewise linear mesh is unable to capture the geometry in detail, whereas the quadratic representation is already a good and quite smooth approximation of the exact geometry.

The effect of this higher-order boundary treatment on the computed solution is prominent especially in the computed skin



(a) near-field view

(b) leading-edge, linear

(c) leading-edge, quadratic

Figure 3: Computational mesh around the RAE 2822 airfoil, illustrating the effect of piecewise linear and piecewise quadratic boundary approximation.

friction coefficient c_f, which is shown in Figure 4. Although the solution is allowed to be discontinuous between elements, the resulting skin friction coefficient is almost continuous in the case of a curved boundary approximation, indicating the good resolution achieved by the higher-order flow solution. The solution in case of the piecewise linear boundary approximation follows the same global trend, but is locally superimposed by an unphysical element-wise oscillation which yields quite large local errors. This effect is obviously more pronounced if either the solution is large in absolute values or changes rapidly. This difference in the solution quality motivates the deployment of a higher-order boundary treatment in spite of the additional effort associated with such an approach.

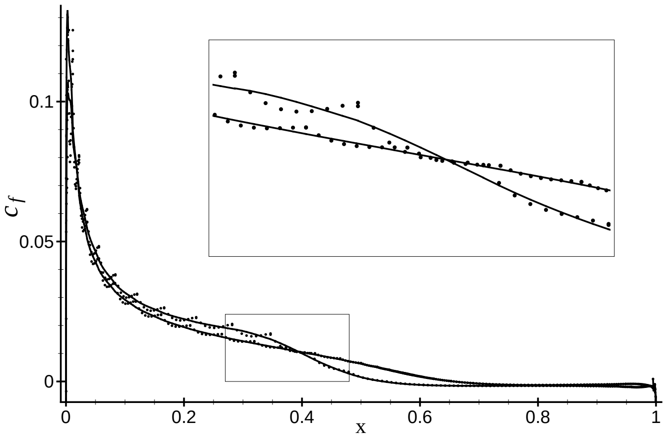


Figure 4: Computed skin friction coefficient c_f for the RAE 2822 airfoil: linear boundary approximation (dotted) and curved boundary approximation (solid line).

We now consider the drag coefficient C_d computed on a sequence of nested meshes originating from the same block-structured mesh by uniform coarsening. Based on a reference value $C_d^{\mathrm{ref}} \approx 0.0407$ obtained by computations on fine meshes we plot the error $|C_d - C_d^{\mathrm{ref}}|$ in the computed drag coefficients vs. the number of degrees of freedom for different polynomial degrees of the elemental basis functions in Figure 5. In the logarithmic plot the approximately straight line for each polynomial degree indicates a constant order of convergence. For increasing polynomial degree, these lines exhibit an increased slope, which corresponds to a higher order and thus a faster convergence of the drag coefficient under mesh refinement. This example supports the theoretical result that basis functions of higher polynomial degree result in higher-order methods.

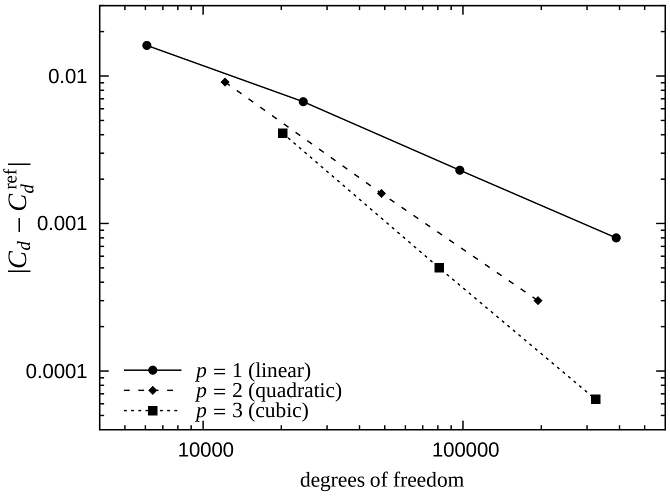


Figure 5: Error in the computed drag coefficient for a sequence of nested meshes using different polynomial degrees p for the basis functions of each element.

