ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

CURVED GRID GENERATION AND DG COMPUTATION FOR THE DLR-F11 HIGH LIFT CONFIGURATION

R. Hartmann[1], **H. McMorris**[2], **T. Leicht**[1]

[1]DLR (German Aerospace Center)
Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, 38104 Braunschweig, Germany
e-mail: {ralf.hartmann, tobias.leicht}@dlr.de

[2] CentaurSoft
3409 Executive Center Drive, Suite 241, Austin, TX 78731
e-mail: harlan@centaursoft.com

Keywords: High-order grid generation, high-order Discontinuous Galerkin method, compressible flows, high lift configuration

Abstract. *The DLR-F11 high lift configuration previously considered at the 2nd AIAA High Lift Prediction workshop (HiLiftPW-2) consists of a wing-body configuration with a three-element high lift system including slat tracks and flap track fairings. This test case has been selected a "computational/meshing challenge" in the 4th International Workshop on High-Order CFD Methods due to its geometrical and computational complexity. In this article, we demonstrate the applicability of current grid generation technology (CENTAUR) to generate a quadratic curved mesh for this configuration. Furthermore, we employ a high-order Discontinuous Galerkin discretization for the Reynolds-averaged Navier-Stokes (RANS) equations with the Wilcox-kω turbulence model which is sufficiently stable to be applicable to this test case. Finally, a 3rd-order Discontinuous Galerkin (DG) flow solution is computed on the quadratic (3rd-order) computational mesh. The numerical results are compared to experimental data as well as to computational results published in the HiLiftPW-2.*

1 INTRODUCTION

In recent years there has been significant progress in the development of curved grid generation as well as of high-order aerodynamic flow solvers, cf. the EU-projects ADIGMA [19] and its successor IDIHOM [20], for example. Nevertheless, in terms of geometrical and aerodynamic complexity the capabilities reached were still far from being applicable to "real-world" industrial test cases.

Particularly problematic was the generation of computational meshes sufficiently coarse to be suited to higher order methods and the subsequent curving of these meshes for a higher order approximation of curved wall boundaries, which is known to be particularly difficult in the presence of the highly stretched mesh elements typically used in the boundary layer of high Reynolds number aerodynamic flow simulations. There have been a few efforts to generate high-order meshes for flow field computations as demonstrated in [7], [22], and [28]. The majority of these existing methods start with a relatively coarse linear grid and insert the extra points needed to convert each element to high-order. There have also been methods that start with a curved, high-order surface mesh. This surface is then marched outwards in order to create the volume elements. The robustness of both the conversion method and the marching process is an issue in the vicinity of highly curved convex and concave regions within the geometry. A primary issue for high-order meshes is to ensure validity of the generated elements [17]. Work on creating high order elements only in curved regions of the geometry has also been studied [18].

Also, the high-order flow solvers reached their limits in terms of the stability of the discretizations and the solvers employed. Much effort concentrated on the development and industrialization of high-order CFD methods [19, 20] as well as on comparing the methods in the International Workshops on High-Order CFD Methods [25]. However, the configurations treated were bound to low to medium complexity. More recently, Discontinuous Galerkin discretizations were applied to the Common Research Model (CRM) wing-body configuration [6, 10] and to a simple 3D high lift configuration in [15]. Here, it was particularly important for the stability of the discretization to use a discretization at the wall boundary which resembles as closely as possible the discretization on interior faces, and to use an according discretization of force coefficients to keep adjoint consistency [14].

The DLR-F11 high lift configuration previously considered at the 2nd AIAA High Lift Prediction workshop [1] consists of a wing-body configuration with a three-element high lift system including slat tracks and flap track fairings. This test case has been selected a "computational / meshing challenge" in the 4th International Workshop on High-Order CFD Methods [16] due to its geometrical and computational complexity.

In the current work, we give details on how we create a coarse linear mesh for this configuration using the CENTAUR grid generator [5], and describe the algorithms developed to curve the mesh with piecewise quadratic polynomials for a 3rd-order boundary approximation. Furthermore, we describe the main ingredients of the discretization employed [14, 15] which we found sufficiently stable to be applicable to a test case of this complexity. Finally, the 3rd-order DG flow solution computed on the quadratic (3rd-order) computational mesh is compared to experimental data [1] as well as to numerical results published in the HiLiftPW-2 [1].

2 LINEAR AND QUADRATIC GRID GENERATION

In order to generate curved meshes for high-order CFD simulations, two problems must be solved. First, a coarse linear mesh must be generated that both sufficiently models the geometry

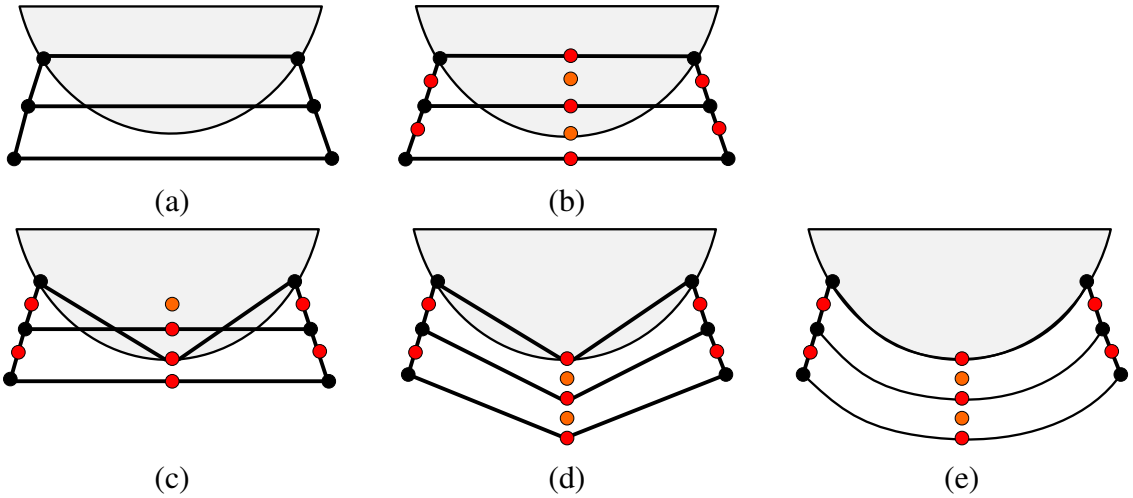


Figure 1: Sequence of steps used by CENTAUR to construct curved, high-order grids.

and ensures that any subsequent high-order mesh is not too big to use for simulations. Second, a robust method to curve the linear mesh is required. This method must be able to work for all types of elements. In particular, the method must be able to handle highly stretched mesh elements used to model the boundary layer region in high Reynolds number aerodynamic flow simulations. The method must also be able to handle highly curved surfaces in both convex and concave regions.

For the purpose of this work, the CENTAUR grid generator [5] is used to create a quadratic curved 3D hybrid grid. The basic algorithm used to create a curved, high-order mesh is as follows:

(a) Generate a coarse, linear hybrid grid.

(b) Insert an additional midpoint for every grid edge and quadrilateral face.

(c) Use the CAD information to map each new boundary point onto the underlying CAD surface.

(d) Adjust the position of the interior points based on the mapped position of the boundary edge midpoints, in order to prevent self-intersecting grid elements and to ensure grid validity.

First, a coarse linear 3D hybrid grid is created. CENTAUR allows for the mesh generation process to be tuned so that fewer elements are created while still preserving the geometry. For the surface mesh generation, a lower analytic curvature clustering is used along with a larger maximum element size. Next, prisms or hexahedra generated in the boundary layer are similarly adjusted to increase the first layer thickness in preparation for the added high-order points. Then, tetrahedra are used to automatically fill the remainder of the domain matching the length scales from both the surface mesh and the boundary layer mesh.

