European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

ON THE APPLICATION OF THE DISCONTINUOUS GALERKIN METHOD TO TURBOMACHINERY FLOWS

Svetlana Cherednichenko[1], **Christian Frey**[1], **and Graham Ashcroft**[1]

[1] German Aerospace Center (DLR), Institute of Propulsion Technology
Linder Höhe, 51147 Cologne, Germany
e-mail: {svetlana.cherednichenko,christian.frey,graham.ashcroft}@dlr.de

Keywords: Discontinuous Galerkin, high-order spatial discretization, non-reflecting boundary conditions, unstructured meshes

Abstract. *To meet future challenges in turbomachinery CFD highly efficient numerical methods capable of handling intricate geometries and resolving complex flow features are required. One approach to this problem is to employ high-order discretization methods in time and space. For spatial discretization the Discontinuous Galerkin (DG) method is particularly attractive as it delivers high order accuracy in the context of unstructured meshes.*

To fully exploit the advantages of the basic DG method it is important to develop accurate boundary conditions on both the physical and artificial boundaries of the computational domains. In this work we present a DG solver for 2D Euler flows with special emphasis on the implementation of robust and accurate boundary conditions for turbomachinery applications. The DG solver supports triangular and quadrilateral elements. The spatial discretization is based on Roe's numerical fluxes and polynomial basis functions of degree $p \leq 3$*, i.e., up to fourth order accuracy. Both explicit and implicit time integration schemes are implemented for unsteady simulations. Steady solutions are obtained using an implicit pseudo-time marching technique.*

1 Introduction

Global air traffic is forecast to grow significantly in the coming years. To facilitate this development, particularly in the context of ambitious environmental goals concerning the reduction of emissions and noise pollution, there is a great need for new and more efficient engine technologies. To accurately investigate such technologies, and at the same time realise the shorter development cycles demanded by customers, more efficient, flexible and accurate numerical tools for the simulation of turbomachinery flows are essential.

In this context, high-order accurate numerical methods suitable for application on unstructured grids represent an attractive solution. One of the most popular high-order methods for unstructured grids is the Discontinuous Galerkin (DG) finite element method. The DG method may be considered as a combination of the Finite Volume (FV) and Finite Element (FE) methods in which the solution is approximated by means of piece-wise polynomials inside the elements and, in contrast to standard FE methods, continuity between adjacent elements is not required.

The DG method was first introduced in [1] for the neutron transport equation. Cockburn and Shu extended the spatial DG discretization to hyperbolic non-linear conservation laws. Moreover, the authors investigated an explicit high-order Runge-Kutta time discretization in combination with DG [2, 3]. Implicit time discretization for the Euler equations is employed in [4, 5, 6]. The discretization of the two-dimensional Euler equations and the high-order treatment of curved boundaries are discussed by Bassi and Rebay in [7]. Here, it is shown that solution accuracy degrades near the boundaries if straight edges are used. Moreover, the local refinement of the mesh does not improve the accuracy near the boundaries. An overview of convection-dominated problems concerning theoretical and computational aspects of DG methods appears in [8, 9]. Progress of high-order DG methods in the context of aerospace applications is discussed in [10].

The important feature of numerical methods for turbomachinery applications remains the construction of the accurate absorbing boundary conditions. The inlet and outlet boundaries of the blade rows are typically close to the near fields of the blades, hence numerical reflections may affect the solutions considerably. The incorporation of standard one-dimensional non-reflecting conditions based on characteristic theory into DG codes is studied in [11, 12].

In this work, the two-dimensional non-reflecting boundary conditions, which are presented by Giles in [13] are integrated into the DG framework. These are formulated in terms of the spatial Fourier coefficients of the flow field along the Gauss points on the boundary edges. For this purpose, a discrete Fourier transform for non-equidistant point distributions is implemented. The application of these boundary conditions to the flow through a cascade of periodically arranged blades demonstrate superior results to the one-dimensional absorbing boundary conditions.

The paper is organized as follows. In Section 2 the governing equations are presented. Furthermore, the DG method for the spatial discretization and the time discretization schemes are introduced. The solid wall, Riemann, non-reflecting boundary conditions and the treatment of curved boundaries are described in Section 3. Numerical results confirming the relevance of the high-order spatial discretization schemes, curved boundaries and their application to a turbomachinery test case are presented in Section 4. Finally the results are summarized in Section 5.

2 Description of the Flow Solver

Two-dimensional compressible inviscid flow is described by the Euler equations. We solve the Euler equations in conservative form i.e.,

$$\frac{\partial q}{\partial t} + \nabla \cdot F(q) = 0 \qquad \text{in } \Omega \times (0,T) \tag{1}$$

with some initial and boundary conditions. $T > 0$ represents the final time and $\Omega \subset \mathbb{R}^2$ is a bounded domain. The Cartesian components of the conservative state vector q and flux vectors $\langle F(q), \vec{n}\rangle = F^{\vec{n}}(q)$ are

$$q = \begin{pmatrix} \rho \\ \rho U \\ \rho E \end{pmatrix}, \qquad F^{\vec{n}}(q) = \begin{pmatrix} \rho\langle U, \vec{n}\rangle \\ \rho\langle U, \vec{n}\rangle + p\vec{n} \\ \rho\langle U, \vec{n}\rangle H \end{pmatrix},$$

where $\vec{n} = (n_x, n_y)$ is the outward unit normal to the boundary, ρ is the density, E is the total energy and $U = (u, v)$ are the velocity components in the x and y directions, respectively. H is the total enthalpy per unit mass and is defined as $H = \frac{\gamma}{(\gamma-1)}\frac{p}{\rho} + \frac{1}{2}\|U\|^2$, where $\gamma = 1.4$ is the ratio of specific heats and p is the pressure. The pressure is related to the other thermodynamic variables by the equation of state for an ideal gas $p = (\gamma - 1)\rho[E - \frac{1}{2}\|U\|^2]$.

2.1 The Discontinuous Galerkin Scheme

In this section in order to provide necessary notations the DG discretization method is summarized. The weak formulation of equation (1) follows in a standard way, cf. [9].

Suppose, the domain Ω is a collection of arbitrary non-overlapping elements Ω_i, such that $\Omega = \cup_{i=1,\dots,N_{\text{el}}}\Omega_i$, where N_{el} denotes the number of 2D elements in the domain. First of all, the construction of a discrete space will be outlined. The DG discretization uses polynomial bases which are discontinuous across the elements. Therefore, the space of piece-wise polynomials is defined as,

$$V_h = \{v_h \in L^2(\Omega) \;\big|\; v_{h|_{\Omega_i}} \in \mathcal{P}^p(\Omega_i) \quad \text{for all } i = 1, \dots, N_{\text{el}}\},$$

where $\mathcal{P}^p(\Omega_i)$ denotes the set of polynomials of degree up to p defined on the cell Ω_i and h stands for a discretization parameter.

The weak formulation is obtained by multiplying equation (1) by a sufficiently smooth test function v and performing integration by parts

$$\int_{\Omega_i} v\frac{\partial q(x,t)}{\partial t}\, d\Omega_i + \oint_{\partial\Omega_i} vF^{\vec{n}}(q(x,t))\, d\sigma - \int_{\Omega_i} F^{\nabla_x v}(q(x,t))\, d\Omega_i = 0. \tag{2}$$

Further, in order to find an approximation of $q(x,t)$ such that, for each time $t \in [0,T]$, the discrete function $q_h(x,t)$ belongs to V_h, where $x = (x_1, x_2)$, we substitute $q(x,t)$ and v by their polynomial expansions.