Laminar flow over a delta wing

As a second more complex example, we consider the laminar flow at a Mach number of $M = 0.3$, a Reynolds number of $Re = 4\,000$ and an angle of attack of $\alpha = 12.5°$ around a delta wing with sharp leading-edge and a blunt-trailing edge. This test case has been considered in the EU project ADIGMA [3] and in [22]. A similar case was treated earlier in [28]. Here, we use the Vijayasundaram flux and an ILU preconditioned Newton-GMRES solver.

In the following, we consider the error of the lift coefficient C_l. Similar results have been obtained for the drag coefficient C_d. We start by computing the lift from the second-order flow solution on a sequence of globally refined meshes starting from

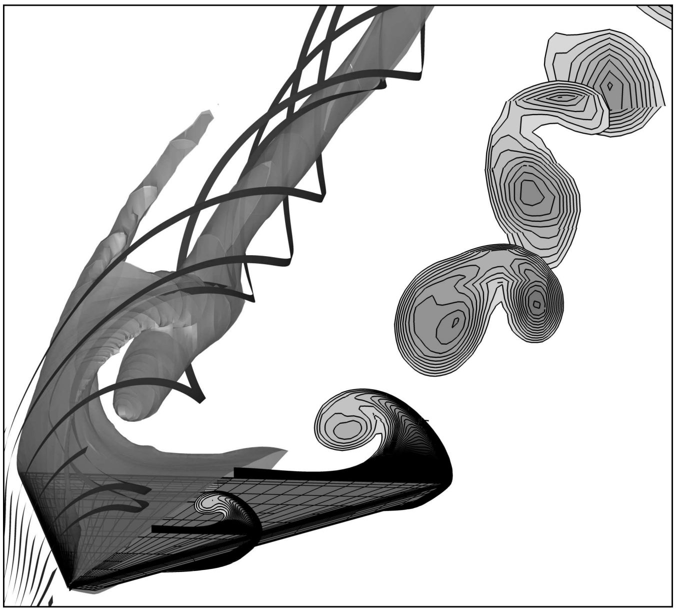


Figure 6: Streamlines and Mach number iso-surface over the port wing as well as Mach number slices over the starboard wing.

a very coarse mesh (3 264 elements) for the half-span wing with symmetry boundary conditions. We then consider local adaptive mesh refinement starting from the solution on the initial coarse mesh.

Figure 7 plots the error in the lift coefficient vs. the number of elements for various refinement strategies. Compared to global mesh refinement, lift coefficients of a specific accuracy are obtained with less elements using residual-based mesh refinement. We notice that the adjoint-based refinement procedure yields even better results.