Second, the linear mesh is converted into a high-order mesh. As shown in Figure 1(b), new mid-edge and mid-face points are added to all edges and quadrilateral faces in the linear mesh. Any new point that is located on a boundary surface is then mapped onto the CAD surface using the same geometry routines used during mesh generation to place points on the surface. At this stage, the mesh may be invalid in convex curved regions as seen in Figure 1(c). While in other

curved regions, the mesh point spacing may be valid but distorted after the mapping process is completed. To correct both types of problems, the motion of the boundary points is smoothed into the interior as shown in Figure 1(d).

After this process is completed, the grid quality and validity is evaluated using the volumes of the complete elements and sub-elements formed using the points added in Step (b). To ensure good quality, volume ratios of the elements and sub-elements are also used. If any element violates the quality or validity measures, its points are adjusted to ensure the criteria are met. This process iterates until the mesh meets the various criteria.

3 DG DISCRETIZATION OF THE RANS AND WILCOX k-ω EQUATIONS

In this section, we give details on the Discontinuous Galerkin discretization similar to that previously developed in [14] and already applied to a simple 3D high lift configuration in [15]. In particular, we consider the Reynolds-averaged Navier-Stokes (RANS) equations and the Wilcox k-ω turbulence model equations [26, 27],

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})) = \mathcal{S}(\mathbf{u}, \nabla\mathbf{u}) \quad \text{in } \Omega, \tag{1}$$

where $\mathbf{u} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E, \rho k, \rho\tilde{\omega})^\top$ is the vector of conservative variables, with ρ, E, k, and $\mathbf{v} = (v_1, v_1, v_3)^\top$, denoting the density, specific total energy, turbulence kinetic energy, and velocity vector, respectively. Similar to Bassi *et al.* [3] the equations are considered in terms of the auxiliary variable $\tilde{\omega} = \ln\omega$ instead of the specific dissipation rate ω for a more moderate near-wall behavior of the variable and for guaranteeing positivity of ω. Additionally, we apply some realizability conditions for the turbulent stresses [3, 11]. For a detailed description of the convective and viscous fluxes, $\mathcal{F}^c(\mathbf{u})$ and $\mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})$, and the source terms $\mathcal{S}(\mathbf{u}, \nabla\mathbf{u})$ involved in (1), and of the boundary conditions imposed on the boundary $\Gamma = \partial\Omega$, we refer to [9, 13]. In this work, adiabatic no-slip wall boundary conditions, $\mathbf{v} = 0$, $\mathbf{n} \cdot \nabla T = 0$, are imposed on the wall boundary Γ_{W}.

Let the domain Ω be subdivided into a shape-regular mesh $\mathcal{T}_h = \{\kappa\}$ consisting of (possibly curved) elements κ. Furthermore, let $\mathbf{V}_h^p$ be the finite element space consisting of discontinuous vector-valued polynomial functions of degree $p \geq 0$ on $\mathcal{T}_h$, then the Discontinuous Galerkin discretization of (1) employed in this work is given by: Find $\mathbf{u}_h$ in $\mathbf{V}_h^p$ such that

$$\begin{aligned}\int_\Omega \{(-\mathcal{F}^c(\mathbf{u}_h) + \mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h)) : \nabla_h\mathbf{v}_h - \mathcal{S}(\mathbf{u}_h, \nabla_h\mathbf{u}_h) \cdot \mathbf{v}_h\} \, \mathrm{d}\mathbf{x} \\ + \sum_{\kappa\in\mathcal{T}_h} \int_{\partial\kappa} \left(\hat{\mathbf{h}}_h - \underline{\hat{\sigma}}_h\mathbf{n}\right) \cdot \mathbf{v}_h \, \mathrm{d}s + \sum_{\kappa\in\mathcal{T}_h} \int_{\partial\kappa} (\hat{\mathbf{u}}_h - \mathbf{u}_h) \otimes \mathbf{n} : \left(G^\top(\mathbf{u}_h)\nabla\mathbf{v}_h\right) \, \mathrm{d}s = 0\end{aligned} \tag{2}$$

for all $\mathbf{v}_h \in \mathbf{V}_h^p$, where $\mathbf{n}|_{\partial\kappa}$ denotes the outward unit normal vector to the boundary $\partial\kappa$ of element κ. Here, the convective and diffusive numerical flux functions, $\hat{\mathbf{h}}_h$ and $\underline{\hat{\sigma}}_h$, are approximations to the normal convective flux $\mathcal{F}^c(\mathbf{u}_h) \cdot \mathbf{n}$ and the viscous flux $\mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h)$, respectively. On an interior face $\partial\kappa \cap \partial\kappa'$ between two neighboring elements $\kappa, \kappa' \in \mathcal{T}_h$, the numerical flux functions

$$\begin{aligned}\hat{\mathbf{h}}_h &= \hat{\mathbf{h}}(\mathbf{u}_h, \mathbf{n}) = \hat{\mathbf{h}}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}), \\ \underline{\hat{\sigma}}_h &= \underline{\hat{\sigma}}(\mathbf{u}_h, \nabla\mathbf{u}_h) = \underline{\hat{\sigma}}(\mathbf{u}_h^+, \mathbf{u}_h^-, \nabla\mathbf{u}_h^+, \nabla\mathbf{u}_h^-),\end{aligned} \tag{3}$$

connect the interior and the exterior traces, $\mathbf{u}_h^+$ and $\mathbf{u}_h^-$, and their derivatives, $\nabla\mathbf{u}_h^+$ and $\nabla\mathbf{u}_h^-$, of

$\mathbf{u}_h$. On a boundary face $\partial\kappa \cap \Gamma \neq \emptyset$, the numerical boundary flux functions,

$$\begin{aligned} \hat{\mathbf{h}}_h|_\Gamma &= \hat{\mathbf{h}}_{\Gamma,h} = \hat{\mathbf{h}}_\Gamma(\mathbf{u}_h^+, \mathbf{n}), \\ \underline{\hat{\sigma}}_h|_\Gamma &= \underline{\hat{\sigma}}_{\Gamma,h} = \underline{\hat{\sigma}}_\Gamma(\mathbf{u}_h^+, \nabla\mathbf{u}_h^+). \end{aligned} \tag{4}$$

depend on the interior trace $\mathbf{u}_h^+$, directly and/or through the boundary function $\mathbf{u}_\Gamma(\cdot)$ given by

$$\mathbf{u}_\Gamma(\mathbf{u}_h) = (u_{h,1}, 0, 0, 0, u_{h,5}, 0, u_{h,1}\tilde{\omega}_{\text{wall}})^\top \qquad \text{on } \Gamma_{\text{W}}, \tag{5}$$

Here, $\tilde{\omega}_{\text{wall}}$ is determined using a projection of the analytic $\ln(\omega)$ near-wall behavior onto the polynomial ansatz space in the elements at the wall [23]. Similarly, the vector-valued numerical flux function $\hat{\mathbf{u}}_h$ in (2) is an approximation to $\mathbf{u}_h$ and is given by $\hat{\mathbf{u}}_h = \hat{\mathbf{u}}(\mathbf{u}_h) = \hat{\mathbf{u}}(\mathbf{u}_h^+, \mathbf{u}_h^-)$ on interior faces and by $\hat{\mathbf{u}}_h|_\Gamma = \hat{\mathbf{u}}_{\Gamma,h} = \hat{\mathbf{u}}_\Gamma(\mathbf{u}_h^+)$ on boundary faces. Finally, $G(\mathbf{u})$ denotes the homogeneity tensors defined by $\mathbf{f}_k^v(\mathbf{u}, \nabla\mathbf{u}) = G_{kl}(\mathbf{u})\partial\mathbf{u}/\partial x_l$, $k, l = 1, 2, 3$, for $\mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h) = (\mathbf{f}_1^v, \mathbf{f}_2^v, \mathbf{f}_3^v)$. Assuming that the numerical fluxes $\hat{\mathbf{h}}_h$ and $\underline{\hat{\sigma}}_h$ are consistent, then (2) is a consistent discretization of the flow equations (1), cf. [13, 14] for more details.