Let $\phi_1(x), \dots, \phi_{n(p)}(x)$ be a basis of $\mathcal{P}^p(\Omega_i)$. Therefore, a function $v_h \in \mathcal{P}^p(\Omega_i)$ has the representation

$$v_{h,i}(x) = \sum_{k=1}^{n(p)} v_{i,k}\phi_{i,k}(x),$$

i.e., each test function $v_h \in \mathcal{P}^p(\Omega_i)$ can be written as a linear combination of the basis functions $\phi_{i,k}(x)$. The number of degrees of freedom for the polynomial basis of the order p in every

element is denoted by $n(p)$. Since $q_h(x,t)$ belongs to the same subspace V_h with the basis $\phi_1(x), \ldots, \phi_{n(p)}(x)$, it satisfies the following

$$q_{h,i}(x,t) = \sum_{l=1}^{n(p)} q_{i,l}(t)\phi_{i,l}(x). \tag{3}$$

Now the basic discrete form of the DG approach can be described

$$\int_{\Omega_i} v_h \frac{\partial q_h(x,t)}{\partial t}\, d\Omega_i + \sum_{e=e_{i,j}\in\partial\Omega_i} \oint_e v_h F^{\vec{n}_e}_{\text{num}}(q_{h,i}(x,t), q_{h,j}(x,t))\, d\sigma_e \\ - \int_{\Omega_i} F^{\nabla_x v_h}(q_h(x,t))\, d\Omega_i = 0. \tag{4}$$

The flux $F^{\vec{n}_e}(q(x,t))$ in (2) is replaced by a numerical flux function $F^{\vec{n}_e}_{\text{num}}(q_{h,i}(x,t), q_{h,j}(x,t))$, which depends on the outward unit vector w. r. t. the edge $\vec{n}_e$, internal interface state $q_{h,i}$ and on the neighbouring element interface state $q_{h,j}$. For the current DG code the Roe numerical flux is employed. The edges of the element Ω_i in (4) are denoted by $\partial\Omega_i$.

Replacing $q_h(x,t)$ in (4) by (3) and testing (4) only against the basis functions $\phi_{i,k}(x)$ for all $k = 1, \ldots, n(p)$, yields

$$\frac{\partial q_{i,l}(t)}{\partial t} \int_{\Omega_i} \phi_{i,k}(x)\phi_{i,l}(x)\, d\Omega_i + \sum_{e=e_{i,j}\in\mathcal{E}_i} \oint_e \phi_{i,k}(x) F^{\vec{n}_e}_{\text{num}}(q_{h,i}(x,t), q_{h,j}(x,t))\, d\sigma_e \\ - \int_{\Omega_i} F^{\nabla_x \phi_{i,k}}(q_{h,i}(x,t))\, d\Omega_i = 0. \tag{5}$$

Substituting integrals of the mass matrix, numerical flux and the stiffness term by M, H and S, respectively, equation (5) can be rewritten in the compact form

$$\frac{\partial q_{i,l}(t)}{\partial t} + R^i_l = 0$$

with $R^i_l = \tilde{M}^i_{lk}\big(H^i_k - S^i_k\big)$ and $\tilde{M}^i_{lk} M^i_{kl} = I$. The surface integrals in M^i_{kl} and S^i_k and the line integrals H^i_k are carried out using numerical Gauss quadrature formulas of appropriate accuracy order, cf. [8], in the following form

$$H^i_k = \sum_{e=e_{i,j}\in\partial\Omega_i} \sum_{x\in\mathcal{G}_e} \omega_{e,x}\phi_{i,k}(x) F^{\vec{n}_e}_{\text{num}}(q_{h,i}(x), q_{h,j}(x))|e| \tag{6}$$

$$S^i_k = \sum_{x\in\mathcal{G}_{\Omega_i}} \omega_{\Omega_i,x} F^{\nabla_x \phi_{i,k}(x)}(q_{h,i}(x))|\Omega_i|, \tag{7}$$

where the set of one-dimensional Gauss integration points is denoted by $\mathcal{G}_e$ and the set of two-dimensional Gauss integration points is denoted by $\mathcal{G}_{\Omega_i}$. The one- and two-dimensional integration weights are $\omega_{e,x}$ and $\omega_{\Omega_i,x}$, respectively.

2.2 Time Integration Methods

The spatial discretization leads to a semi-discrete system of ordinary differential equations in time

$$\frac{\partial q}{\partial t} + R(q) = 0. \tag{8}$$

Explicit and implicit time integration schemes have been implemented to solve (8).

2.2.1 Explicit Time Integration Methods

The following three-stage third order accurate Runge-Kutta method is used to integrate (8):

$$\begin{aligned} q^{(1)} &= q^n + \Delta t R(q^n, t^n) \\ q^{(2)} &= \frac{3}{4}q^n + \frac{1}{4}\Big(q^{(1)} + \Delta t R(q^{(1)}, t^n + \Delta t)\Big) \\ q^{n+1} &= \frac{1}{3}q^n + \frac{2}{3}\Big(q^{(2)} + \Delta t R(q^{(2)}, t^n + \frac{1}{2}\Delta t)\Big) \end{aligned} \tag{9}$$

Applied to the DG system with elements of order p this method has been shown to be linearly stable for a Courant number less than or equal to $\frac{1}{2p+1}$, cf. [9]. In this work the explicit solver has been used for unsteady simulations only.

2.2.2 Implicit Time Integration Methods

As explicit time-integration schemes are only conditionally stable their application to problems of engineering interest typically requires the use of restrictively small time-steps, making such methods inefficient. For stiff numerical problems, implicit time-integration schemes, being unconditionally stable, are therefore often preferred although the computational overhead per time-step is significantly higher than with explicit methods. Therefore, in this work implicit time-integration schemes have been implemented for steady simulations. The application of an implicit time-integration method to (8) results in a complex non-linear system that has to be solved at each iteration. In particular, implicit time-integration schemes require the evaluation of the residual at the unknown time level $n+1$

$$\frac{\delta q^n}{\Delta t} + R(q^{n+1}) = 0. \tag{10}$$

The equation (10) can be approximated using the first-order time accurate Backward Differentiation Formulas method, namely BDF1 or the second-order time accurate BDF2:

$$\begin{aligned} \frac{1}{\Delta t}(q^{n+1} - q^n) + R(q^{n+1}) = 0, &\qquad \text{BDF1} \\ \frac{1}{\Delta t}(\tfrac{3}{2}q^{n+1} - 2q^n + \tfrac{1}{2}q^{n-1}) + R(q^{n+1}) = 0. &\qquad \text{BDF2} \end{aligned} \tag{11}$$

