Hartmann, Ralf and Houston, Paul (2005) Symmetric Interior Penalty DG Methods for the Compressible Navier-Stokes Equations I: Method Formulation.

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 00, Number 0, Pages 000–000

SYMMETRIC INTERIOR PENALTY DG METHODS FOR THE COMPRESSIBLE NAVIER–STOKES EQUATIONS I: METHOD FORMULATION

RALF HARTMANN AND PAUL HOUSTON

Abstract. In this article we consider the development of discontinuous Galerkin finite element methods for the numerical approximation of the compressible Navier–Stokes equations. For the discretization of the leading order terms, we propose employing the generalization of the symmetric version of the interior penalty method, originally developed for the numerical approximation of linear self-adjoint second–order elliptic partial differential equations. In order to solve the resulting system of nonlinear equations, we exploit a (damped) Newton–GMRES algorithm. Numerical experiments demonstrating the practical performance of the proposed discontinuous Galerkin method with higher–order polynomials are presented.

Key words. Finite element methods, discontinuous Galerkin methods, compressible Navier–Stokes equations

1. Introduction

In recent years there has been tremendous interest in the design of discontinuous Galerkin finite element methods (DGFEMs, for short) for the discretization of compressible fluid flow problems; see, for example, [3, 4, 5, 8, 9, 10, 11, 24] and the references cited therein. The key advantages of these schemes are that DGFEMs provide robust and high-order accurate approximations, particularly in transport-dominated regimes, and that they are locally conservative. Moreover, there is considerable flexibility in the choice of the mesh design; indeed, DGFEMs can easily handle non-matching grids and non-uniform, even anisotropic, polynomial approximation degrees. Additionally, orthogonal bases can easily be constructed which lead to diagonal mass matrices; this is particularly advantageous for unsteady problems. Finally, in combination with block-type preconditioners, DGFEMs can easily be parallelized.

In our previous work, see the series of papers [16, 17, 18, 21], for example, we have been concerned with the development of DGFEMs for the numerical approximation of inviscid compressible fluid flows, coupled with automatic adaptive mesh generation. In particular, the key focus of these articles was the derivation of so–called 'goal–oriented' *a posteriori* error bounds together with the design of corresponding adaptive mesh refinement algorithms in order to yield guaranteed error control; for the generalization of these ideas to the *hp*–version of the DGFEM, we refer to the article [28] and the references cited therein. In contrast to traditional *a posteriori* error estimation which seeks to bound the error with respect to a given norm, goal–oriented *a posteriori* error estimation bounds the error measured in terms of certain target functionals of real or physical interest. Typical examples include the mean value of the field over the computational domain Ω, the normal flux through the outflow boundary of Ω, the evaluation of the solution at a given point in Ω and the drag and lift coefficients of a body immersed in a fluid. For related work, we refer to [6, 24], for example.

The purpose of this article and its companion–article [20] is to extend our earlier work on DGFEMs for nonlinear systems of first–order hyperbolic conservation laws to the compressible Navier–Stokes equations. For the discretization of the leading order terms, we propose employing the symmetric version of the interior penalty DGFEM. One of the key aspects of this discretization scheme is the satisfaction of the adjoint consistency condition, cf. [1], for linear problems. This condition is essential to guarantee that the optimal order of convergence of the numerical approximation to the underlying analytical solution is attained when the discretization error is measured in terms of either the L_2–norm, or in the 'goal–oriented' setting, in terms of a given target functional of practical interest. This property is *not* shared by, for example, the non-symmetric version of the interior penalty DGFEM, cf. [14]. Indeed, this latter method is sub-optimal by a full order of the mesh size h, when the error is measured in terms of the L_2-norm, for even polynomial degrees; though it has been noted experimentally, that the optimal rate of convergence of the scheme is achieved for odd orders, cf. [22, 25].

The paper is structured as follows. After introducing, in Section 2, the compressible Navier–Stokes equations, in Section 3 we formulate its discontinuous Galerkin finite element approximation; here, a detailed description of the implementation of the corresponding boundary conditions is also outlined. Section 4 is devoted to the practical implementation of the underlying discretization method; in particular, here we propose a damped Newton–GMRES algorithm for the solution of the system of nonlinear equations arising from the DGFEM discretization of the underlying PDE system. In Section 5 we present a series of numerical experiments to illustrate the performance of the proposed symmetric interior penalty DGFEM when higher–order polynomial degrees are employed. In particular, we demonstrate the performance of the nonlinear Newton iteration with different preconditioning strategies. Then, we compare the convergence of force coefficients under both global and local grid refinement for a standard laminar test case, as well as highlighting the numerical resolution of boundary layer profiles when linear and higher-order polynomial degrees are employed. Finally, in Section 6 we summarize the work presented in this paper and draw some conclusions.

2. The compressible Navier-Stokes equations

We consider the two–dimensional steady state compressible Navier-Stokes equations. Writing ρ, $\mathbf{v} = (v_1, v_2)^T$, p, E and T to denote the density, velocity vector, pressure, specific total energy and temperature, respectively, the equations of motion are given by

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})) \equiv \frac{\partial}{\partial x_i}\mathbf{f}_i^c(\mathbf{u}) - \frac{\partial}{\partial x_i}\mathbf{f}_i^v(\mathbf{u}, \nabla\mathbf{u}) = 0 \quad \text{in } \Omega, \tag{1}$$

where Ω is an open bounded domain in $\mathbb{R}^2$; here, and throughout the rest of this article, we use the summation convention, i.e., repeated indices are summed through their range. The vector of conservative variables $\mathbf{u}$ and the convective fluxes $\mathbf{f}_i^c$, $i = 1, 2$, are defined by