3.1 Numerical fluxes on interior faces

In the following, we give details on the numerical fluxes used on interior faces. For the convective numerical flux $\hat{\mathbf{h}}$ a Roe flux is chosen based on the diagonalization of the full Jacobian of the convective flux [24]. Furthermore, an entropy fix is employed similar to that of Harten [8]. Finally, the numerical fluxes for the diffusive terms are those of the BR2 scheme [3],

$$\hat{\mathbf{u}}_h = \{\!\{\mathbf{u}_h\}\!\}, \qquad\qquad \underline{\hat{\sigma}}_h = \{\!\{G(\mathbf{u}_h)\nabla_h\mathbf{u}_h\}\!\} - \underline{\delta}(\mathbf{u}_h), \tag{6}$$

where the penalization term $\underline{\delta}(\mathbf{u}_h)$ is given by

$$\underline{\delta}(\mathbf{u}_h) = C_{\text{BR2}}\{\!\{G(\mathbf{u}_h)\underline{L}_0^e(\mathbf{u}_h)\}\!\}. \tag{7}$$

Here $\{\!\{\underline{\tau}\}\!\} = \frac{1}{2}(\underline{\tau}^+ + \underline{\tau}^-)$ denotes the standard mean used in the DG context, and $\underline{L}_0^e(\mathbf{u}_h)$ denotes the local lifting operator, cf. [3, 14] for more details. Note that the penalization constant C_{BR2} must be sufficiently large for stability [4] and is taken as the number of faces of an element.

3.2 Numerical fluxes on wall boundary faces

The discretization on the boundary is chosen as close as possible to the discretization employed on interior faces. In particular, we use the same numerical flux functions for defining the numerical boundary fluxes as employed on interior faces, i.e., we consider

$$\hat{\mathbf{h}}_\Gamma(\mathbf{u}_h^+, \mathbf{n}) = \hat{\mathbf{h}}(\mathbf{u}_h^+, \mathbf{u}_\Gamma^-(\mathbf{u}_h^+), \mathbf{n}), \;\; \hat{\mathbf{u}}_{\Gamma,h} = \{\!\{\mathbf{u}_h\}\!\}_\Gamma, \;\; \underline{\hat{\sigma}}_{\Gamma,h} = \{\!\{\widetilde{\mathcal{F}}^v(\mathbf{u}_h, \nabla\mathbf{u}_h)\}\!\}_\Gamma - \underline{\tilde{\delta}}_\Gamma(\mathbf{u}_h^+), \tag{8}$$

where the boundary mean value $\{\!\{\cdot\}\!\}_\Gamma$ of $\mathbf{u}_h$ is defined by $\{\!\{\mathbf{u}_h\}\!\}_\Gamma = \frac{1}{2}\left(\mathbf{u}_h^+ + \mathbf{u}_\Gamma^-(\mathbf{u}_h^+)\right)$, and the wall exterior state $\mathbf{u}_\Gamma^- = \mathbf{u}_\Gamma^-(\mathbf{u}_h^+)$ is obtained by mirroring the interior state $\mathbf{u}_h^+$ at the wall boundary state $\mathbf{u}_\Gamma(\mathbf{u}_h^+)$ (cf. (5)) according to $\frac{1}{2}\left(\mathbf{u}_h^+ + \mathbf{u}_\Gamma^-(\mathbf{u}_h^+)\right) = \mathbf{u}_\Gamma(\mathbf{u}_h^+)$. Furthermore, for the BR2 scheme [3] the penalization term $\underline{\tilde{\delta}}_\Gamma$ in (8) is given by

$$\begin{aligned} \underline{\tilde{\delta}}_\Gamma(\mathbf{u}_h^+) &= C_{\text{BR2}}\{\!\{\widetilde{G}(\mathbf{u}_h)\underline{L}_\Gamma^e(\mathbf{u}_h)\}\!\}_\Gamma = \tfrac{C_{\text{BR2}}}{2}\left(\widetilde{G}(\mathbf{u}_h^+)\underline{L}_\Gamma^e(\mathbf{u}_h^+) + \widetilde{G}(\mathbf{u}_\Gamma^-)\underline{L}_\Gamma^e(\mathbf{u}_\Gamma^-)\right) \\ &= C_{\text{BR2}}\{\!\{\widetilde{G}(\mathbf{u}_h)\}\!\}_\Gamma\, \underline{L}_\Gamma^e(\mathbf{u}_h^+), \end{aligned} \tag{9}$$

where $\underline{L}^e_\Gamma(\mathbf{u})$ denotes the local lifting operator on the boundary with $\underline{L}^e_\Gamma(\mathbf{u}^+_h) = \underline{L}^e_\Gamma(\mathbf{u}^-_\Gamma)$, cf. [13, 14] for more details. Finally, the diffusive flux $\widetilde{\mathcal{F}}^v$ in (8) and the corresponding homogeneity tensor $\widetilde{G}$ in (9) is modified on the adiabatic walla boundary Γ_{W} such that no heat flux is added, corresponding to the adiabatic condition $\mathbf{n} \cdot \nabla T = 0$, i. e.,

$$\mathbf{n} \cdot \widetilde{\mathcal{F}}^v(\mathbf{u}, \nabla\mathbf{u}) = \mathbf{n} \cdot \Big(\widetilde{G}(\mathbf{u})\nabla\mathbf{u}\Big) = (0,\ (\underline{\tau}\,\mathbf{n})_1,\ (\underline{\tau}\,\mathbf{n})_2,\ \mathbf{n} \cdot (\underline{\tau}\,\mathbf{v}))^\top \text{on } \Gamma_{\mathrm{W}}. \tag{10}$$

The boundary treatment of fluxes in (8) corresponds to introducing a ghost layer of elements at the wall boundary and evaluating the discretization on the wall boundary like on interior faces. Finally, $(\nabla\mathbf{u})^-_\Gamma$ in $\{\!\{\widetilde{\mathcal{F}}^v(\mathbf{u}_h, \nabla\mathbf{u}_h)\}\!\}_\Gamma = \frac{1}{2}(\widetilde{\mathcal{F}}^v(\mathbf{u}^+_h, \nabla\mathbf{u}^+_h) + \widetilde{\mathcal{F}}^v(\mathbf{u}^-_\Gamma, (\nabla\mathbf{u})^-_\Gamma))$ is the wall exterior gradient which might depend on the interior state and gradient but is chosen as $(\nabla\mathbf{u})^-_\Gamma = \nabla\mathbf{u}^+_h$, for simplicity, which results in $\{\!\{\widetilde{\mathcal{F}}^v(\mathbf{u}_h, \nabla\mathbf{u}_h)\}\!\}_\Gamma = \{\!\{\widetilde{G}(\mathbf{u}_h)\}\!\}_\Gamma \nabla\mathbf{u}^+_h$.

3.3 Evaluation of force coefficients

Let us consider the total drag and lift coefficients, C_{D} and C_{L},

$$J(\mathbf{u}) = \int_{\Gamma_{\mathrm{W}}} (p\,\mathbf{n} - \underline{\tau}\,\mathbf{n}) \cdot \boldsymbol{\psi}\,\mathrm{d}s = \int_{\Gamma_{\mathrm{W}}} (p\,n_i - \tau_{ij} n_j)\,\psi_i\,\mathrm{d}s, \tag{11}$$

where $\tau = (\mu + \mu_t)\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^\top - \frac{2}{3}(\nabla\cdot\mathbf{v})I\right)$ presents the stress tensor at the wall including the viscous and turbulent viscosities μ and μ_t. Furthermore, $\boldsymbol{\psi}$ is given by $\boldsymbol{\psi}_d = \frac{1}{C_\infty}(\cos(\alpha), 0, \sin(\alpha))^\top$ or $\boldsymbol{\psi}_l = \frac{1}{C_\infty}(-\sin(\alpha), 0, \cos(\alpha))^\top$ for the drag and lift coefficient, respectively, and α is the angle of attack. Finally, $C_\infty = q_\infty A$, where $q = \frac{1}{2}\rho|\mathbf{v}|^2$ denotes the dynamic pressure and A denotes a reference area and subscripts $_\infty$ indicate freestream quantities.