In this work BDF1 will be used for computations, since the first-order scheme is more robust and time accuracy is irrelevant for steady problems [14]. The general implicit formulation leads to the nonlinear unsteady residual

$$\tilde{R}(q^{n+1}) = 0. \tag{12}$$

Observe that this system must be solved at each time step. Therefore, the same implicit solution techniques as for the steady equation $R(q^{n+1}) = 0$ is applied. One of the approaches to solve the system of equations resulting from the implicit formulations (11) is a pseudo-time marching method. Therefore, (12) can be rewritten as

$$\frac{1}{\Delta \tau}\delta q_m^{n+1} + \tilde{R}(q_{m+1}^{n+1}) = 0$$

with $\delta q_m^{n+1} = q_{m+1}^{n+1} - q_m^{n+1}$, where q_{m+1}^{n+1} approximates q^{n+1} for $m = 0, \ldots, M-1$ and $\Delta\tau$ denotes a pseudo-time step. The residual $\tilde{R}(q_{m+1}^{n+1})$ is linearized as follows

$$\tilde{R}(q_{m+1}^{n+1}) \approx \tilde{R}(q_m^{n+1}) + \left.\frac{\partial \tilde{R}}{\partial q}\right|_{q_m^{n+1}} \delta q_m^{n+1},$$

where $\frac{\partial \tilde{R}}{\partial q}$ is referred to as the Jacobian matrix of the system. The dual-time formulation is written as follows

$$\left(\alpha + \left.\frac{\partial R}{\partial q}\right|_{q_m^{n+1}}\right)\delta q_m^{n+1} = -\tilde{R}(q_m^{n+1}). \tag{13}$$

In the equation above α is represented by $\frac{1}{\Delta t} + \frac{1}{\Delta\tau}$ for BDF1 and $\alpha = \frac{1}{\Delta t} + \frac{3}{2}\frac{1}{\Delta\tau}$ for BDF2. The approximate solution to (13) is obtained by one iteration of the Symmetric Gauss-Seidel solution method and updated by

$$q_{m+1}^{n+1} = q_m^{n+1} + \delta q_m^{n+1}.$$

Now, we will focus on the discretization procedure of the Jacobian matrix $\frac{\partial R}{\partial q}$ in the context of the DG scheme. Let us first represent the residual R_l^i as it is defined in Section 2.1

$$R_l^i = \tilde{M}_{lk}^i (H_k^i - S_k^i),$$

where $k, l \in \{1, \cdots, n(p)\}$ denote the indices of degrees of freedom and i is the cell number. Therefore, the components of the discrete Jacobian matrix are the derivatives of the line integrals (6) and the surface integrals (7). The global Jacobian matrix is sparse with a block structure. Each subblock matrix contributed by the cell i depends on its degrees of freedom k and on the neighbouring element's degrees of freedom l. The diagonal block matrices are described by (14a). Equation (14b) corresponds to the block submatrices contributed by the neighbour elements. The interface states q_h^- and q_h^+ indicate here the element i itself and its neighbour j, respectively:

$$\frac{\partial H_k^i}{\partial q_l^i} = \sum_{e=e_{i,j}\in\partial\Omega_i} \sum_{x\in\mathcal{G}_e} |e|\omega_{e,x}\phi_{i,k}(x)\phi_{i,l}(x) \left.\frac{\partial F_{\text{num}}^{\vec{n}_e}}{\partial q_h^-}\right|_{(q_{h,i}, q_{h,j})} \tag{14a}$$

$$\frac{\partial H_k^i}{\partial q_l^j} = \sum_{x\in\mathcal{G}_e} |e|\omega_{e,x}\phi_{i,k}(x)\phi_{j,l}(x) \left.\frac{\partial F_{\text{num}}^{\vec{n}_e}}{\partial q_h^+}\right|_{(q_{h,i}, q_{h,j})} \tag{14b}$$

The numerical flux $F_{\text{num}}^{\vec{n}_e}$ in (14) is approximated by the Roe flux difference method, cf. [15]

$$F_{\text{Roe}}^{\vec{n}_e}(q_h^-, q_h^+) = \frac{1}{2}(F^{\vec{n}_e}(q_h^-) + F^{\vec{n}_e}(q_h^+) - \psi\left(\left.\frac{\partial F^{\vec{n}_e}}{\partial q}\right|_{\bar{q}_{\text{Roe}}}\right)\cdot(q_h^+ - q_h^-)),$$

where ψ is a modification of modulus function, cf. [16, 17]. The Jacobian matrix $\psi\left(\frac{\partial F^{\vec{n}_e}}{\partial q}\right)$ is evaluated at the Roe average state $\bar{q}_{\text{Roe}}$. The approximation of the numerical flux Jacobians consists of neglecting the dependencies of the Roe flux on the average state $\bar{q}_{\text{Roe}}$. Therefore, the numerical flux Jacobians can be approximated by

$$\frac{\partial F_{\text{Roe}}^{\vec{n}_e}}{\partial q_h^-} \approx \frac{1}{2}\left(\left.\frac{\partial F^{\vec{n}_e}}{\partial q}\right|_{q_h^-} + \psi\Big(\left.\frac{\partial F^{\vec{n}_e}}{\partial q}\right|_{\bar{q}_{\text{Roe}}}\Big)\right)$$

$$\frac{\partial F_{\text{Roe}}^{\vec{n}_e}}{\partial q_h^+} \approx \frac{1}{2}\left(\left.\frac{\partial F^{\vec{n}_e}}{\partial q}\right|_{q_h^+} - \psi\Big(\left.\frac{\partial F^{\vec{n}_e}}{\partial q}\right|_{\bar{q}_{\text{Roe}}}\Big)\right).$$

Additionally, the contribution to the diagonal subblocks of $\frac{\partial R}{\partial q}$ consists of the Jacobians of the stiffness matrices (7) evaluated at the mean flow $\bar{q}$

$$\frac{\partial S_k^i}{\partial q_l^i} = \sum_{x \in \mathcal{G}_{\Omega_i}} \omega_{\Omega_i,x} \left(\frac{\partial F^{\nabla_x \phi_{i,k}(x)}}{\partial q} \right) \Big|_{\bar{q}} \cdot \phi_{i,l}(x) \, |\Omega_i|.$$

3 Boundary Conditions

3.1 Solid Wall Slip Conditions

At solid walls the slip boundary conditions are imposed. The velocity components normal to the wall are assumed to be zero on the boundary, i.e.,

$$\vec{n} \cdot \vec{U} = 0 \qquad \text{on} \quad \partial\Omega. \tag{15}$$

The numerical implementation of (15) is carried out using interior solutions and ghost cells at the wall. At every integration point $x \in \mathcal{G}_e$ a ghost value is created, where the velocity is obtained by

$$(\rho\vec{U})_{\text{ext}} = (\rho\vec{U})_{\text{int}} - 2((\rho\vec{U})_{\text{int}}\vec{n}_e)\vec{n}_e.$$

Finally, the numerical flux on the solid boundary is given by

$$F_{\text{bd}}^{\vec{n}_e} = (0, p_h(x) \cdot \vec{n}_e, 0), \qquad x \in \mathcal{G}_e,$$

where $p_h(x)$ is the arithmetic mean of the interior and exterior pressure components.