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

respectively, and the viscous fluxes $\mathbf{f}_i^v$, $i = 1, 2$, are defined by

$$\mathbf{f}_1^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{1j} v_j + \mathcal{K} T_{x_1} \end{bmatrix} \quad \text{and} \quad \mathbf{f}_2^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{2j} v_j + \mathcal{K} T_{x_2} \end{bmatrix}, \tag{3}$$

respectively, where $\mathcal{K}$ is the thermal conductivity coefficient. Additionally, H is the total enthalpy given by

$$H = E + \frac{p}{\rho} = e + \tfrac{1}{2}\mathbf{v}^2 + \frac{p}{\rho},$$

where e is the specific static internal energy, and the pressure is determined by the equation of state of an ideal gas

$$p = (\gamma - 1)\rho e, \tag{4}$$

where $\gamma = c_p/c_v$ is the ratio of specific heat capacities at constant pressure, c_p, and constant volume, c_v; for dry air, $\gamma = 1.4$. Finally, the viscous stress tensor is defined by

$$\tau = \mu \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T - \tfrac{2}{3}(\nabla \cdot \mathbf{v}) I \right), \tag{5}$$

where μ is the dynamic viscosity coefficient, and the temperature T is given by $e = c_v T$; thus

$$\mathcal{K} T = \tfrac{\mu\gamma}{Pr} \left(E - \tfrac{1}{2}\mathbf{v}^2 \right),$$

where $Pr = 0.72$ is the Prandtl number.

For the purposes of discretization, we rewrite the compressible Navier–Stokes equations (1) in the following (equivalent) form:

$$\frac{\partial}{\partial x_i} \left(\mathbf{f}_i^c(\mathbf{u}) - G_{ij}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j} \right) = 0 \quad \text{in } \Omega. \tag{6}$$

Here, the matrices $G_{ij}(\mathbf{u}) = \partial \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) / \partial u_{x_j}$, for $i, j = 1, 2$, i.e., $\mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) = G_{ij}(\mathbf{u}) \partial \mathbf{u} / \partial x_j$, $i = 1, 2$, where

$$G_{11} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3} v_1 & \frac{4}{3} & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ -\left(\frac{4}{3} v_1^2 + v_2^2 + \frac{\gamma}{Pr} \left(E - \mathbf{v}^2 \right) \right) & \left(\frac{4}{3} - \frac{\gamma}{Pr} \right) v_1 & \left(1 - \frac{\gamma}{Pr} \right) v_2 & \frac{\gamma}{Pr} \end{pmatrix},$$

$$G_{12} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3} v_2 & 0 & -\frac{2}{3} & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{1}{3} v_1 v_2 & v_2 & -\frac{2}{3} v_1 & 0 \end{pmatrix}, G_{21} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ \frac{2}{3} v_1 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3} v_1 v_2 & -\frac{2}{3} v_2 & v_1 & 0 \end{pmatrix},$$

$$G_{22} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{4}{3} v_2 & 0 & \frac{4}{3} & 0 \\ -\left(v_1^2 + \frac{4}{3} v_2^2 + \frac{\gamma}{Pr} \left(E - \mathbf{v}^2 \right) \right) & \left(1 - \frac{\gamma}{Pr} \right) v_1 & \left(\frac{4}{3} - \frac{\gamma}{Pr} \right) v_2 & \frac{\gamma}{Pr} \end{pmatrix}.$$

Given that $\Omega \subset \mathbb{R}^2$ is a bounded region, with boundary Γ, the system of conservation laws (6) must be supplemented by appropriate boundary conditions. For simplicity of presentation, we assume that Γ may be decomposed as follows

$$\Gamma = \Gamma_{\mathrm{D,sup}} \cup \Gamma_{\mathrm{D,sub\text{-}in}} \cup \Gamma_{\mathrm{D,sub\text{-}out}} \cup \Gamma_{\mathrm{N}} \cup \Gamma_{\mathrm{W}},$$

where $\Gamma_{\mathrm{D,sup}}$, $\Gamma_{\mathrm{D,sub\text{-}in}}$, $\Gamma_{\mathrm{D,sub\text{-}out}}$, Γ_{N}, and Γ_{W} are distinct subsets of Γ representing Dirichlet (supersonic), Dirichlet (subsonic-inflow), Dirichlet (subsonic-outflow), Neumann (supersonic-outflow), and solid wall boundaries, respectively. Thereby, we may specify the following boundary conditions:

$$\mathcal{B}(\mathbf{u}) = \mathcal{B}(\mathbf{g}_D) \text{ on } \Gamma_{\text{D,sup}} \cup \Gamma_{\text{D,sub-in}} \cup \Gamma_{\text{D,sub-out}}, \quad \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})\cdot\mathbf{n} = \mathbf{g}_{\text{N}} \text{ on } \Gamma_{\text{N}}, \tag{7}$$

where $\mathbf{g}_{\text{D}}$ and $\mathbf{g}_{\text{N}}$ are given Dirichlet and Neumann boundary conditions, respectively. Here, $\mathcal{B}$ is a boundary operator employed to enforce appropriate Dirichlet conditions on $\Gamma_{\text{D,sup}} \cup \Gamma_{\text{D,sub-in}} \cup \Gamma_{\text{D,sub-out}}$. For simplicity of presentation, we assume that $\mathcal{B}(\mathbf{u}) = \mathbf{u}$ on $\Gamma_{\text{D,sup}}$, $\mathcal{B}(\mathbf{u}) = (u_1, u_2, u_3, 0)^T$ on $\Gamma_{\text{D,sub-in}}$ and $\mathcal{B}(\mathbf{u}) = \big(0, 0, 0, (\gamma-1)(u_4 - (u_2^2 + u_3^2)/(2u_1))\big)^T$ on $\Gamma_{\text{D,sub-out}}$; we note that this latter condition enforces a specific pressure $p_{\text{out}} = (\mathcal{B}(\mathbf{g_D}))_4$ on $\Gamma_{\text{D,sub-out}}$.

For solid wall boundaries, we consider the distinction between *isothermal* and *adiabatic* conditions. To this end, decomposing $\Gamma_W = \Gamma_{\text{W,iso}} \cup \Gamma_{\text{W,adia}}$, we set

$$\mathbf{v} = \mathbf{0} \text{ on } \Gamma_{\text{W}}, \qquad T = T_{\text{wall}} \text{ on } \Gamma_{\text{W,iso}}, \qquad \mathbf{n}\cdot\nabla T = 0 \text{ on } \Gamma_{\text{W,adia}},$$

where T_{wall} is a given wall temperature; we refer to [3, 4, 7, 9, 12] and the references cited therein for further details concerning the imposition of suitable boundary conditions.