Given that the numerical boundary fluxes $\hat{\mathbf{h}}_h|_\Gamma$ and $\underline{\hat{\sigma}}_h|_\Gamma$ are consistent, $J_h(\mathbf{u}_h)$ defined by

$$\begin{aligned} J_h(\mathbf{u}_h) &= \int_{\Gamma_{\mathrm{W}}} \Big(\hat{\mathbf{h}}_{\Gamma,h} - \underline{\hat{\sigma}}_{\Gamma,h}\mathbf{n}\Big) \cdot \tilde{\boldsymbol{\psi}}\,\mathrm{d}s \\ &= \int_{\Gamma_{\mathrm{W}}} \Big(\hat{\mathbf{h}}(\mathbf{u}^+_h, \mathbf{u}^-_\Gamma(\mathbf{u}^+_h), \mathbf{n}) - \{\!\{\widetilde{G}(\mathbf{u}_h)\}\!\}_\Gamma \nabla u^+_h + \tilde{\underline{\delta}}_\Gamma(\mathbf{u}^+_h)\Big) \cdot \tilde{\boldsymbol{\psi}}\,\mathrm{d}s. \end{aligned} \tag{12}$$

with $\tilde{\boldsymbol{\psi}} = (0, \psi_1, \psi_2, \psi_3, 0)^\top$, is a consistent discretization of the force coefficient $J(\mathbf{u})$ in (11). Finally, the discretization of the force coefficients according to (12) is required for a discretization of (1) to be adjoint consistent [14].

3.4 Evaluation of c_p- and c_f-values

Related to the evaluation of integral quantities (the force coefficients) is the evaluation of local quantities at the wall boundary like surface pressure and skin friction as involved in c_{p}- and c_{f}-distributions. The local c_{p}- and c_{f}-values at the wall boundary Γ_{W} are given by

$$c_{\mathrm{p}} = c_{\mathrm{p}}(\mathbf{u}) = \frac{p(\mathbf{u}) - p_\infty}{\frac{1}{2}\rho_\infty v^2_\infty}, \qquad c_{\mathrm{f}} = c_{\mathrm{f}}(\mathbf{u}, \nabla\mathbf{u}) = \frac{\tau_W(\mathbf{u}, \nabla\mathbf{u})}{\frac{1}{2}\rho_\infty v^2_\infty}, \tag{13}$$

where $\tau_W = -(\underline{\tau}\,\mathbf{n}) \cdot \mathbf{t}$ denotes the viscous stress at the wall, and $\underline{\tau} = \mu S$ is the viscous stress tensor defined in Section 3.3. Here, we recall that $\mathbf{n}$ is the unit outward normal vector to the boundary Γ of the (fluid dynamics) computational domain Ω; thus the vector $\mathbf{n}$ points into the

airfoil. Furthermore, $\mathbf{t}$ denotes the unit tangential vector to Γ parallel to the flow direction off the wall and directed such that $\mathbf{t} \cdot \mathbf{v}_\infty > 0$. Similar to evaluating the force coefficients (11) based on (12), the local c_{p}- and c_{f}-values (13) are evaluated based on (cf. [14])

$$c_{\mathrm{p},h} = c_{\mathrm{p},h}(\mathbf{u}_h) = \frac{\hat{\mathbf{h}}_{\Gamma,h} \cdot \tilde{\mathbf{n}} - p_\infty}{\frac{1}{2}\rho_\infty v_\infty^2}, \qquad c_{\mathrm{f},h} = c_{\mathrm{f},h}(\mathbf{u}_h, \nabla \mathbf{u}_h) = -\frac{\left(\hat{\underline{\sigma}}_{\Gamma,h}\mathbf{n}\right) \cdot \tilde{\mathbf{t}}}{\frac{1}{2}\rho_\infty v_\infty^2}, \tag{14}$$

with $\tilde{\mathbf{n}} = (0, n_1, n_2, n_3, 0)^\top$ for $\mathbf{n} = (n_1, n_2, n_3)^\top$, and $\tilde{\mathbf{t}} = (0, t_1, t_2, t_3, 0)^\top$ for $\mathbf{t} = (t_1, t_2, t_3)^\top$.

4 APPLICATION TO THE DLR-F11 HIGH LIFT CONFIGURATION

In the 4th International Workshop on High-Order CFD Methods (HioCFD-4), the DLR-F11 high lift configuration is considered a "computational/meshing challenge". This configuration has already been extensively analyzed for various configuration detail levels in the 2nd AIAA High Lift Prediction Workshop (HiLiftPW-2). In HioCFD-4, the so-called *Config 4* detail level is considered which includes slat tracks and flap track fairings but no slat pressure tube bundles. This configuration is considered at Mach number $M = 0.175$, Reynolds number $Re = 15.1 \cdot 10^6$ based on the mean aerodynamic chord (MAC) of 347.09mm, and angle of attack $\alpha = 7°$. These are the flow conditions of one of the high Reynolds number cases of Case 2b in HiLiftPW-2.

4.1 Linear and quadratic mesh generation process

For the DLR-F11 configuration under consideration, first a linear hybrid mesh has been generated using the CENTAUR grid generator. To reduce the number of surface faces, the curvature clustering was lowered to 6 compared to a typical value of 12-20. The curvature clustering represents the number of points used to model a full circle. To further reduce the number of surface faces, the faces on the wing and high lift surfaces were stretched in the spanwise direction by a factor of 3 near the leading and trailing edges. To properly model the boundary layer region, a 20-layer prismatic mesh was used with the initial layer thickness set to 0.003mm and the stretching set to 1.44. To decrease the number of tetrahedra generated, a tetrahedral stretching value of 2.15 was used. Sources were also used to locally add additional elements where needed to model complex geometric regions like the intersection between the slat and slat tracks.

With 2 365 919 prisms, 42 603 pyramids and 1 116 213 tetrahedra, in total 3 524 735 elements and 1 427 392 nodes, the linear mesh is rather coarse considering the complexity of the configuration. In a second step, also using the CENTAUR grid generator, the wall boundary faces of this mesh have been curved to create a quadratic, 3rd-order boundary approximation, and boundary as well as interior elements have been curved in order to avoid crossover of element faces. The high-order mesh contains the same number of elements and has 11 237 409 nodes. Figure 2 shows the quadratic mesh for the DLR-F11 configuration including curved surface faces on the fuselage. Figure 3 shows the surface mesh for both the linear and quadratic meshes in the region of the outboard flap track fairing. Note the curved surface elements on the trailing edge of the flap track fairing and its lower surface. Figure 4 shows a cut through both the linear and the quadratic hybrid meshes around the slat. The quadratic mesh clearly shows the curved prisms in the boundary layer. Both the first layer elements and elements farther away from the surface have been curved to ensure a good quality mesh in this region. Finally, Figure 5 shows a cut through the prism layers around the slat and leading edge of the main wing and their connection to the tetrahedral farfield mesh. Note the relatively smooth surface of the slat and main wing despite the relatively coarse surface mesh in this region of high curvature.