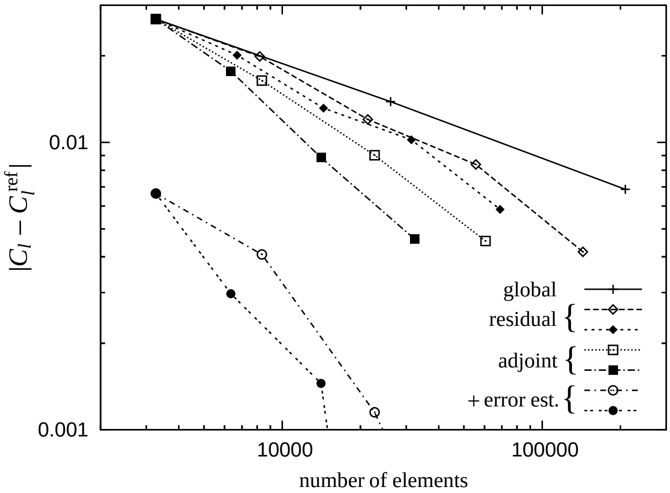
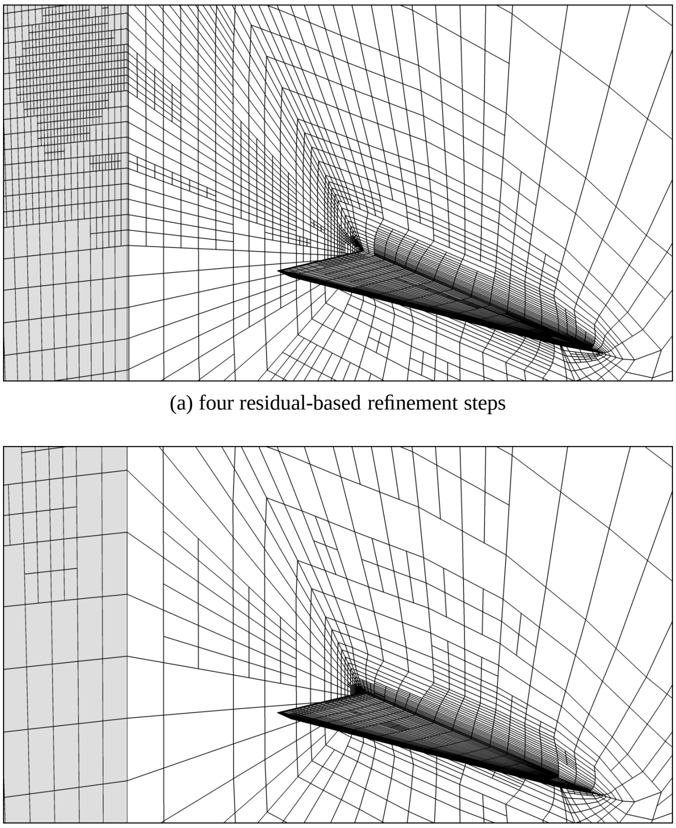


Figure 7: Delta wing: Error in the computed lift coefficient for sequences of locally refined meshes using different refinement indicators and isotropic (open symbols) as well as anisotropic refinement (filled).

Additionally, in the case of adjoint-based mesh refinement, Figure 7 illustrates the error of the enhanced lift coefficients obtained by adding the global error estimate to the computed lift coefficient. Already on the first adapted mesh the enhanced lift coefficient is more accurate than that computed on the last adapted meshes.

Finally, we note that anisotropic mesh refinement performs better than isotropic mesh refinement, requiring half the number of elements in the adjoint-based case, see Figure 7. In general, the gain becomes more obvious with increasing accuracy requirements.



(a) four residual-based refinement steps

(b) three adjoint-based refinement steps

Figure 8: Adaptive isotropic mesh refinement for the delta wing using residual-based and adjoint-based refinement indicators. The lift values computed on both meshes have similar errors.

The resulting meshes are visualized in Figure 8, where the selected refinement step corresponds to the last data point in Figure 7, so the accuracy of the computed lift coefficient is comparable. The major effect is the good resolution of the vortex visible in the cut-plane behind the wing for the residual-based refinement indicator and the corresponding lack of resolution in this area in the case of goal-oriented adjoint-based refinement. It is quite obvious that the global flow field is better resolved using the first indicator while the resolution of this prominent vortex is not of much influence on the lift value, as both the pressure at the wall and the skin friction are only weakly dependent on the downstream vortex evolution. Thus, concentrating the refinement closer to the wing the adjoint-based refinement indicators are capable of creating more efficient meshes for the approximation of the given target quantity.

Nonlinear solution procedure

Finally, some details of the iterative solution process, sketched in Section 4, are discussed. We consider the discrete nonlinear system of equations arising from a 5th order BR2 discretization of the delta wing test case. A straight-forward, fully implicit solution approach on the coarse 3 264 element mesh with tensor product elements would require the assembly of a Jacobian matrix that contains of $8.58 \cdot 10^9$ non-zero entries (2.04 million unknowns). The p-hierarchical V-cycle offers a memory efficient alternative with a coarse-level Jacobian matrix consisting of only $5.5 \cdot 10^5$ non-zeros (16 320 degrees of freedom). For the hexahedral mesh the storage requirements of the line-implicit smoother is reduced by a factor of three compared to the full Jacobian matrix. In order to further reduce the memory requirements the smoother could also be reassembled on each smoother application which is perfectly parallelizable.