3. Discontinuous Galerkin Discretization with Interior Penalty

In this section we introduce the discontinuous Galerkin method with interior penalty for the discretization of the compressible Navier–Stokes equations (6).

We assume that Ω can be subdivided into shape-regular meshes $\mathcal{T}_h = \{\kappa\}$ consisting of quadrilateral elements κ. For each $\kappa \in \mathcal{T}_h$, we denote by $\mathbf{n}_\kappa$ the unit outward normal vector to the boundary $\partial\kappa$, and by h_κ the elemental diameter. An interior edge of $\mathcal{T}_h$ is the (non-empty) one-dimensional interior of $\partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$. Similarly, a boundary edge of $\mathcal{T}_h$ is the (non-empty) one-dimensional interior of $\partial\kappa \cap \Gamma$ which consists of entire edges of $\partial\kappa$. We denote by $\Gamma_{\mathcal{I}}$ the union of all interior edges of $\mathcal{T}_h$.

Next, we define average and jump operators. To this end, let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior edge $e = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_{\mathcal{I}}$. Moreover, let $\mathbf{v}$ and $\underline{\tau}$ be vector- and matrix-valued functions, respectively, that are smooth inside each element $\kappa^\pm$. By $(\mathbf{v}^\pm, \underline{\tau}^\pm)$ we denote the traces of $(\mathbf{v}, \underline{\tau})$ on e taken from within the interior of $\kappa^\pm$, respectively. Then, we define the averages at $\mathbf{x} \in e$ by $\{\!\{\mathbf{v}\}\!\} = (\mathbf{v}^+ + \mathbf{v}^-)/2$ and $\{\!\{\underline{\tau}\}\!\} = (\underline{\tau}^+ + \underline{\tau}^-)/2$. Similarly, the jump at $\mathbf{x} \in e$ is given by $\underline{[\![\mathbf{v}]\!]} = \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}$. On a boundary edge $e \subset \Gamma$, we set $\{\!\{\mathbf{v}\}\!\} = \mathbf{v}$, $\{\!\{\underline{\tau}\}\!\} = \underline{\tau}$ and $\underline{[\![\mathbf{v}]\!]} = \mathbf{v} \otimes \mathbf{n}$. For matrices $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m\times n}$, $m, n \geq 1$, we use the standard notation $\underline{\sigma} : \underline{\tau} = \sum_{k=1}^m \sum_{l=1}^n \sigma_{kl}\tau_{kl}$; additionally, for vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$, the matrix $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{m\times n}$ is defined by $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k w_l$.

In order to derive the interior penalty discontinuous Galerkin discretization of equations (6), we first re-write (6) as a system of first–order partial differential equations, by introducing appropriate auxiliary variables. To this end, we have

$$\underline{\sigma} = \left(G_{1j}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j}, G_{2j}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j} \right), \quad \text{in } \Omega, \tag{8}$$

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \underline{\sigma}) = 0, \qquad \text{in } \Omega, \tag{9}$$

subject to appropriate boundary conditions, cf. above.

Next, we introduce the finite element space $\underline{\Sigma}_h \times \mathbf{V}_h$, where

$$\begin{aligned}\underline{\Sigma}_h &= \{\underline{\tau} \in [L_2(\Omega)]^{4\times 2} : \underline{\tau}|_\kappa \in [\mathcal{Q}_p(\kappa)]^{4\times 2},\ \kappa \in \mathcal{T}_h\},\\ \mathbf{V}_h &= \{\mathbf{v} \in [L_2(\Omega)]^{4} : \mathbf{v}|_\kappa \in [\mathcal{Q}_p(\kappa)]^{4},\ \kappa \in \mathcal{T}_h\},\end{aligned}$$

for an approximation order $p \geq 1$. Here, $\mathcal{Q}_p(\kappa)$ denotes the space of tensor product polynomials on κ of degree p in each coordinate direction.

Taking the $L_2(\kappa)$, $\kappa \in \mathcal{T}_h$, inner product of (8) and (9) with smooth test functions $\underline{\tau} = (\underline{\tau}_1, \underline{\tau}_2)$ and $\mathbf{v}$, respectively, and integrating by parts gives

$$\begin{aligned}\int_\kappa \underline{\sigma} : \underline{\tau}\,\mathrm{d}\mathbf{x} &= \int_{\partial\kappa} \mathbf{u}\cdot (G_{ij}^T \underline{\tau}_i) n_{x_j}\,\mathrm{d}s - \int_\kappa \mathbf{u}\cdot \frac{\partial}{\partial x_j}(G_{ij}^T \underline{\tau}_i)\,\mathrm{d}\mathbf{x},\\ -\int_\kappa \mathcal{F}^c(\mathbf{u}) : \nabla\mathbf{v}\,\mathrm{d}\mathbf{x} &+ \int_{\partial\kappa} (\mathcal{F}^c(\mathbf{u})\cdot \mathbf{n}_\kappa)\cdot \mathbf{v}\,\mathrm{d}s + \int_\kappa \underline{\sigma} : \nabla \mathbf{v}\,\mathrm{d}\mathbf{x}\\ &- \int_{\partial\kappa\setminus(\Gamma_{\mathrm{N}}\cup\Gamma_{\mathrm{W,adia}})} \underline{\sigma} : \mathbf{v}\otimes\mathbf{n}_\kappa\,\mathrm{d}s - \int_{\partial\kappa\cap\Gamma_{\mathrm{N}}} \mathbf{g}_{\mathrm{N}}\cdot\mathbf{v}\,\mathrm{d}s\\ &\qquad - \int_{\partial\kappa\cap\Gamma_{\mathrm{W,adia}}} \underline{\sigma}^{\mathrm{adia}} : \mathbf{v}\otimes\mathbf{n}_\kappa\,\mathrm{d}s = 0,\end{aligned}$$

where on the adiabatic boundary $\Gamma_{\mathrm{W,adia}}$, we define $\underline{\sigma}^{\mathrm{adia}}$ such that

$$\underline{\sigma}^{\mathrm{adia}}\cdot\mathbf{n} \equiv \mathcal{F}^{v,\mathrm{adia}}(\mathbf{u},\nabla\mathbf{u})\cdot\mathbf{n} = (0, \tau_{1j}n_{x_j}, \tau_{2j}n_{x_j}, \tau_{ij}v_j n_{x_i})^T.$$