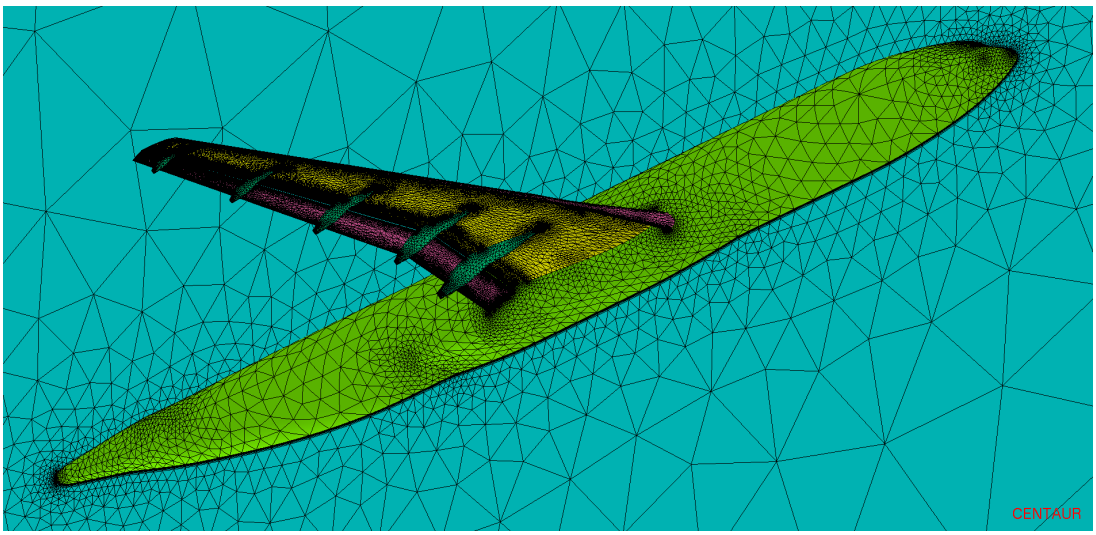
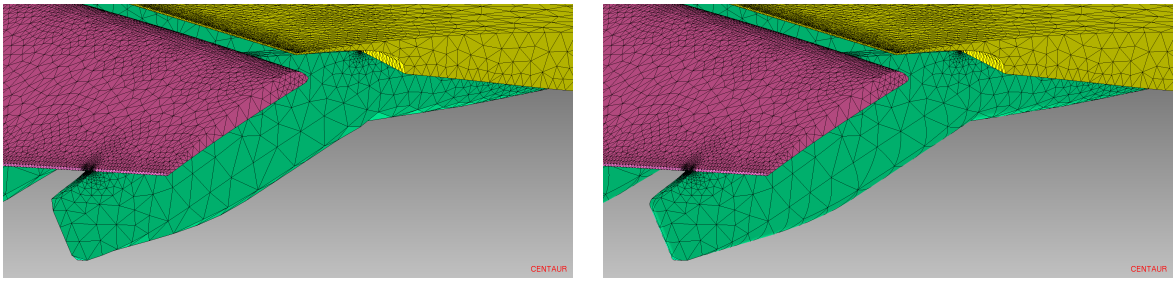
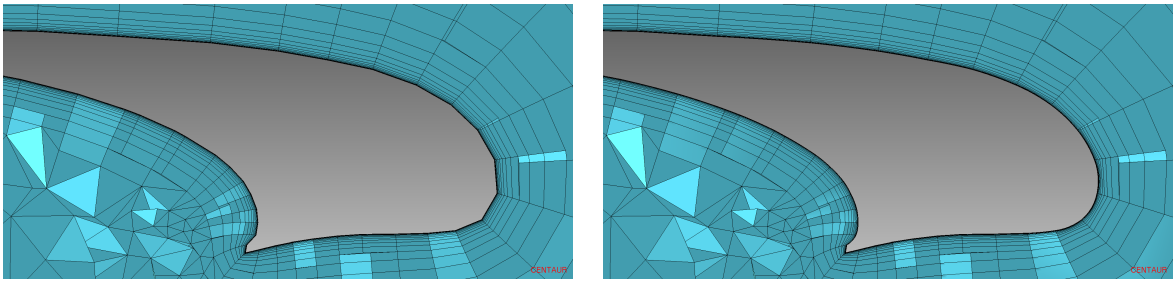


Figure 2: Quadratic mesh for the DLR-F11 configuration.



(a) (b)

Figure 3: Outboard flap track fairing surface mesh: (a) the linear mesh, and (b) the quadratic mesh.



(a) (b)

Figure 4: Hybrid mesh cuts in the slat region: (a) the linear mesh, and (b) the quadratic mesh.

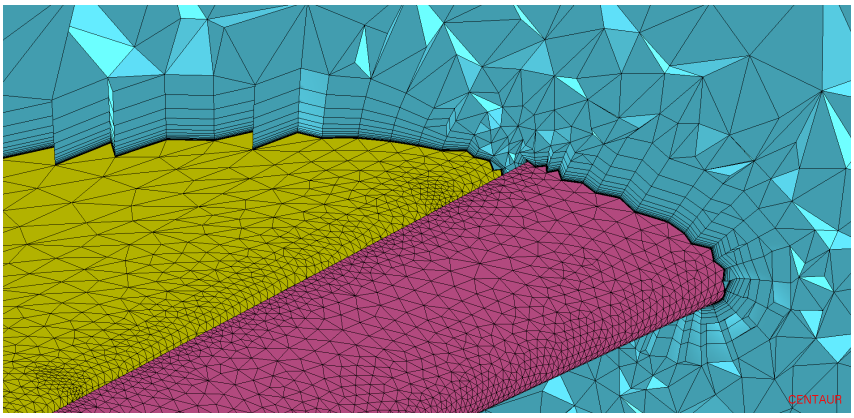


Figure 5: Cut throught the prism layers around the slat and the leading edge of the main wing.

4.2 Flow solution process

The DG solver PADGE [12] employs the discretization of the RANS and Wilcox-$k\omega$ equations as described in Section (3). The discrete equations are solved fully implicitly using a linearized Backward-Euler method. The CFL number of this pseudo-time iteration method starts with a low value and increases as the nonlinear residual decreases. Divergent iteration steps are (automatically) repeated with half the CFL number or even with a lower factor if required. As the solution continues to converge the CFL reduction factor is subsequently increased to unity again. This way, the solver can overcome difficult stages in the solution process like the transient phase but still benefits from (relatively) high CFL numbers in the final stage of the solution process. The linear systems of the implicit iteration scheme are solved using the GMRES method and the ILU(0) preconditioner as offered by the PETSc library [2] which runs within a block-Jacobi iteration in parallel. The Jacobian matrices required in the implicit scheme are hand-differentiated and complete in the sense that the derivatives of all terms are included and all equations are fully coupled.

On the mesh generated as described in Sections 2 and 4.1 we first solve the DG discretization (2) for the polynomial degree $p = 0$, and subsequently for $p = 1$ and $p = 2$. The solver for the $p = 0$ solution starts from freestream values, the solver for each of the $p > 0$ solutions starts from the converged $p-1$ solution. With this p-sequencing the solver for each polynomial degree has a sufficiently good initial guess to start the solution process with. We choose the number of quadrature points per elements depending on the polynomial degree. For a polynomial degree p we use $p+2$ Gaussian quadrature points in each dimension, i. e., $(p+2)^3$ per element. Before the PADGE solver starts solving it checks the regularity of the mesh by evaluating the Jacobian determinants[1] at the quadrature points for the particular polynomial degree. As the degree increases the number of quadrature points per element increases and the coverage of the element by checking the determinants in quadrature points is increased. Thus, a curved mesh which might occur "discretely regular" for one specific degree (and thus quadrature formula) might turn out to be irregular if checked at the quadrature points involved for a higher polynomial degree.

degree	# DoFs/eqn	# quadrature points/elem.	# irregular elements	worst # neg. Jac./element
0	3 524 735	8	0	0/8
1	14 098 940	27	0	0/27
2	35 247 350	64	14	2/64

Table 1: PADGE's "discrete regularity" check of the mesh (with 3 524 735 elements) for $p = 0$, $p = 1$, and $p = 2$.