Here, we employ a line-implicit three-stage Runge-Kutta smoother on a V-cycle with 5 levels, and we apply two pre- and postsmoothing steps on each level. We note that the application of a multi-stage smoother still requires only a single inversion of the block-tridiagonal line preconditioner. Figure 9 shows the residual convergence history of a nested iteration, plotted vs. the number of V-cycles. A nested strategy is applied in order to get a physical initial state: The solution process starts with the piecewise trilinear discretization and successively prolongates the solution approximation after a sufficient reduction of the residual has been achieved.

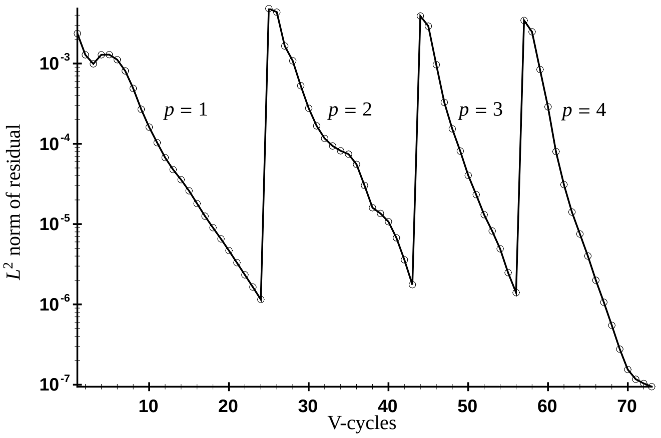


Figure 9: 5th order solution of the laminar flow around the delta wing: Residual L^2 norm convergence history of the p-hierarchical V-cycle. Parallel computation on 32 processors.

In terms of the number of iterations, the multi-p V-cycle yields a convergence rate roughly uniform w. r. t. the number of levels. The algorithm is formally defined in the same way as a standard FAS multigrid. A closer look reveals, however, that the computational cost of the nonlinear smoothing iteration does not scale with the number of unknowns. Therefore, the complexity grows in a disproportionate manner for discretizations of very high order. However, in practice, i. e. for the moderate polynomial degrees used in computational aerodynamics, this fact plays a minor role.

L1T2 high-lift configuration

We close our series of examples with results of computations of the turbulent flow past a typical high-lift configuration, the L1T2 three-element airfoil. The geometry of this configuration is shown in Figure 10. The flow conditions are given by a Mach number of $M = 0.197$, a Reynolds number of $Re = 3.52 \cdot 10^6$ and an angle of attack of $\alpha = 20.18°$. This case has been documented extensively in the literature, see e. g. [29, 30]. In particular, there is data of two wind tunnel experiments available, see [31], in the following referred to as experiment 1 and experiment 2.

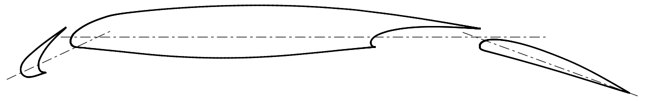


Figure 10: Geometry of the L1T2 three-element airfoil. The slat angle is 25°, the flap angle is 20°.

We present numerical results generated with the PADGE code in comparison with results generated with the well validated finite volume code TAU, see [32], as well as with experimental data. The BR2 discretization of the turbulent equations in PADGE includes the local Lax-Friedrichs flux and is solved fully coupled using the backward Euler method. The PADGE computations are performed with polynomial degrees $p = 3$ and $p = 4$, each on the same quadrilateral mesh with 4 740 elements. This mesh originates from a 75 840 element mesh by agglomerating twice. The curved mesh representation in this case is realized by a piecewise quartic approximation based on extra point data, which has been extracted from the original mesh. Reference results have been produced on the original mesh by means of the TAU code.

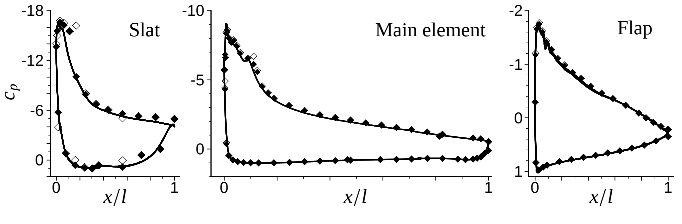


Figure 11: Pressure distributions for each L1T2 airfoil element computed with PADGE (solid line) compared to reference results with TAU (dotted) and data of experiment 1 (open symbols) and experiment 2 (filled).