Summing over all elements κ in the computational mesh $\mathcal{T}_h$ and introducing appropriate numerical flux functions, which will be defined below, we deduce the following auxiliary formulation of the interior penalty DGFEM: find $(\underline{\sigma}_h, \mathbf{u}_h) \in \underline{\Sigma}_h \times \mathbf{V}_h$ such that

$$\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \underline{\sigma}_h : \underline{\tau}_h\,\mathrm{d}\mathbf{x} = \sum_{\kappa\in\mathcal{T}_h}\left\{\int_{\partial\kappa} \widehat{\mathbf{u}}_h\cdot(G_{ij}^T\underline{\tau}_{h,i})n_{x_j}\,\mathrm{d}s - \int_\kappa \mathbf{u}_h\cdot\frac{\partial}{\partial x_j}(G_{ij}^T\underline{\tau}_{h,i})\,\mathrm{d}\mathbf{x}\right\}, \tag{10}$$

$$\begin{aligned}\sum_{\kappa\in\mathcal{T}_h}\Bigg\{-\int_\kappa \mathcal{F}^c(\mathbf{u}_h) : \nabla\mathbf{v}_h\,\mathrm{d}\mathbf{x} &+ \int_{\partial\kappa\setminus\Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa)\cdot\mathbf{v}_h^+\,\mathrm{d}s\\ &+ \int_{\partial\kappa\cap\Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}_\kappa)\cdot\mathbf{v}_h^+\,\mathrm{d}s + \int_\kappa \underline{\sigma}_h : \nabla\mathbf{v}_h\,\mathrm{d}\mathbf{x}\Bigg\}\\ -\int_{\Gamma_{\mathcal{I}}\cup\Gamma\setminus(\Gamma_{\mathrm{N}}\cup\Gamma_{\mathrm{W,adia}})} \widehat{\underline{\sigma}}_h : \underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s &- \int_{\Gamma_{\mathrm{W,adia}}} \widehat{\underline{\sigma}}_h^{\mathrm{adia}} : \underline{[\![\mathbf{v}_h]\!]}\,\mathrm{d}s - \int_{\Gamma_{\mathrm{N}}} \mathbf{g}_{\mathrm{N}}\cdot\mathbf{v}_h\,\mathrm{d}s = 0\end{aligned} \tag{11}$$

for all $(\underline{\tau}_h, \mathbf{v}_h) \in \underline{\Sigma}_h \times \mathbf{V}_h$. Here, the numerical flux functions $\mathcal{H}(\cdot,\cdot,\cdot)$, $\widehat{\mathbf{u}}_h$, $\widehat{\underline{\sigma}}_h$ and $\widehat{\underline{\sigma}}_h^{\mathrm{adia}}$ may be chosen to be any Lipschitz continuous, consistent and conservative fluxes which are discrete approximations to traces on the boundary of the elements in the mesh.

For generality the hyperbolic numerical flux function $\mathcal{H}(\cdot,\cdot,\cdot)$, will be left unspecified at this point; however, we point out that any Lipschitz continuous, consistent and conservative numerical flux function employed for the numerical approximation of systems of hyperbolic conservation laws may be employed. For example, in Section 5 we use the Vijayasundaram flux; see [23, 29], for example.

To define the symmetric interior penalty discretization of the viscous terms, the remaining numerical fluxes are defined as follows: for an edge e which lies inside the domain Ω, we have

$$\widehat{\mathbf{u}}_h = \{\!\{\mathbf{u}_h\}\!\} \qquad \text{and} \qquad \widehat{\underline{\sigma}}_h = \{\!\{\mathcal{F}^v(\mathbf{u}_h, \nabla \mathbf{u}_h)\}\!\} - \delta \underline{[\![\mathbf{u}_h]\!]};$$

while for boundary edges we write

$$\widehat{\mathbf{u}}_h = \mathbf{u}_\Gamma(\mathbf{u}_h^+), \qquad \widehat{\underline{\sigma}}_h = \mathcal{F}^v(\mathbf{u}_h^+, \nabla \mathbf{u}_h^+) - \delta(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \otimes \mathbf{n},$$

and on $\Gamma_{\mathrm{W,adia}}$, we set

$$\widehat{\underline{\sigma}}_h^{\mathrm{adia}} = \mathcal{F}^{v,\mathrm{adia}}(\mathbf{u}_h^+, \nabla \mathbf{u}_h^+) - \delta(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \otimes \mathbf{n}. \tag{12}$$

Here, for simplicity, we set the discontinuity penalization matrix $\delta = \mathrm{diag}\{\delta_i, i = 1, \ldots, 4\}$, where

$$\delta_i|_e = C_{\mathrm{IP}} \frac{\mu p^2}{\tilde{h}} \quad \text{for } e \subset \Gamma_{\mathcal{I}} \cup \Gamma, \tag{13}$$

$\tilde{h} = \min(\mathrm{meas}(\kappa), \mathrm{meas}(\kappa'))/\mathrm{meas}(e)$ represents the element dimension orthogonal to the edge e of elements κ and κ' adjacent to e, and C_{IP} is a positive constant, which, for reasons of stability, must be chosen sufficiently large, cf. [1]; see also [13] for the extension to the anisotropic case.