3.2 Curved Edges

The importance of accurate approximation of geometric boundaries for DG has been pointed out by many authors, cf. [10, 18]. As is described by Bassi, cf. [7], it is mandatory to modify the discretization schemes for high-order elements ($p > 1$) at curved solid wall boundaries in order to avoid oscillatory results and spurious entropy production. These artefacts can be avoided by using higher-order boundary elements. Therefore, for elements adjacent to curved solid walls the standard linear parametrisation is replaced with a polynomial representation on a simplex element $\hat{\Omega}$ of the order q

$$x = \sum_{k=1}^{n(q)} x_k \psi_k(\xi) \qquad \forall \xi \in \hat{\Omega}. \tag{16}$$

Here, the transformation from a reference element to the curved element is expressed in terms of Lagrange polynomials ψ_k and is calculated from additional points on the curved boundary x.

The corresponding formulas for the Jacobians of 1D and 2D elements and the edge normal vectors must be adapted. Therefore, the Jacobians of the edge and the element transformations in integrals (6) and (7) containing higher-order geometric mapping are no longer constant for the triangular elements. These integrals must be recalculated in every integration point $\xi = (\xi_1, \xi_2)$ according to the applied geometric mapping. In this paper, quadratic and cubic polynomials are used in (16). Therefore, the line integral (6) is recalculated as follows

$$\sum_{x \in \mathcal{G}_e} \omega_{e,x} \phi_{i,k}(x) F_{\text{num}}^{\vec{n}_e}(q_{h,i}(x), q_{h,j}(x)) |e| = \sum_{\xi \in \mathcal{G}_{\hat{e}}} \omega_{\hat{e},\xi} \hat{\phi}_{i,k}(\xi) F_{\text{num}}^{\vec{n}_e}(q_{h,i}(\xi), q_{h,j}(\xi)) \|x_\xi\|,$$

here the curved edge length varies at every integration point

$$\|x_\xi\| = \sqrt{\Big(\sum_{k=1}^{n(q)} x_{1,k}\frac{\partial\psi_k(\xi)}{\partial\xi_1}\Big)^2 + \Big(\sum_{k=1}^{n(q)} x_{2,k}\frac{\partial\psi_k(\xi)}{\partial\xi_2}\Big)^2}.$$

Moreover, the evaluation of integrals along the curved edges implies that the normal vectors $\vec{n}_e$ are not constant. The surface integrals (7) are recalculated by

$$\sum_{x\in\mathcal{G}_{\Omega_i}} \omega_{\Omega_i,x} F^{\nabla_x\phi_{i,k}(x)}(q_{h,i}(x))|\Omega_i| = \sum_{\xi\in\mathcal{G}_{\hat{\Omega}}} \omega_{\hat{\Omega},\xi} F^{((J_i(\xi))^{-1}\nabla_\xi\hat{\phi}_{i,k}(\xi))}(q_{h,i}(\xi))\,|J_i(\xi)|,$$

where the determinant $|J_i(\xi)| = |\sum_{k=1}^{n(q)} x_k \frac{\partial\psi_k(\xi)}{\partial\xi}|$ is used. Furthermore, taking into account the non-constant Jacobians, the mass matrices become computationally more expensive for the elements with curved edges.

3.3 Riemann Boundary Conditions

The fundamental idea of Riemann boundary conditions is to impose boundary values at subsonic inlets and outlets by extrapolating the Riemann invariants

$$R_\pm = \|U\| \pm \frac{2a}{\gamma - 1}$$

from the interior of the computational domain [19, 20]. At the Gauss integration points of each boundary edge we introduce virtual exterior states q_{ext} which are computed from the interior states q_{int} which in turn are obtained by evaluating the polynomial q_h of the inner cell. At inflows q_{ext} is the flow such that the Riemann invariant $R_-(q_{\text{ext}})$ and $R_-(q_{\text{int}})$ are identical and such that the boundary conditions for the total pressure, total temperature and velocity angles are satisfied. At outflows q_{ext} is the flow such that the imposed condition on the back pressure is satisfied and such that the Riemann invariant R_+, the entropy and the flow angle coincide with those computed from q_{int}.

Given the exterior state the numerical flux at the Gauss integration point is the standard Roe flux for q_{int} and q_{ext}.

3.4 2D Non-reflecting Boundary Conditions

Since Riemann boundary conditions are based on one-dimensional approximations of Euler flows the quality of two or three-dimensional flow solutions may suffer close to such boundaries. For steady Euler flows non-reflecting boundary conditions based on the theory of the two-dimensional linearized Euler equations have been presented by Giles [13]. These boundary conditions are satisfied if the fluctuation at the inlets and outlets originates from perturbations at the inner domain, see Figure 1. In the case of small fluctuations these boundary conditions are considered as perfect in that the x-position of the boundary will have practically no influence on the flow solution. As proposed by Giles the non-reflecting boundary conditions are expressed as a linear condition amongst the spatial Fourier coefficients of the characteristic variables. For the solver under consideration the implementation outlined in [21] has been adapted to the DG context as follows.

The general idea of the implementation is that at the Gauss points of each entry or exit boundary edge an external state q_{ext} is calculated as a correction of the internal state q_{int} which,

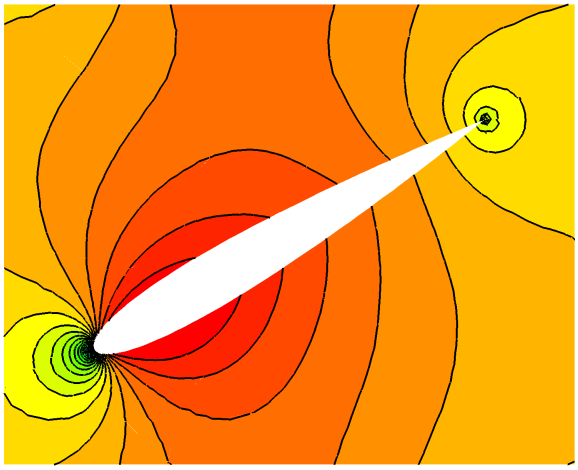


Figure 1: Mach number contours around a cascade of aerofoils using periodic boundaries at upper and lower boundaries as well as non-reflecting boundary conditions at the inlet and outlet

as before, is computed from the polynomial q_h at the inner cell. The non-reflecting boundary conditions are satisfied if the correction vanishes, i.e., if the exterior state is identical to the interior state. More precisely a correction of the characteristics is applied to the inner state,

$$q_{\text{ext}} = q_{\text{int}} + \sigma \frac{\partial q}{\partial c}(\delta c_{\text{glb}} + \delta c_{\text{nrf}}), \tag{17}$$

with a relaxation parameter $\sigma < 1$. Splitting the vector of characteristic variables into incoming and outgoing characteristics, c^{inc} and c^{out}, respectively, the change of outgoing characterstics is set to zero $\delta c^{\text{out}} = 0$. On the one hand the correction of the incoming characteristics is the update of one iteration of the Newton method applied to the global boundary condition

$$\delta c^{\text{inc}}_{\text{glb}} = -\left(\frac{\partial R_{\text{bv}}}{\partial c^{\text{inc}}}\right)^{-1} R_{\text{bv}}(\bar{q}),$$

where $\bar{q}$ is the flow state averaged over the entry or exit boundary. On the other hand a local change of the incoming characteristics, δc_{nrf}, is calculated from the fluctuation $q - \bar{q}$ following the procedure described in [21]. The numerical flux function used for the surface integration of the DG scheme is a central flux applied to the inner and outer state variables at the Gauss points of the edges.