First, we look at the results concerning the pressure distribution over each of the airfoil elements, i. e. slat, main element and flap. Figure 11 shows the output of the PADGE code being in a good agreement with the experimental data and with only minor differences compared to the TAU reference results. The number of degrees of freedom (dof) were nearly the same in both cases: The PADGE code result was generated with the polynomial degree $p = 4$, which corresponds to a number of dof of about $4.3 \cdot 10^5$. The computation with the TAU code involved about $4.6 \cdot 10^5$ dof.

Moreover, we consider the skin friction distribution computed with the PADGE code in comparison to the TAU reference result. The result for $p = 4$ is overall in good agreement with the reference as can be seen in Figure 12. However, in contrast to the reference, the skin friction features a quasi-laminar drop on the upper side of the flap, although the computation did not include a prescribed transition from laminar to turbulent conditions. Figure 12 also shows considerable differences between the computed skin friction distribution for $p = 3$ and $p = 4$. With the decreased number of dof (case $p = 3$) the PADGE result does not meet the reference result. This fact underlines the requirement for local refinement strategies in order to take advantage of higher-order methods. This remains an essential aspect of our activities and will be considered in the future, especially within the framework of *hp*-methods.

Finally, Table 1 contains computed values of lift and drag coefficients for $p = 3, 4$ beside reference values of a TAU computation and wind tunnel data. The lift coefficients computed with PADGE for $p = 4$ and TAU differ by only 1.1%, while the drag coefficients deviate by 70 drag counts. It should be mentioned that corresponding results documented in the literature (as well as the experimental values itself) vary considerably. This might indicate a strong influence of many different parameters making a reliable prediction a complex matter. Both the lift coefficient computed with PADGE with $p = 4$ as well as the one computed with TAU are below the measured values, which is at least partly due to the fact that both simulations were fully turbulent.

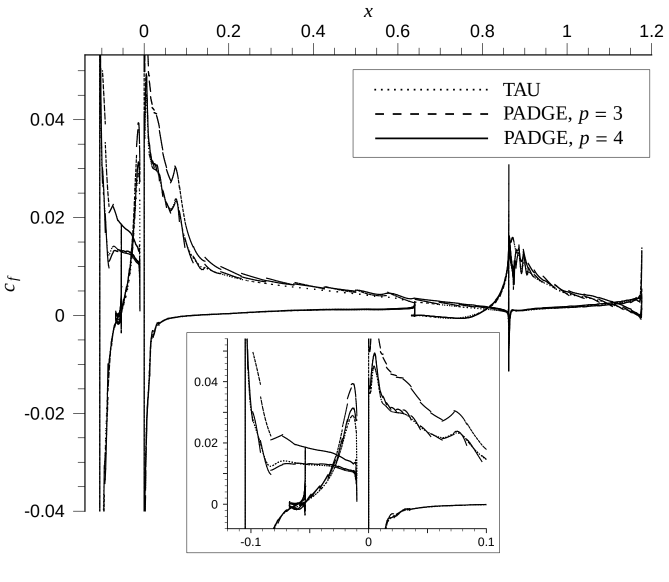


Figure 12: Comparison of computed skin friction distributions with detailed view of the slat region.

	PADGE results		References		
	$p = 3$	$p = 4$	TAU	Exp. 1	Exp. 2
C_l	4.017	3.976	3.932	4.110	4.075
C_d	0.0560	0.0674	0.0744	0.0677	0.0734

Table 1: Computed lift (C_l) and drag (C_d) coefficients for $p = 3, 4$ compared to the reference values with TAU and the experimental values.

7. Outlook

The examination of two- and three-dimensional laminar test cases solved with the PADGE code clearly demonstrates the ability of the underlying higher-order method and the local adaptive mesh refinement approach to produce highly accurate solutions with comparatively few degrees of freedom. In order to exploit this potential for the computation of complex aerodynamic flows, several issues have to be considered.