Finally, the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ is given according to the type of boundary condition imposed. To this end, we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_D \ \text{ on } \ \Gamma_{\mathrm{D,sup}}, \qquad \mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u} \ \text{ on } \ \Gamma_{\mathrm{N}},$$

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left((g_D)_1, (g_D)_2, (g_D)_3, \frac{p(\mathbf{u})}{\gamma - 1} + \frac{(g_D)_2^2 + (g_D)_3^2}{2(g_D)_1}\right)^T \quad \text{on } \ \Gamma_{\mathrm{D,sub\text{-}in}}, \tag{14}$$

and

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(u_1, u_2, u_3, \frac{p_{\mathrm{out}}}{\gamma - 1} + \frac{u_2^2 + u_3^2}{2u_1}\right)^T \quad \text{on } \ \Gamma_{\mathrm{D,sub\text{-}out}}. \tag{15}$$

Here, $p \equiv p(\mathbf{u})$ denotes the pressure evaluated using the equation of state (4). Finally, we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_1 c_v T_{\mathrm{wall}})^T \quad \text{on } \ \Gamma_{\mathrm{W,iso}},$$

and

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_4)^T \quad \text{on } \ \Gamma_{\mathrm{W,adia}}.$$

Remark 3.1. *We note that the flux functions $\widehat{\underline{\sigma}}_h$ and $\widehat{\underline{\sigma}}_h^{adia}$ are consistent for any choice of δ; however, it is well known that the stability of the underlying discretization crucially depends on the magnitude of this discontinuity–penalization parameter, cf.* [1], *for example. For the choice of the corresponding numerical flux functions for the non-symmetric interior penalty method, together with other schemes proposed in the literature, we refer to the article* [1].

Remark 3.2. *We note that an alternative way of enforcing the adiabatic wall boundary condition is to define the numerical flux $\widehat{\underline{\sigma}}_h^{adia}$ as in (12), with $\mathcal{F}^{v,adia}$ defined by*

$$\mathcal{F}^{v,adia}(\mathbf{u}, \nabla \mathbf{u}) \cdot \mathbf{n} = (0, \tau_{1i} n_{x_i}, \tau_{2i} n_{x_i}, 0)^T;$$

in this case, we also select the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ *by*

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(u_1, 0, 0, \frac{p(\mathbf{u})}{\gamma - 1}\right)^T = \left(u_1, 0, 0, u_4 - \frac{u_2^2 + u_3^2}{2u_1}\right)^T \quad \text{on } \Gamma_{W,adia},$$

cf. [19]. *Numerical comparisons we have performed indicate that there are only minor differences in the computed numerical solution when the two implementations of the adiabatic wall boundary condition are computed.*

It is usually desirable to eliminate the auxiliary variables $\underline{\sigma}_h$, in order to reduce the size of the underlying system of nonlinear equations. This can be achieved by selecting $\underline{\tau}_h = \nabla \mathbf{v}_h$ in (10), integrating by parts, and inserting the resulting expression for the term involving the product of $\underline{\sigma}_h$ and $\nabla \mathbf{v}_h$ into (11). Thereby, the so–called primal formulation of the symmetric interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations (6) is as follows: find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$\begin{aligned}
\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) \equiv & - \int_\Omega \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa) \cdot \mathbf{v}_h^+ \,\mathrm{d}s \\
& + \int_\Omega \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} - \int_{\Gamma_\mathcal{I}} \{\!\!\{\mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h)\}\!\!\} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& - \int_{\Gamma_\mathcal{I}} \{\!\!\{\left(G_{i1}^T \partial_h \mathbf{v}_h / \partial x_i, G_{i2}^T \partial_h \mathbf{v}_h / \partial x_i\right)\}\!\!\} : \underline{[\![\mathbf{u}_h]\!]} \,\mathrm{d}s + \int_{\Gamma_\mathcal{I}} \delta \underline{[\![\mathbf{u}_h]\!]} : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& + \int_\Gamma \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) \cdot \mathbf{v}_h^+ \,\mathrm{d}s + \int_{\Gamma \setminus \Gamma_N} \delta \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right) \cdot \mathbf{v}_h^+ \,\mathrm{d}s, \\
& - \int_{\Gamma \setminus (\Gamma_N \cup \Gamma_{\mathrm{W,adia}})} \mathcal{F}^v(\mathbf{u}_h^+, \nabla_h \mathbf{u}_h^+) : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s - \int_{\Gamma_\mathrm{N}} \mathbf{g}_\mathrm{N} \cdot \mathbf{v}_h \,\mathrm{d}s \\
& - \int_{\Gamma_{\mathrm{W,adia}}} \mathcal{F}^{v,\mathrm{adia}}(\mathbf{u}_h^+, \nabla_h \mathbf{u}_h^+) : \underline{[\![\mathbf{v}_h]\!]} \,\mathrm{d}s \\
& - \int_{\Gamma \setminus \Gamma_\mathrm{N}} \left(G_{i1}^T(\mathbf{u}_h^+) \partial_h \mathbf{v}_h^+ / \partial x_i, G_{i2}^T(\mathbf{u}_h^+) \partial_h \mathbf{v}_h^+ / \partial x_i\right) : \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right) \otimes \mathbf{n} \,\mathrm{d}s = 0
\end{aligned} \tag{16}$$

for all $\mathbf{v}_h$ in $\mathbf{V}_h$. Here, the subscript h on the operators ∇_h and $\partial_h / \partial x_i$, $i = 1, 2$, is used to denote the discrete counterparts of ∇ and $\partial / \partial x_i$, $i = 1, 2$, respectively, taken elementwise.

Remark 3.3. *We note that the non-symmetric version of the interior penalty DGFEM is obtained by changing the sign in front of the fifth and twelveth terms on the left-hand side of (16); see, for example,* [3, 10].