The related data is collected in Table 1 for $p = 0$, $p = 1$, and $p = 2$. For each of the polynomial degrees it includes the number of degrees of freedoms per equation, the number of quadrature points per element, the number of irregular elements when checking the Jacobian determinants in the quadrature points, and finally the worst/highest number of quadrature points per element at which the Jacobian determinant is negative. Here, we see that the mesh is "discretely regular" when checking in the quadrature points of the discretization of polynomial degrees $p = 0$ and $p = 1$ as in each of the quadrature points involved the Jacobian determinant

[1] Jacobian determinant: The determinant of the derivative of the mapping function from the reference element to the element in physical space.

is positive. Furthermore, we see that for $p = 2$ there are 14 irregular elements. However, in these elements the Jacobian determinants are negative in at most 2 out of 64 quadrature points. Depending on the sensitivity of the discretization to irregular elements this "slightly irregular" mesh might be acceptable to the flow solver or not.

On this mesh we computed $p = 0$, $p = 1$ and $p = 2$ (i. e., 1^{st}-, 2^{nd}-, and 3^{rd}-order) flow solutions. For each of the polynomial degrees the (l2-norm of the) nonlinear residual (vector) is reduced by a factor of 10^{-10} relative to the freestream residual for that polynomial degree. Then the flow solution is transferred to (embedded into) the function space of the next higher polynomial degree, and the solution process is continued. Figure 6 plots the convergence history

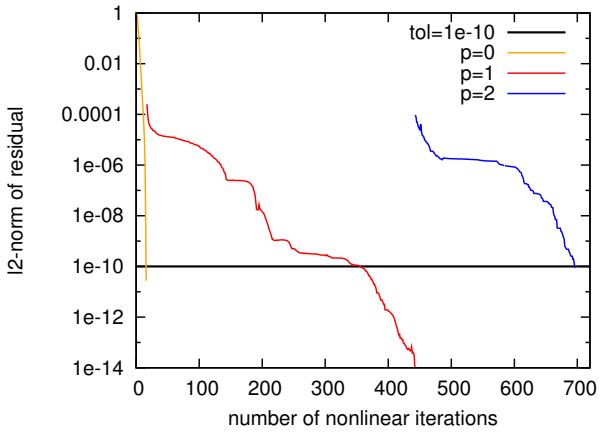


Figure 6: DLR-F11, Config 4: $M = 0.175$, $Re = 15.1 \cdot 10^6$, $\alpha = 7°$: Solver convergence of the nonlinear residual vs. the number of fully implicit iteration steps for the $p = 0$, $p = 1$, and $p = 2$ flow solutions.

of the (l2-norm of the) nonlinear residual (vector) normalized by the freestream residual for each polynomial degree. The $p = 0$ solution for which several of the physical and discretization terms vanish is converged in a very small number (17) of iterations steps. The $p = 1$ solver does not start from freestream (corresponding to a normalized residual of one) but from a lower residual as it uses the $p = 0$ solution as initial solution. The $p = 1$ solution requires a relatively large number (339) of iteration steps. This is a typical effect in p-sequencing on fine meshes where the $p = 1$ solver starts with a rather bad $p = 0$ solution as initial guess. In the convergence history of the $p = 1$ solution there are several stages where the convergence rate is reduced to almost zero. Here, the solver experienced problems and decided to recompute some iteration steps with a decreased CFL number. Given that in this computation 45 out of the 339 iteration steps diverged/failed and needed to be recomputed the initial CFL number and the settings of its subsequent increase seemed to be chosen too high. Even close to the 10^{-10} convergence criterion the solver experienced problems which is why the computation had been restarted to check whether the nonlinear residual would converge below 10^{-14}, and it actually did after 87 further steps 18 of which were recomputed. Overall, the CFL number in the $p = 1$ solution process started with 5 and increased to 500 until residual 10^{-10} and over 40 000 until residual 10^{-14}. Finally, the $p = 2$ solver took the $p = 1$ solution as initial guess and converged the nonlinear residual below the prescribed 10^{-10} tolerance. Here, only 24 out of 254 nonlinear iteration steps failed and needed to be recomputed with a decreased CFL number. Here, the CFL number started at 10 and increased up to 200.

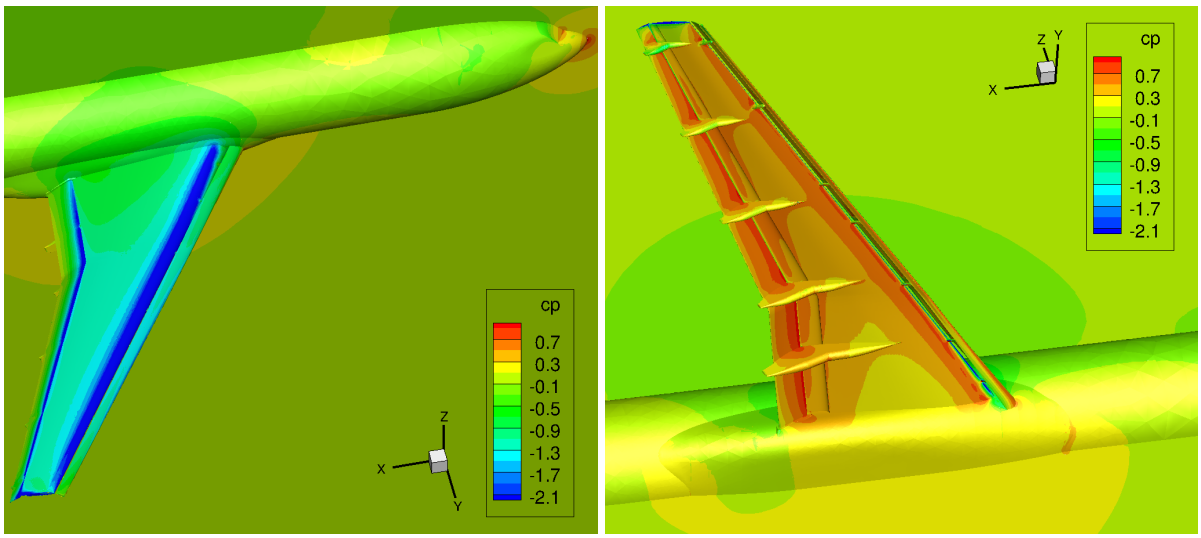
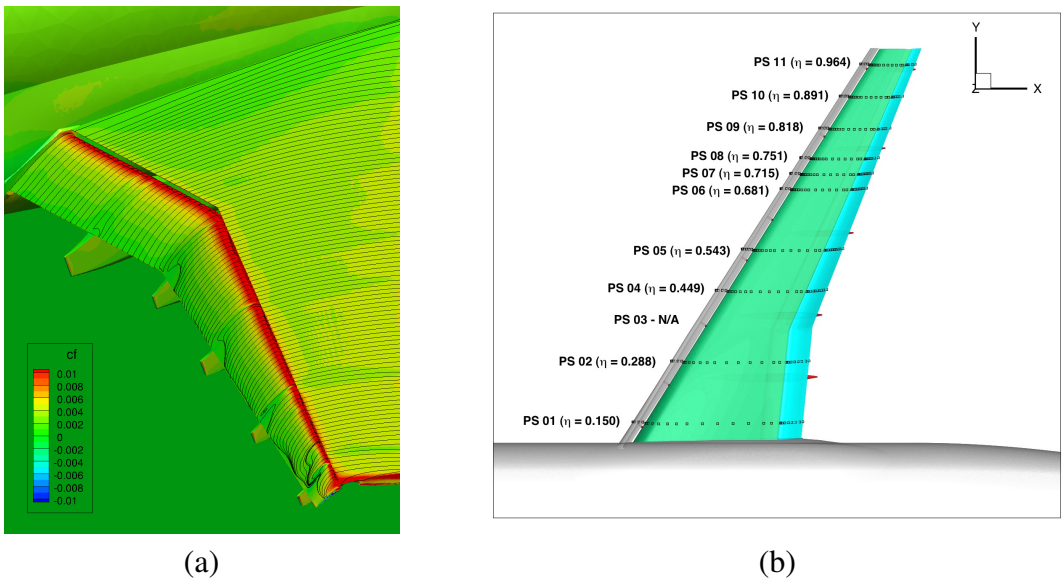


Figure 7: DLR-F11, Config 4: $M = 0.175$, $Re = 15.1 \cdot 10^6$, $\alpha = 7°$: c_p-distributions of the 3[rd]-order DG solution viewed from above and below the configuration.