In order to use these boundary conditions on general meshes a Fourier transform is implemented for non-equidistant points. Given a function q along y the complex Fourier coefficients $\hat{q}_0$, $\hat{q}_1$,... are determined such that

$$\int_0^{\Delta y} \left(q(y) - \sum_{k=0}^{K} \hat{q}_k e^{\frac{2\pi i k y}{\Delta y}} \right)^2 dy \tag{18}$$

is minimized. Here, the integral in (18) is approximated using Gauss quadrature formulas implemented for the spatial discretization scheme. If the number of complex Fourier coefficients is sufficiently small the inverse transform

$$(\hat{q}_k)_{k=-K}^{K} \mapsto (\hat{q}_k e^{\frac{2\pi i k y}{\Delta y}})_{x \in \mathcal{G}_\Gamma}$$

is injective. Denoting the corresponding matrix by $\tilde{F}$, the Fourier transform F is the Moore-Penrose pseudoinverse, i.e.,

$$F = (\tilde{F})^{+} = (\tilde{F}^{*}\tilde{F})^{(-1)}\tilde{F}^{*}.$$

Here the adjoint matrix $\tilde{F}^{*}$ is the adjoint w.r.t. the L^2 scalar product defined by the Gauss integration formulas.

4 Numerical Results

4.1 Acoustic Pulse

To assess the potential of the DG method a benchmark problem (Category 4, Problem 1) from the first workshop on benchmark problems in computational aeroacoustics is solved [22]. This test case consists in the propagation of acoustic waves generated by a Gaussian pulse. Since the exact solution to the Linearized Euler Equations (LEEs) is available here, the results of simulations are used to study the accuracy of the spatial discretization scheme. Moreover, by this numerical example the effectiveness of the wall boundary conditions combined with the DG high-order spatial discretization scheme and the straight element boundaries is examined.

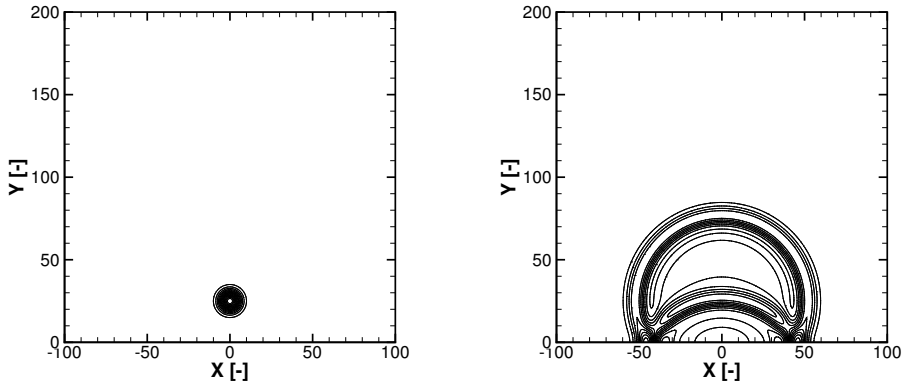


Figure 2: Initialisation of the pressure contour (left) and the solution at $T = 50$ (right)

The computational domain used in the simulations is shown on the left of Figure 2 along with the initial pressure perturbation field. The computed pressure disturbance field at time $T = 50$ is shown on the right of Figure 2. Here the acoustic waves can be seen reflecting off the solid wall located along the lower boundary of the computational domain.

Mesh convergence analysis is necessary in order to provide the accuracy verification of the numerical scheme. The computations are performed on four grids at the final time $T = 50$. These grids contain unstructured triangular elements with varying grid spacing. The coarsest grid employed here contains cells with the edge length $\Delta x = 12$. For the finest grid the edges of the length $\Delta x = 1$ are used. Integration in time is performed by the explicit method (9) with a time step of 0.005. This time step is chosen in such a way that the error of the temporal discretization scheme has a minimal influence on the overall error resulting from the temporal and spatial discretizations.

Figure 3 illustrates the computational time history at the node in space $(x, y) = (0, 50)$ at the time $T = 50$ for the different polynomial approximations: $p = 0, 1, 2, 3$. Here, the results of the computations for various orders p are compared with the analytic solution, which is represented by a black line. In order to measure the errors between the analytic and the numerical solutions

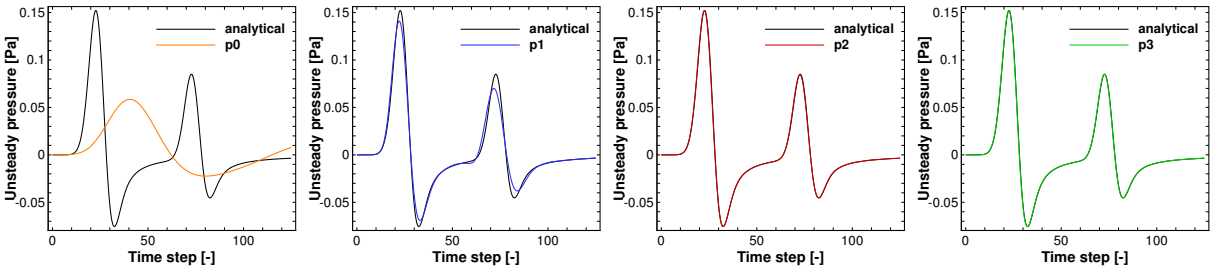
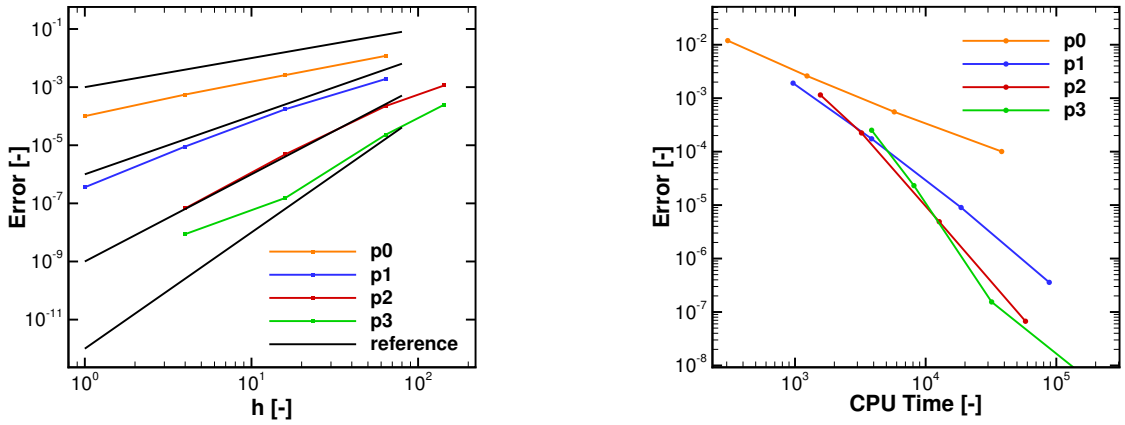


Figure 3: Time history for polynomial orders: $p = 0, 1, 2, 3$



(a) Error estimates compared to the theoretical convergence slopes

(b) CPU time behaviour

Figure 4: Spatial accuracy and efficiency rates

the L_2-norm has been applied. The accuracy of the obtained results is validated up to the fourth order in space, see Figure 4. Spatial discretization errors are shown to correspond to the theoretical orders of convergence, see Figure 4(a).