4. Newton–GMRES algorithm

To determine the numerical solution $\mathbf{u}_h$ of the system of nonlinear equations (16), we employ a damped Newton method. This nonlinear iteration generates a sequence of approximations $\mathbf{u}_h^n$, $n = 0, 1, \ldots$, to the actual numerical solution $\mathbf{u}_h$, using the following algorithm. Given an iterate $\mathbf{u}_h^n$, the update $\mathbf{d}_h^n$ of $\mathbf{u}_h^n$ to get to the next iterate

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \omega^n \mathbf{d}_h^n$$

is defined by: find $\mathbf{d}_h^n \in \mathbf{V}_h$ such that

$$\mathcal{N}'_{\mathbf{u}}[\mathbf{u}_h^n](\mathbf{d}_h^n, \mathbf{v}_h) = R(\mathbf{u}_h^n, \mathbf{v}_h) \equiv -\mathcal{N}(\mathbf{u}_h^n, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h. \tag{17}$$

Here, w^n denotes a damping parameter, which is dynamically chosen to guarantee that the discrete l_2-norm of the residual computed with $\mathbf{u}_h^{n+1}$ is less than the same quantity computed with $\mathbf{u}_h^n$. Additionally, $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ denotes (an approximation to) the Fréchet derivative of $\mathbf{u} \to \mathcal{N}(\mathbf{u}, \mathbf{v})$, for $\mathbf{v} \in \mathbf{V}_h$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$, where $\mathbf{V}$ is some suitable chosen function space such that $\mathbf{V}_h \in \mathbf{V}$. More precisely, we approximate $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ by

$$\begin{aligned}
\hat{\mathcal{N}}'_{\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}) &= -\int_\Omega (\mathcal{F}^c_{\mathbf{u}}(\mathbf{w})\boldsymbol{\phi}) : \nabla_h \mathbf{v} \,\mathrm{d}\mathbf{x} \\
&+ \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \Big(\hat{\mathcal{H}}'_{\mathbf{u}^+}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)\boldsymbol{\phi}^+ + \hat{\mathcal{H}}'_{\mathbf{u}^-}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)\boldsymbol{\phi}^- \Big) \cdot \mathbf{v}^+ \,\mathrm{d}s \\
&+ \int_\Omega (\mathcal{F}^v_{\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\boldsymbol{\phi}) : \nabla_h \mathbf{v} \,\mathrm{d}\mathbf{x} + \int_\Omega (\mathcal{F}^v_{\nabla\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\nabla_h \boldsymbol{\phi}) : \nabla_h \mathbf{v} \,\mathrm{d}\mathbf{x} \\
&- \int_{\Gamma_{\mathcal{I}}} \{\!\{\mathcal{F}^v_{\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\boldsymbol{\phi}\}\!\} : \underline{[\![\mathbf{v}]\!]} \,\mathrm{d}s - \int_{\Gamma_{\mathcal{I}}} \{\!\{\mathcal{F}^v_{\nabla\mathbf{u}}(\mathbf{w}, \nabla_h \mathbf{w})\nabla_h \boldsymbol{\phi}\}\!\} : \underline{[\![\mathbf{v}]\!]} \,\mathrm{d}s \\
&- \int_{\Gamma_{\mathcal{I}}} \{\!\{ \Big(\Big(\big(G_{i1}^T\big)'(\mathbf{w})\boldsymbol{\phi} \Big) \partial_h \mathbf{v}/\partial x_i, \Big(\big(G_{i2}^T\big)'(\mathbf{w})\boldsymbol{\phi} \Big) \partial_h \mathbf{v}/\partial x_i \Big) \}\!\} : \underline{[\![\mathbf{w}]\!]}) \,\mathrm{d}s \\
&- \int_{\Gamma_{\mathcal{I}}} \{\!\{ \big(G_{i1}^T(\mathbf{w})\partial_h \mathbf{v}/\partial x_i, G_{i2}^T(\mathbf{w})\partial_h \mathbf{v}/\partial x_i \big) \}\!\} : \underline{[\![\boldsymbol{\phi}]\!]}) \,\mathrm{d}s + \int_{\Gamma_{\mathcal{I}}} \delta \underline{[\![\boldsymbol{\phi}]\!]} : \underline{[\![\mathbf{v}]\!]} \,\mathrm{d}s \\
&+ \hat{\mathcal{N}}'_{\Gamma,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}),
\end{aligned}$$

where $\mathbf{w} \to \hat{\mathcal{H}}'_{\mathbf{u}^+}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)$ and $\mathbf{w} \to \hat{\mathcal{H}}'_{\mathbf{u}^-}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}_\kappa)$ denote approximations to the derivatives of the flux function $\mathcal{H}(\cdot, \cdot, \cdot)$ with respect to its first and second arguments, respectively; for a detailed description of these derivatives for two specific choices of numerical fluxes, we refer to the article [15]. Furthermore, $\hat{\mathcal{N}}'_{\Gamma,\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ denotes the Fréchet derivative of the boundary terms, for $\mathbf{v} \in \mathbf{V}_h$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$, which is given by