There is also first experience in PADGE with turbulent flows using the scheme of Bassi et al.[8]. Extending the ability to simulate turbulent flows will be the focus of future developments. In order to cover the transonic flow regime a shock capturing technique will be considered. Approaches incorporating the treatment of shocks in the discrete equations through an artificial viscosity term are probably well suited to both implicit solution algorithms and the adjoint-based error estimation procedure. Such models have already been successfully used in the DG context, see e. g. [33, 34]. Another aspect of physical modeling, the treatment of unsteady flows, will be covered by a time-discretization scheme. These three extensions combined will form the first steps towards the transition from the model problems considered so far to a wide range of aircraft flow phenomena.

For the applicability of the method to complex geometries it seems essential to extend the underlying meshes from purely hexahedral meshes to unstructured hybrid meshes. With these it is possible to have high quality prismatic and hexahedral elements generated by an advancing layer technique in the vicinity of bodies, while the remaining domain can easily be partitioned into simplices. Generating completely block-structured meshes is generally more difficult, cannot be done automatically and is thus time-consuming.

In the DG context, the definition of a mesh contains both the partition into elements of given type and the polynomial basis functions used on those elements. So far, a uniform polynomial degree and thus the same basis functions have been used for all elements of a mesh. Since the DGM allows for discontinuities across inter-element boundaries, the method readily extends to local variations of the polynomial degree according to resolution requirements of the underlying solution. Considering both local mesh and local order adaptation at once is possible in *hp*-refinement algorithms, [35, 36]. The most challenging aspect of such a technique is the definition of an automatic indicator deciding locally whether an h-subdivision or an increase of the polynomial degree p is the more effective strategy.

Up to now, the focus of the PADGE code has been on capability rather then efficiency. While the hitherto existing results clearly demonstrate the efficiency of the discretization, measured in the required amount of degrees of freedom needed for a given accuracy, the efficiency of the solver itself is influenced by many factors as outlined in Section 4. For large-scale problems on todays' computer architectures, we believe that it is necessary to develop a solver satisfying a compromise between memory and runtime efficiency. This can be achieved by incorporating a semi-implicit solution technique with reduced memory requirements and efficient problem-specific preconditioners based on line smoothing, p-multilevel and h-multigrid approaches [16, 17, 37]. The development of such an algorithm and its reliable and robust implementation will constitute a major effort.

As a final outlook, the entirety of the suggested enhancements of the PADGE code will enable a thorough investigation of complex large-scale aerodynamic flow phenomena. These results will then provide the answer to the question for which type of applications the DG approach is beneficial and for which cases well established second-order finite volume schemes should be preferred. Current results already indicate that higher accuracy requirements are beneficial to our method.

Acknowledgements. The authors gratefully acknowledge 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 [3].

References

[1] I. Lomtev, C. Quillen, G. Karniadakis, Spectral/hp methods for viscous compressible flows on unstructured 2D meshes, J. Comput. Phys. 144 (2) (1998) 325–357.

[2] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini, A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows, in: R. Decuypere, G. Dibelius (Eds.), 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, March 5–7, 1997, Technologisch Instituut, 1997, pp. 99–108.

[3] N. Kroll, ADGIMA – A European project on the development of adaptive higher-order variational methods for aerospace applications, 47th AIAA Aerospace Sciences Meeting, AIAA 2009-176 (2009).

[4] W. Bangerth, R. Hartmann, G. Kanschat, deal.II – A general purpose object oriented finite element library, ACM Transactions on Mathematical Software 33 (4).

[5] M. Feistauer, J. Felcman, I. Straškraba, Mathematical and Computational Methods for Compressible Flow, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2003.

[6] D. C. Wilcox, Reassessment of the scale-determining equation for advanced turbulence models, AIAA J. 26 (11) (1988) 1299–1310, 1988.

[7] D. C. Wilcox, Turbulence Modeling for CFD, DCW Industries, Inc., La Canada CA, 1993.

[8] F. Bassi, A. Crivellini, S. Rebay, M. Savini, Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-\omega$ turbulence model equations, Computers & Fluids 34 (2005) 507–540.