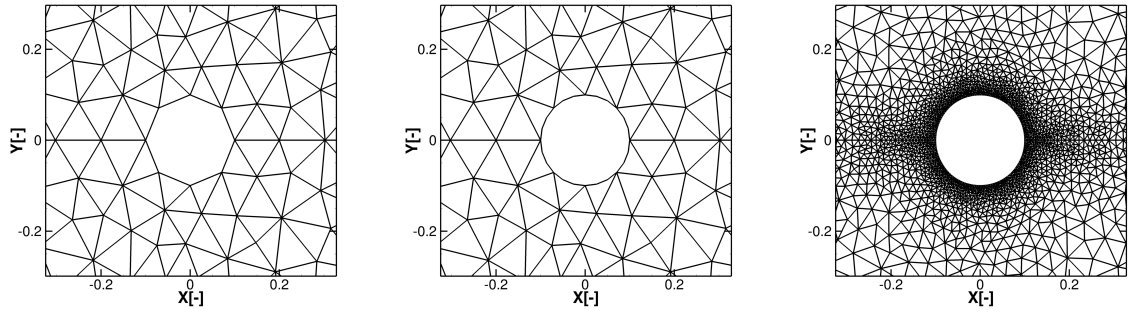
The CPU behaviour obtained by analysing the run time, is provided by Figure 4(b). Here, the L_2-error is plotted against the computational time. It is observed that the higher-order schemes outperform the lower-order ones in achieving the certain accuracy level. However, as it is already mentioned, for the fourth-order scheme the separation of the spatial discretization errors and the temporal discretization errors is not evident enough. The computations on the finest grid here lead to better efficiency results for the third-order of accuracy. The CPU behaviour obtained by analysing the run time, is provided by Figure 4(b). Here, the L_2-error is plotted against the computational time. It is observed that the higher-order schemes generally outperform the lower-order ones in achieving the certain accuracy level. However, as it is already mentioned, for the fourth-order scheme the separation of the spatial discretization errors and the temporal discretization errors is not evident enough. The computations on the finest grid here lead to better efficiency results for the third-order of accuracy.

4.2 Flow Around a Circle

In the present section, a steady flow will be considered by means of a two-dimensional circular cylinder. Moreover, the curvatures of the grid sides near the solid walls should be

taken into account and their performance should be examined in detail here. As discussed in Subsection 3.2, employing grids with straight-sided edges leads to the production of spurious oscillations near the element discontinuities. Therefore, curvilinear elements obtained by a quadratic geometric mapping are used in the second- and third-order accurate computations, and the cubic geometric mapping is applied to the fourth-order accurate elements.

The discretization of a computational domain is symmetric along the axis $y = 0$ and resolved by unstructured triangles. The Riemann boundary conditions are employed on the exterior boundaries. However, the boundaries here are placed very far from the cylinder surface in order to imply far field boundaries and avoid unwanted influence of these Riemann boundary conditions on the solution. The coarsest and the finest grids contain 8 and 120 nodes at the inner wall surface, respectively, see Figures 5(a) and 5(c). An example of the coarse grid with curved edges is displayed by Figure 5(b).



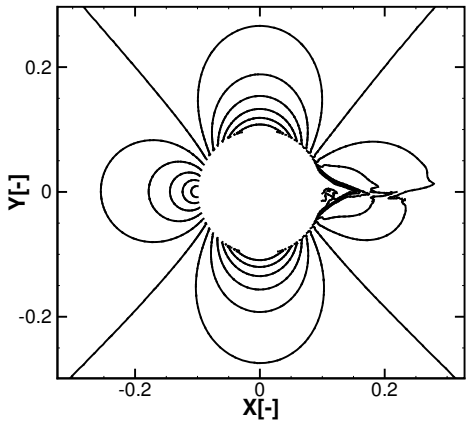
(a) The coarsest mesh with straight-sided boundaries (b) The coarsest mesh with curved boundary edges (c) The finest mesh

Figure 5: Circular cylinder grids

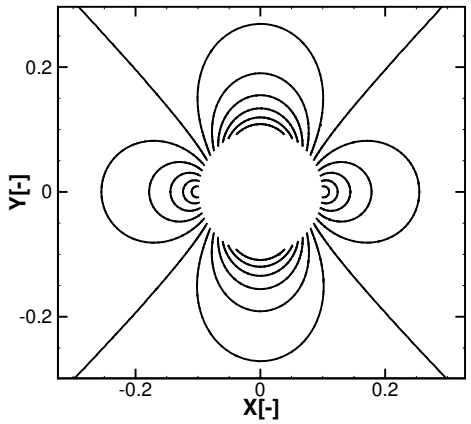
The problem is solved at Mach number $M_\infty = 0.3$. Plots 6(a) and 6(b) display the Mach isolines obtained with cubic elements on the finest mesh. Figure 6(a) shows an unsymmetric solution at the trailing edge of the surface which is due to the straight boundary edges along the cylinder surface. On the other hand, using additional information about the curvature of the geometry, the solution improves significantly, see Figure 6(b). The results of the computations performed on the coarsest mesh with a combination of curved boundaries and the cubic polynomials are shown by Figure 6(c). Clearly the results obtained in this case compare very favourably with the reference solution. The agreement between the fourth-order accurate solutions on the coarse and fine grids is confirmed by the pressure coefficient distributions which are represented in Figure 6(d). One observes here an excellent agreement of the pressure coefficients belonging to the test calculation on the coarse mesh and the results on the finest mesh. Thus, providing elements with curved boundary edges are used, the fourth order scheme is able to retain sufficient accuracy on the very coarse grid.

4.3 Turbomachinery Test Case

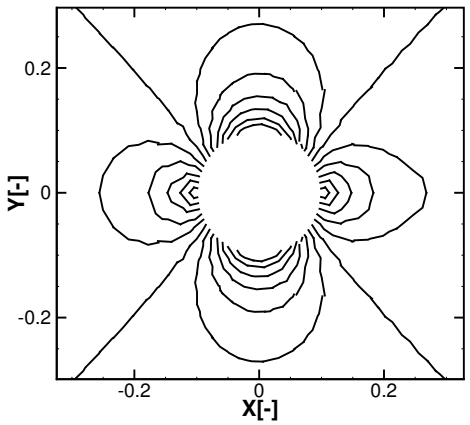
The performance of the non-reflecting boundary conditions is demonstrated by means of a cascade of periodically arranged 2D turbomachinery profiles. The profile corresponds to the midsection of the stator blade of DLR's Ultra-High Bypass Ratio (UHBR) Fan stage [23]. At the operating point considered here the flow is subsonic with Mach numbers in the range of 0.3 to 0.6.