$$\begin{aligned}
&\hat{\mathcal{N}}'_{\Gamma,\mathbf{u}}[\mathbf{w}](\boldsymbol{\phi}, \mathbf{v}) = \\
&\int_\Gamma \Big(\hat{\mathcal{H}}'_{\mathbf{u}^+}\left(\mathbf{w}^+, \mathbf{u}_\Gamma(\mathbf{w}^+), \mathbf{n}\right) + \hat{\mathcal{H}}'_{\mathbf{u}^-}\left(\mathbf{w}^+, \mathbf{u}_\Gamma(\mathbf{w}^+), \mathbf{n}\right) \mathbf{u}'_\Gamma(\mathbf{w}^+) \Big) \boldsymbol{\phi}^+ \cdot \mathbf{v}^+ \,\mathrm{d}s \\
&+ \int_{\Gamma \setminus \Gamma_N} \delta \left(\boldsymbol{\phi}^+ - \mathbf{u}'_\Gamma(\mathbf{w}^+)\boldsymbol{\phi}^+ \right) \cdot \mathbf{v}^+ \,\mathrm{d}s, \\
&- \int_{\Gamma \setminus (\Gamma_N \cup \Gamma_{\mathrm{W,adia}})} \left(\mathcal{F}^v_{\mathbf{u}}(\mathbf{w}^+, \nabla_h \mathbf{w}^+)\boldsymbol{\phi}^+ + \mathcal{F}^v_{\nabla\mathbf{u}}(\mathbf{w}^+, \nabla_h \mathbf{w}^+)\nabla_h \boldsymbol{\phi}^+ \right) : \underline{[\![\mathbf{v}]\!]} \,\mathrm{d}s \\
&- \int_{\Gamma_{\mathrm{W,adia}}} \left(\mathcal{F}^{v,\mathrm{adia}}_{\mathbf{u}}(\mathbf{w}^+, \nabla_h \mathbf{w}^+)\boldsymbol{\phi}^+ + \mathcal{F}^{v,\mathrm{adia}}_{\nabla\mathbf{u}}(\mathbf{w}^+, \nabla_h \mathbf{w}^+)\nabla_h \boldsymbol{\phi}^+ \right) : \underline{[\![\mathbf{v}]\!]} \,\mathrm{d}s \\
&- \int_{\Gamma \setminus \Gamma_{\mathrm{N}}} \Big(\Big(\big(G_{i1}^T\big)'(\mathbf{w}^+)\boldsymbol{\phi}^+ \Big) \partial_h \mathbf{v}^+/\partial x_i, \Big(\big(G_{i2}^T\big)'(\mathbf{w}^+)\boldsymbol{\phi}^+ \Big) \partial_h \mathbf{v}^+/\partial x_i \Big) \\
&\qquad : \big(\mathbf{w}^+ - \mathbf{u}_\Gamma(\mathbf{w}^+) \big) \otimes \mathbf{n} \,\mathrm{d}s \\
&- \int_{\Gamma \setminus \Gamma_{\mathrm{N}}} \left(G_{i1}^T(\mathbf{w}^+)\partial_h \mathbf{v}^+/\partial x_i, G_{i2}^T(\mathbf{w}^+)\partial_h \mathbf{v}^+/\partial x_i \right) : \left(\boldsymbol{\phi}^+ - \mathbf{u}'_\Gamma(\mathbf{w}^+)\boldsymbol{\phi}^+ \right) \otimes \mathbf{n} \,\mathrm{d}s,
\end{aligned}$$

where $\mathbf{u}'_\Gamma(\mathbf{u})$ denotes the derivative of the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ with respect to the conservative variables (i.e. the components of) $\mathbf{u}$. On the supersonic parts of the boundary, we have

$$\mathbf{u}'_\Gamma(\mathbf{u}) = 0 \quad \text{on} \quad \Gamma_{\mathrm{D,sup}} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = I \in \mathbb{R}^{4,4} \quad \text{on} \quad \Gamma_{\mathrm{N}};$$

on the subsonic boundaries $\Gamma_{\text{D,sub-in}}$ and $\Gamma_{\text{D,sub-out}}$, $\mathbf{u}'_\Gamma(\mathbf{u})$ is given by

$$\mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}|\mathbf{v}|^2 & -v_1 & -v_2 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2}|\mathbf{v}|^2 & v_1 & v_2 & 0 \end{pmatrix},$$

respectively. Finally, we have

$$\mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c_v T_{\mathrm{wall}} & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{u}'_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

on the isothermal and adiabatic no-slip boundaries, respectively.

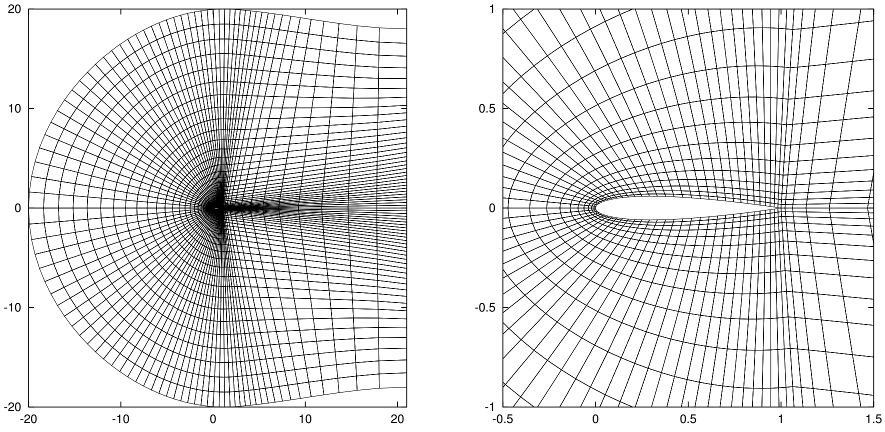
5. Numerical results

In this section we present a series of numerical experiments to highlight the practical performance of the interior penalty DGFEM introduced in this article for the numerical approximation of the compressible Navier-Stokes equations. In all cases, we employ the (damped) Newton algorithm outlined in Section 4 to solve the resulting set of nonlinear equations. Within each inner (linear) iteration, we exploit a (left) preconditioned GMRES algorithm; comparisons between different preconditioning strategies will be presented in Section 5.1.2 for the first test problem. Throughout this section we select the constant C_{IP} appearing in the discontinuity penalisation parameter δ_i, $i = 1, \ldots, 4$, defined in (13) as follows: $C_{\mathrm{IP}} = 10$. This choice is based purely on numerical experience; indeed, we have consistently employed the same value of C_{IP} for a wide range of problems, including linear advection–diffusion equations, second–order quasi-linear elliptic partial differential equations, and the time-harmonic Maxwell system. In all cases, this choice of C_{IP} is sufficiently large to guarantee stability of the underlying interior penalty DGFEM, without being so large as to adversely affect the conditioning of the resulting system of linear/nonlinear equations.