[9] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, J. Comp. Phys. 131 (1997) 267–279.

[10] D. Arnold, An interior penalty finite element method with discontinuous elements, Ph.D. thesis, The University of Chicago, Chicago, IL (1979).

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

[12] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations, J. Comput. Phys. 227 (22) (2008) 9670–9685.

[13] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comp. Phys. 138 (1997) 251–285.

[14] B. Landmann, M. Kessler, S. Wagner, E. Kramer, A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows, Comput. Fluids 37 (4) (2008) 427–438.

[15] K. Stein, T. Tezduyar, R. Benney, Mesh moving techniques for fluid-structure interactions with large displacements, J. Appl. Mech. 70 (2003) 58–63.

[16] K. J. Fidkowski, T. A. Oliver, J. Lu, D. L. Darmofal, p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, J. Comput. Phys. 207 (2005) 92–113.

[17] D. J. Mavriplis, On Convergence Acceleration Techniques for Unstructured Meshes, Tech. Rep. TR-98-44, ICASE, NASA Langley Research Center (1998).

[18] F. Prill, M. Lukáčová-Medviďová, R. Hartmann, A Multilevel Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations, in: A. H. et al. (Ed.), Algoritmy 2009. 18th conference on scientific computing, Vysoké Tatry – Podbanské, Slovakia, 2009, pp. 91–100.

[19] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numerica 10 (2001) 1–102.

[20] R. Hartmann, Multitarget error estimation and adaptivity in aerodynamic flow simulations, SIAM J. Sci. Comput. 31 (1) (2008) 708–731.

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

[22] T. Leicht, R. Hartmann, Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations, J. Comput. Phys.In review.

[23] D. A. Venditti, D. L. Darmofal, Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows, J. Comp. Phys. 187 (2003) 22–46.

[24] T. Leicht, R. Hartmann, Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations, Int. J. Numer. Meth. Fluids 56 (11) (2008) 2111–2138.

[25] M. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, Int. J. Numer. Meth. Fluid 25 (1997) 475–491.

[26] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems, Appl. Numer. Math. 51 (2004) 511–533.

[27] P. J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Meth. Appl. Mech. Engrg. 194 (2005) 5068–5082.

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

[29] I. Fejtek, Summary of code validation results for a multiple element airfoil test case, 28th AIAA fluid dynamics conference, AIAA Paper 97-1932 (1997).

[30] A. Hellsten, New Two-Equation Turbulence Model for Aerodynamics Applications, Tech. Rep. Report No. A-21, Helsinki University of Technology, Laboratory of Aerodynamics (2004).

[31] I. R. M. Moir, Measurements on a two-dimensional aerofoil with high-lift devices, AGARD Advisory Report 303, Advisory Group for Aerospace Research & Development, Neuilly-sur-Seine, test case A2 (1994).

[32] D. Schwamborn, T. Gerhold, R. Heinrich, The DLR TAU-code: Recent applications in research and industry, in: P. Wesseling, E. Oñate, J. Périaux (Eds.), Proceedings of European Conference on Computational Fluid Dynamics ECCOMAS CDF 2006, Delft The Netherland, 2006, pp. 91–100.

[33] P. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2006-0112 (2006).

[34] F. Bassi, A. Crivellini, A. Ghidoni, S. Rebay, High-order discontinuous Galerkin discretization of transonic turbulent flows, 47th AIAA

Aerospace Sciences Meeting, AIAA 2009-180 (2009).

[35] K. Harriman, P. Houston, B. Senior, E. Süli, *hp*–Version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form, in: Recent Advances in Scientific Computing and Partial Differential Equations, Vol. 330 of Contemporary Mathematics, AMS, 2003, pp. 89–119.

[36] L. Wang, D. J. Mavriplis, Adjoint-based *h*-*p* adaptive discontinuous Galerkin methods for the 2d compressible Euler equations, J. Comput. Phys. 228 (20) (2009) 7643–7661.

[37] C. R. Nastase, D. J. Mavriplis, High-order discontinuous Galerkin methods using an *hp*-multigrid approach, J. Comput. Phys. 213 (1) (2006) 330–357.