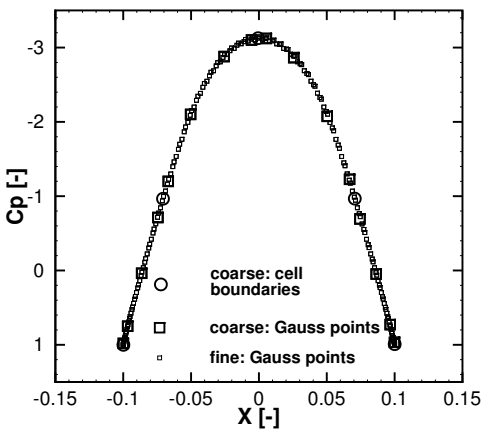
(a) Mach isolines on the finest mesh for $p = 3$ with straight edges



(b) Mach isolines on the finest mesh for $p = 3$ with curved edges



(c) Mach isolines on the coarsest mesh for $p = 3$ with curved edges



(d) Comparison of the pressure coefficients on coarsest and finest meshes

Figure 6: Mach isolines and pressure coefficients along the surface of the circular cylinder using the fourth-order elements

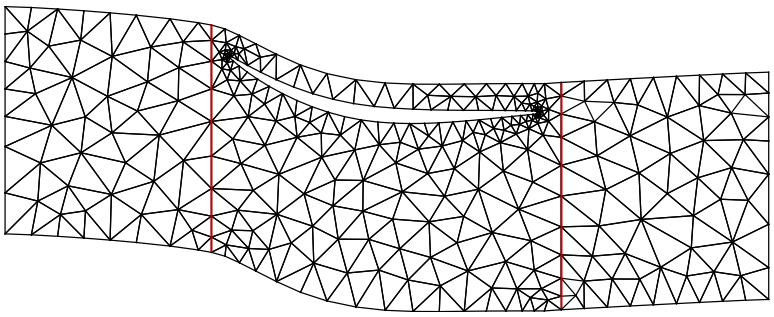
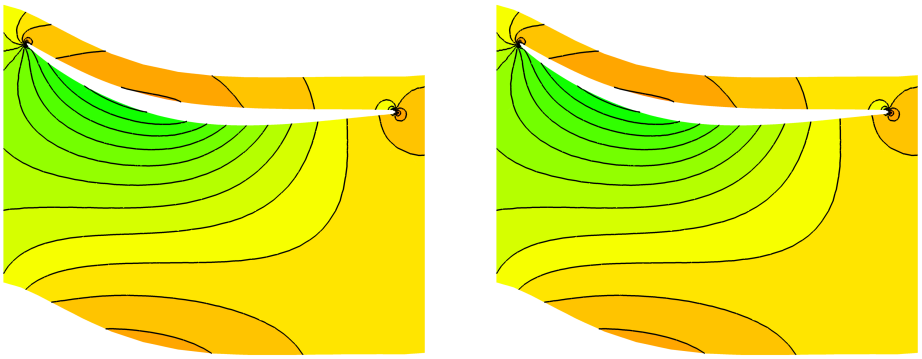


Figure 7: Long (358 cells, black) and short (245 cells, red) computational domains

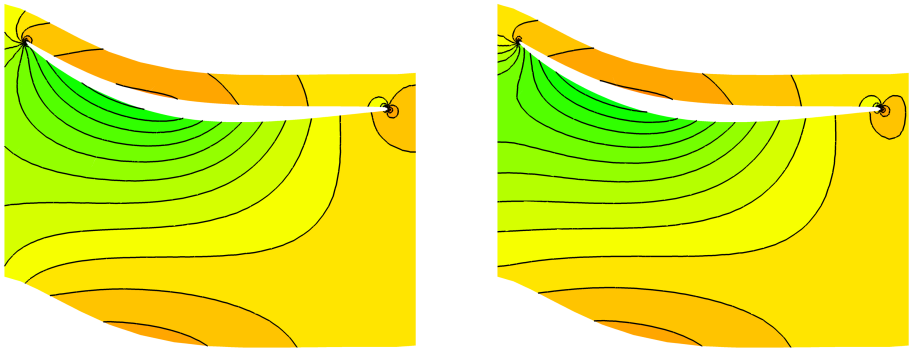
The use of appropriate inflow and outflow boundary conditions is crucial for turbomachinery applications since the inlet and outlet boundaries of the blade rows are typically close to the



(a) Nonreflecting boundary conditions, long domain

(b) Riemann boundary conditions, long domain

Figure 8: Comparison of pressure contours on the long computational domain



(a) Nonreflecting boundary conditions, short domain

(b) Riemann boundary conditions, short domain

Figure 9: Comparison of pressure contours on the short computational domain

near fields of the blades. Therefore, numerical reflections may affect the solutions considerably.

Here, the steady simulations are performed using simple one-dimensional Riemann boundary conditions as well as more sofisticated two-dimensional, non-reflecting boundary conditions at the inflow and outflow boundaries. In order to demonstrate the performance of the incorporated boundary conditions the computations are carried out on short and long computational domains. These domains consist of 245 and 358 unstructured triangular elements, respectively, see Figure 7. The simulations are performed using curved boundary elements on the profile. The turbomachinery flow solver TRACE [24] has been used to obtain an inviscid reference solution on the long computational domain resolved by a fine unstructured grid containing 65512 elements. TRACE is a second-order FV solver for the compressible RANS equations that has been developed at DLR's Institute of Propulsion Technology.

Figures 8 and 9 display the pressure contours predicted by the DG solver employing cubic elements with non-reflecting (left) and Riemann (right) boundary conditions on the short and long computational domains, respectively. Whereas the solutions on the long domain are in perfect agreement, the pressure contours near the leading and trailing edges that have been computed on the short domain differ considerably when Riemann boundary conditions are employed. However, the flow solution obtained with two-dimensional non-reflecting boundary conditions is nearly independent of the location of the entry and exit boundaries.

To better quantify the accuracy of the non-reflecting boundary conditions, the distribution of the pressure coefficients close to the leading edge is shown in Figure 10. Comparison with the reference solutions shows that in spite of the extremely coarse mesh the DG solver can accurately predict the pressure distribution near the leading edge provided high-order elements and non-reflecting boundary conditions are used. In contrast, the solutions obtained with Riemann boundary conditions are of low quality even when high-order elements are employed.

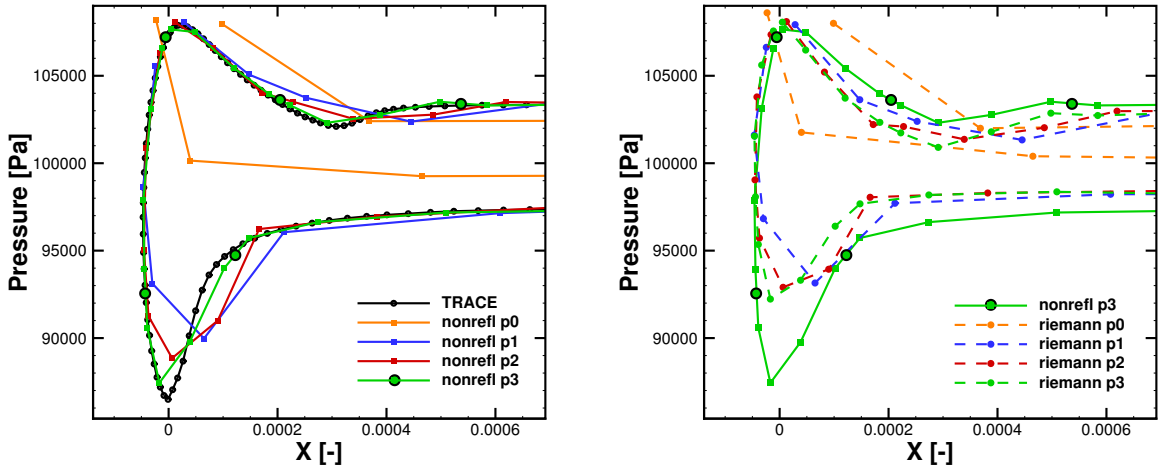


Figure 10: Distribution of pressure coefficients at the leading edge of UHBR stator profile. Nonreflecting (left) and Riemann (right) boundary conditions on the short computational domain