(a) (b)

Figure 8: DLR-F11, Config 4: $M = 0.175$, $Re = 15.1 \cdot 10^6$, $\alpha = 7°$: (a) c_f-distribution and streamlines of the 3[rd]-order DG solution on the main wing and flap. (b) Pressure tap locations in the experiment (figure from [1]).

The c_p-distribution of the resulting $p = 2$ flow solution is shown in Figure 7. Here, we recognize the complexity of the flow which passes the slat, the slat tracks, the upper wing or the lower wing with the flap track fairings, and the flap. Furthermore, Figure 8(a) shows the c_f-distribution and $\mathbf{c}_f$-streamlines zoomed in on the flap and parts of the main wing. Here, we see that the flow stays attached on the main wing as well as on most of the flap. However, close to the flap track fairings (located below the flap) there are small separation regions on the upper side of the flap. Furthermore, there is a larger separation region on the flap near its tip.

Figure 8(b) shows the location of the pressure taps in the experimental setup. Slices of the c_p-distribution at the pressure tap locations PS1, 4, 6, and 10 (at the y-positions $\eta = 0.150$, 0.449, 0.681, and 0.891, respectively) are shown in Figure 9 compared to the experimental data at similar flow conditions. Here, we see that the c_p-distribution matches very well that of the experiments, not only on the main wing but also on the flap and the slat.

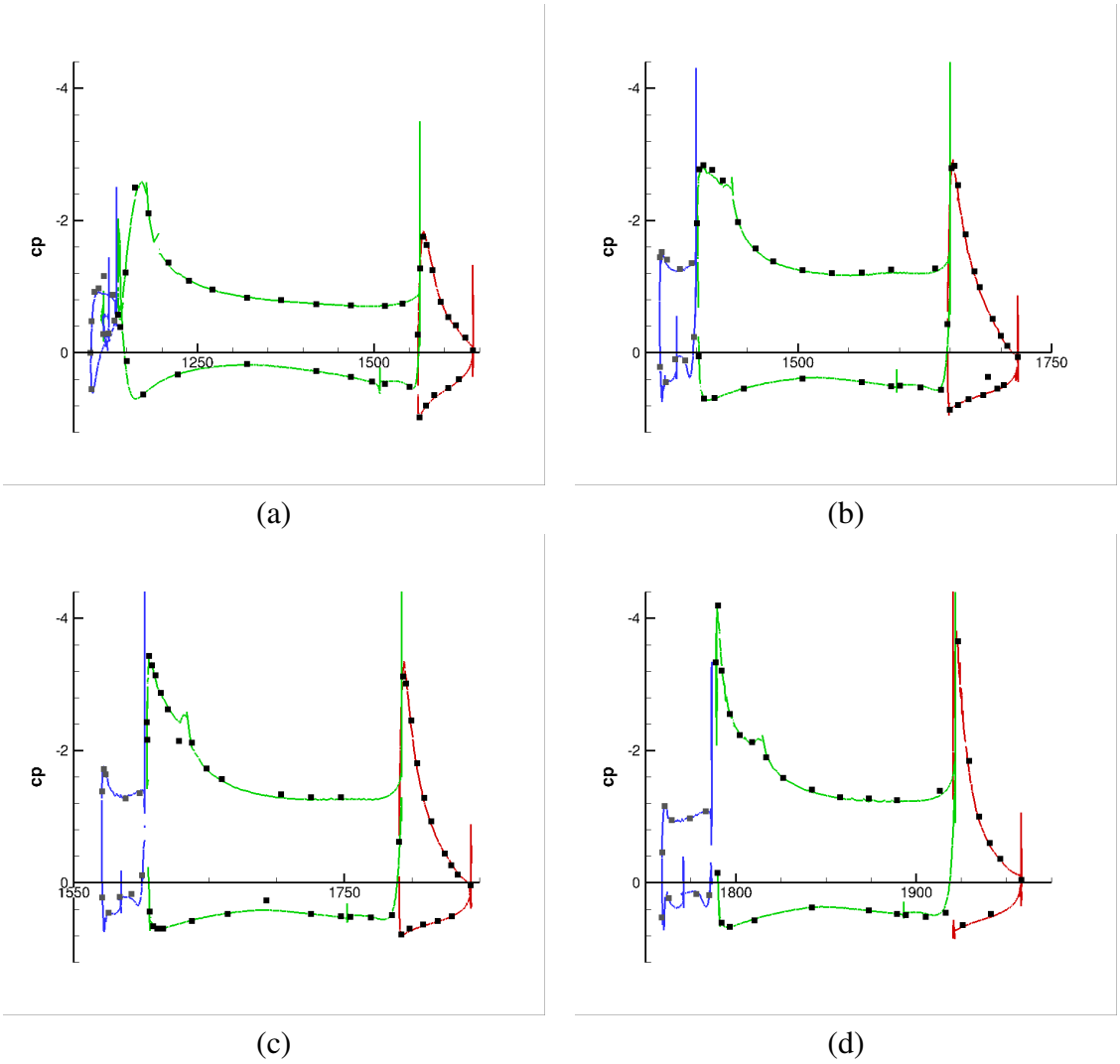


Figure 9: DLR-F11, Config 4: $M = 0.175$, $Re = 15.1 \cdot 10^6$, $\alpha = 7°$: c_p-distributions of the 3rd-order DG solution compared to experimental data (at $M = 0.176$, $Re = 15.1 \cdot 10^6$, $\alpha = 7.03938$) [1]. c_p-slices at pressure tap locations (a) PS01 at $\eta = 0.150$, (b) PS04 at $\eta = 0.449$, (c) PS6 at $\eta = 0.681$, and (d) PS10 at $\eta = 0.891$.

Finally, Table 2 collects the integrated forces. Here we see that the force coefficient values for the lift, drag and moment coefficients, C_L, C_D, and C_M, are quite close to the experimental data. Furthermore, they are in a range similar to that obtained in RANS-SA computations during the HiLiftPW-2, with a slightly larger deviation in the moment but closer to the experiments in the drag coefficient.

5 CONCLUSIONS AND OUTLOOK

The computational/meshing challenge of the DLR-F11 configuration which is "close to industrial needs" was considered to improve and challenge current high-order mesh generation and solver techniques. For this complex geometry, a quadratic curved mesh has been generated using the CENTAUR grid generator [5]. On this 3rd-order mesh a fully turbulent 3rd-order DG solution to the RANS and Wilcox-$k\omega$ turbulence equations has been obtained using the fully implicit DLR-PADGE solver [12]. The numerical results compare very well to experiments, with particularly well matching c_p-distributions, and good approximations were obtained to global

	C_L	C_D	C_M
Exp.	1.9270	0.1615	-0.5390
TAU*	1.8794	0.1681	-0.5647
	-2.5%	+4.1%	-4.8%
PADGE	1.8781	0.1649	-0.5704
	-2.5%	+2.1%	-5.8%

Table 2: DLR-F11, Config 4: $M = 0.175$, $Re = 15.1 \cdot 10^6$, $\alpha = 7°$: Force coefficients of the $p = 2$ flow solution compared to HiLiftPW-2 results (* Rudnik, Melber-Wilkending (DLR), RANS-SAO, 2013) and experiments [1].

force coefficients with deviations in the range of previous HiLiftPW-2 results.