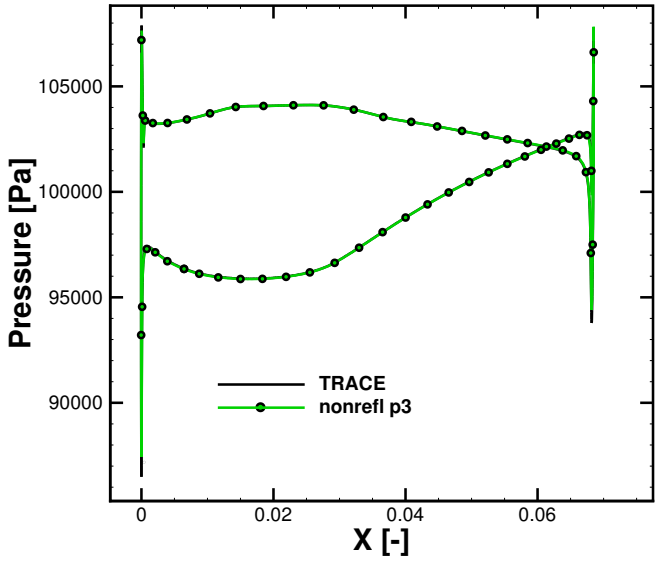


Figure 11: Distribution of the pressure coefficients of DG solution compared to the reference result at UHBR stator.

The distribution of the pressure coefficient on the entire blade is shown in Figure 11. Here the solution is obtained with the DG discretization of the fourth-order and non-reflecting boundary

conditions. The element edges are indicated here with black circles. One can see an excellent agreement between the DG solution and the reference result obtained by TRACE on a very fine grid using the second-order discretization scheme.

5 Conclusions

To evaluate the DG method for turbomachinery CFD applications the method has been implemented for the compressible, two-dimensional Euler equations. The application of the method to several academic problems has been used to validate the solver, guage the efficiency of the high order $(p > 1)$ accuracy discretizations and demonstrate the importance of curved boundary treatments when dealing with complex geometries. Furthermore, a strategy for the integration of two-dimensional, non-reflecting boundary conditions into the DG framework has been presented and applied to simulate the subsonic flow about a stator vane near midspan. Future work will focus on the incorporation of the DG method into a three-dimensional Navier-Stokes solver for the investigation and design of next generation compressors and turbines.

REFERENCES

[1] W. H. Reed and T. R. Hill, "Triangular mesh methods for the neutron transport equation," tech. rep., Los Alamos Scientific Laboratory, 1973.

[2] B. Cockburn and C.-W. Shu, "TVD Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws ii: General framework," *Mathematics of Computations*, vol. 52, no. 186, pp. 411–435, 1989.

[3] B. Cockburn and C.-W. Shu, "The Runge-Kutta discontinuous Galerkin method for conservation laws v: Multidimensional systems," *J. Computational Physics*, vol. 141, no. 1998, pp. 199–224, 1997.

[4] B. Landmann, *A parallel discontinuous Galerkin code for the Navier-Stokes and Reynolds-averaged Navier-Stokes equations*. PhD thesis, the Faculty of Aerospace Engineering and Geodesy of the University of Stuttgart, 2009.

[5] W. Li and D. J. Mavriplis, "Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations," *J. Comput. Phys.*, vol. 225, no. 2, pp. 1994–2015, 2007.

[6] G. Arambatzis, P. Vavilis, I. Toulopoulos, and A. Ekaterinaris, "Implicit high-order time-marching schemes for the linearized Euler equations," *AIAA Journal*, vol. 45, no. 8, pp. 1819–1826, 2007.

[7] F. Bassi and S. Rebay, "High-order accurate discontinuous solution of the 2D Euler equations," *Journal of computational physics*, vol. 138, pp. 251–285, 1997.

[8] B. Cockburn, *Discontinuous Galerkin methods for convection-dominated problems. High-order methods for computational physics, Lecture Notes in Computational Science and Engineering, 69–224.*, vol. 9. Springer, 1999.

[9] B. Cockburn and C. Shu, "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, vol. 16, no. 3, pp. 173–261, 2001.

[10] D. Mavriplis, N. C., K. Shahbazi, L. Wang, and N. Burgess, "Progress high-order discontinuous Galerkin methods for aerospace applications," *AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, vol. 601, 2009.

[11] I. Toulopoulos and A. Ekaterinaris, "Implementation of characteristic boundary conditions to the discontinuous Galerkin method," *44th AIAA Aerospace Science Meeting and Exhibit*, no. 108, 2006.

[12] I. Toulopoulos and A. Ekaterinaris, "Characteristic boundary conditions for the numerical solution of Euler equations by the discontinuous Galerkin methods," in *V European Conference on Computational Fluid Dynamics ECCOMAS CFD* (J. Pereira and A. Sequeira, eds.), (Lisbon, Portugal), 2010.

[13] M. B. Giles, "Non-reflecting boundary conditions for the Euler equations," tech. rep., MIT Dept. of Aero. and Astr., 1988. CFDL Report 88-1.

[14] J. Blazek, *Computational fluid dynamics: principles and applications*. Elsevier Science, 2001.

[15] P. L. Roe, "Approximate riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, no. 2, pp. 357 – 372, 1981.

[16] A. Harten, P. D. Lax, and B. Van Leer, "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws," *Soc. Industrial and applied mathematics Rew.*, vol. 25, no. 1, 1983.

[17] A. Harten and J. M. Hyman, "Self adjusting grid methods for one-dimensional hyperbolic conservation laws," *J. Computational Physics*, no. 50, pp. 235–269, 1983.

[18] L. Krivodonova and M. Berger, "High-order accurate implementation of solid wall boundary conditions in curved geometries," *J. Computational Physics*, no. 211, pp. 492–512, 2006.

[19] C. B. Laney, *Computational gasdynamics*. Cambridge University Press, 1998.

[20] C. Hirsch, *Numerical Computation of Internal and External Flows – Computational Methods for Inviscid and Viscous Flows*, vol. 2. Wiley, 1 ed., 1990.

[21] M. Giles, "UNSFLO: A numerical method for the calculation of unsteady flow in turobmachinery," tech. rep., Gas Turbine Laboratory Report GTL 205, MIT Dept. of Aero. and Astro., 1991.

[22] J. C. Hardin, J. R. Ristorcelli, and C. K. W. Tam, "ICASE/LaRC workshop on benchmark problems in computational aeroacoustics," *NASACP3300*, 1995.

[23] B. Kaplan, E. Nicke, and C. Voss, "Design of a highly efficient low-noise fan for ultra-high bypass engines," 2006. ASME Paper No. GT2006-90363.

[24] K. Becker, K. Heitkamp, and E. Kügeler, "Recent progress in a hybrid-grid cfd solver for turbomachinery flows," in *Proceedings Fifth European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010. V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.