This work can be considered as one of possibly many steps towards offering a complete and automated chain of high-order grid generation and high-order aerodynamic CFD solver technology. At the beginning of this work curved grids were generated for this complex geometry which included a large number (> 100) of heavily irregular elements. In contrast to that the grid on which a 3$^{\text{rd}}$-order solution could finally be computed had only a few (in the range of 10) elements left which were "slightly irregular" only. Given this it is clear that the high-order grid generation took a significant step forward. Nevertheless, further steps are required in this direction. First of all, the final goal would be that all elements in a high-order mesh are regular; in a first step not necessarily "regular" as can be analyzed with a validity check based on Bézier functions [17], but "discretely regular" in the sense that the Jacobian determinants are positive at least in each of the quadrature points involved in a (quadrature-based) discretization of a specific polynomial degree. Secondly, meshes which are generated to be used with high-order methods, and which will be taken as starting point for local mesh (or hp-)refinement [18, 21] should be as coarse as possible but clearly still sufficiently fine to capture the geometry. On this fine line of ostensibly conflicting requirements (coarse but still sufficiently fine) it is particular important that mesh generation capabilities are stable and deliver high quality elements even in case of particularly coarse meshes. Finally, curved grid generation should not stop at quadratic boundary approximation. The difference between using quadratic curved meshes as compared to using linear (straight-sided) meshes is quite large and can be expected to be larger than the difference between e. g., a quadratic and a cubic boundary approximation. Nevertheless, a boundary approximation higher than quadratic – even though introducing additional complexity in the grid generation process – would increase the quality of the boundary approximation in case of a fixed mesh size or would allow coarser meshes with a similar boundary approximation quality.

No doubt, the complexity of the configuration considered stressed the PADGE solver [12] close to its current limits. Using a discretization on the boundary which resembles the discretization employed on interior faces as close as possible, and thus introducing a level of numerical diffusion which is equal/similar throughout the domain until (and including) the boundary, is – to our experience [15] – an essential ingredient to stabilize flow computations on geometries of this complexity. With an appropriate discretization of force coefficients the particular discretization employed on the boundary does not break adjoint consistency [14], in which case corresponding discrete adjoint solutions can be expected to be smooth. Furthermore, evaluating local quantities like surface pressure and skin-friction at the wall accordingly, allows to obtain accurate and smooth c_p- and c_f-distributions [14] even in our case of using numerical flux functions at the wall boundary. A fully implicit solver has been used, with solver recovery techniques enabled which recompute divergent/failed steps with a lower CFL number. Further

development of the high-order discretization and solvers are still required. While PADGE can be considered a research code, not particularly trimmed to efficiency or scalability, high-order DG methods are currently being implemented alongside cell-centered 2nd-order Finite Volume methods into the new DLR flow solver Flucs dedicated (amongst others) to future industrial use. Significant effort will be required to make high-order DG methods readily available to industry for configurations of this complexity, at cruise as well as at high lift conditions.

REFERENCES

[1] 2nd AIAA CFD High Lift Prediction workshop. http://hiliftpw.larc.nasa.gov/index-workshop2.html.

[2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[3] 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*, 34:507–540, 2005.

[4] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Num. Meth. Part. Diff. Eq.*, 16(4):365–378, 2000.

[5] CentaurSoft. CENTAUR software. Computational grids for Advanced Modelling. www.centaursoft.com.

[6] M. Ceze and K. Fidkowski. Drag prediction using adaptive discontinuous finite elements. 51st AIAA Aerospace Sciences Meeting, Jan. 2013, AIAA 2013-0051, 2013.

[7] C. Geuzaine, A. Johnen, J. Lambrechts, J.-F. Remacle, and T. Toulorge. The generation of curvilinear meshes. In H. Deconinck and R. Abgrall, editors, *VKI LS 2014-03: 37th Advanced CFD Lecture Series: Recent developments in higher order methods and industrial application in aeronautics, Dec. 9-12, 2013*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2014.

[8] A. Harten and J. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 50(2):416–433, 1983.

[9] R. Hartmann. Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. *Int. J. Numer. Meth. Fluids*, 72(8):883–894, 2013.

[10] R. Hartmann. Results for the C3.5 test case with the DLR-PADGE code. 3rd International Worskhop on High-Order CFD methods, Kissimmee, Florida, Jan 3-4, 2015. https://www.grc.nasa.gov/hiocfd/wp-content/uploads/sites/22/C3.5_DLR.pdf, 2015.

[11] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations. *J. Comput. Phys.*, 230(11):4268–4284, 2011.

[12] R. Hartmann, J. Held, T. Leicht, and F. Prill. Discontinuous Galerkin methods for computational aerodynamics – 3D adaptive flow simulation with the DLR PADGE code. *Aerosp. Sci. Technol.*, 14:512–519, 2010.

[13] R. Hartmann and T. Leicht. Higher order and adaptive DG methods for compressible flows. In H. Deconinck and R. Abgrall, editors, *VKI LS 2014-03: 37th Advanced CFD Lecture Series: Recent developments in higher order methods and industrial application in aeronautics, Dec. 9-12, 2013*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2014.

[14] R. Hartmann and T. Leicht. Generalized adjoint consistent treatment of wall boundary conditions for compressible flows. *J. Comput. Phys.*, 300:754–778, 2015.

[15] R. Hartmann and T. Leicht. Generation of unstructured curvilinear grids and high-order discontinuous Galerkin discretization applied to a 3D high-lift configuration. *Int. J. Num. Meth. Fluids*, 2016. published online. DOI: 10.1002/fld.4219.

[16] 4th International Worskhop on High-Order CFD methods, Heraklion (Crete), Greece, June 4-5, 2016. http://how4.cenaero.be/.

[17] A. Johnen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. *J. Comput. Phys.*, 233:359–272, 2013.

[18] Y. Kallinderis, A. Karkoulias, and P. Antonellis. Adaptive high-order hybrid grids for field computations. In *8th Int. Congress on Computational Mechanics, 12-15 July 2015, Volos, Greece*, 2015.

[19] N. Kroll, H. Bieler, H. Deconinck, V. Couallier, H. van der Ven, and K. Sorensen, editors. *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer, 2010.

[20] N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, editors. *IDIHOM - Industrialization of High-Order Methods - A Top Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer, 2015.

[21] T. Leicht and R. Hartmann. Error estimation and hp–adaptive mesh refinement for discontinuous Galerkin methods. In Z. J. Wang, editor, *Adaptive High-Order Methods in Computational Fluid Dynamics*, volume 2 of *Advances in Computational Fluid Dynamics*, chapter 3, pages 67–94. World Science Books, 2011.

[22] O. Sahni, X. Luo, K. Jansen, and M. Shephard. Curved boundary layer meshing for adaptive viscous flow simulations. *Finite Elements in Analysis and Design*, 46(1-2):132–139, 2010.

[23] S. Schoenawa and R. Hartmann. Discontinuous Galerkin discretization of the Reynolds-averaged Navier-Stokes equations with the shear-stress transport model. *J. Comput. Phys.*, 262:194–216, 2014.

[24] M. Wallraff, T. Leicht, and M. Lange-Hegermann. Numerical flux functions for Reynolds-averaged Navier-Stokes and $k\omega$ turbulence model computations with a line-preconditioned p-multigrid discontinuous Galerkin solver. *Int. J. Numer. Meth. Fluids*, 71(8):1055–1072, 2013.

[25] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *Int. J. Num. Meth. Fluids*, 72(8):811–845, 2013.

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

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

[28] Z. Q. Xie, R. Sevilla, O. Hassan, and K. Morgan. The generation of arbitrary order curved meshes for 3D finite element analysis. *Computational Mechanics*, 51(3):361–374, 2013.