5.1. Subsonic flow around a NACA0012 airfoil. In this first example, we consider the subsonic viscous flow around a NACA0012 airfoil; here, the upper and lower surfaces of the airfoil geometry are specified by the function $g^\pm$, respectively, where

$$g^\pm(s) = \pm 5 \times 0.12 \times (0.2969 s^{1/2} - 0.126 s - 0.3516 s^2 + 0.2843 s^3 - 0.1015 s^4).$$

As the chord length l of the airfoil is $l \approx 1.00893$ we use a rescaling of g in order to yield an airfoil of unit (chord) length. The computational domain Ω is subdivided into quadrilateral elements; cf. the C-type grid depicted in Figures 1(a) and (b). Curved boundaries are approximated by piecewise quadratic polynomials. At the farfield (inflow) boundary we specify a Mach 0.5 flow at a zero angle of attack, with Reynolds number Re = 5000; on the walls of the airfoil geometry, we impose a zero heat flux (adiabatic) no-slip boundary condition. This is a standard laminar



(a) (b)

FIGURE 1. Computational mesh: (a) Full view; (b) Zoom of coarse grid with 3072 elements.

test case which has been investigated by many other authors, cf. [4], for example. The solution to this problem consists of a strictly subsonic flow which is symmetric about the x-axis, see Figures 2(a) and (b).

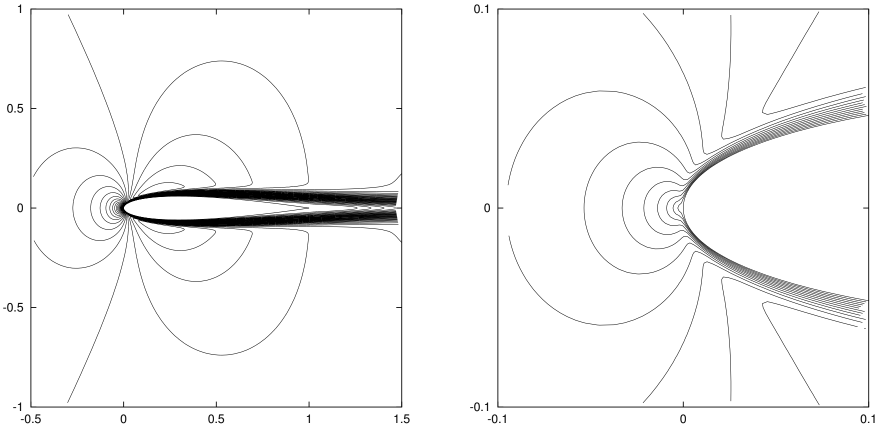
In this section we demonstrate the practical performance of both the (damped) Newton algorithm proposed in Section 4, as well as illustrating the accuracy of the proposed DGFEM with increasing polynomial order. To this end, in Section 5.1.1 we first study the effects of the imposition of different boundary conditions at the farfield boundary on the computed numerical solution. In Section 5.1.2 we provide a computational comparison of four different preconditioning strategies for a single inner (linear) iteration of Newton's method; the performance of the full Newton algorithm will then be outlined in Section 5.1.3. Finally, in Section 5.1.4 we demonstrate the accuracy of the proposed DGFEM for the numerical evaluation of the force coefficients on the airfoil; here, we consider both a sequence of globally refined meshes, as well as sequence of meshes generated by employing local refinement using standard residual–based error indicators; see [20], for example.

5.1.1. Farfield boundary conditions. In this section we present numerical results to justify the choice of the farfield inflow and outflow boundary conditions outlined in Section 3 for the current case of a subsonic laminar flow. Let us first consider the subsonic outflow condition; here, we propose enforcing the following condition:

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(u_1, u_2, u_3, \frac{p_{\text{out}}}{\gamma - 1} + \frac{u_2^2 + u_3^2}{2u_1}\right)^T \quad \text{on } \Gamma_{\text{D,sub-out}}, \tag{18}$$

This boundary condition sets the density and velocity components from the computed flow field and imposes the pressure p_{out} only, corresponding to the fact that at subsonic outflow boundaries three characteristics leave the domain and only one enters; here, we have set p_{out} to be the pressure of free-stream flow conditions.

Of course, if the farfield boundary is a sufficiently large distance from the airfoil geometry, then an alternative condition is simply to enforce the outflow boundary



(a) (b)

FIGURE 2. Mach isolines of the flow around the NACA0012 airfoil: (a) $\mathrm{Ma} = \frac{i}{50}, i = 1, \ldots$; (b) $\mathrm{Ma} = \frac{i}{20}, i = 1, \ldots$.

condition

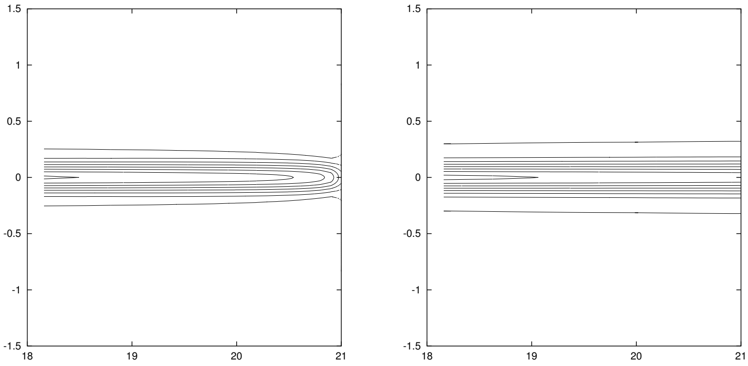
$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u} \quad \text{on} \quad \Gamma_{\text{D,sub-out}},$$

which simply extrapolates all the conserved variables from computed flow; this is precisely what is done at supersonic outflow boundaries. However, as expected, in the subsonic setting the resulting nonlinear solver does not converge. The alternative to these approaches, is to simply set the Dirichlet boundary condition

$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_D, \tag{19}$$

on the outflow boundary; here, $\mathbf{g}_D$ represents the free-stream flow conditions. While the resulting nonlinear solver converges to machine accuracy, it can be seen in Figure 3(a) that an artificial boundary layer is introduced in the vicinity of the computational domain where the wake behind the profile crosses the outflow boundary. In contrast, the imposition of subsonic outflow boundary condition, cf. the condition (18) on $\Gamma_{\text{D,sub-out}}$, ensures that the wake behind the airfoil geometry is undisturbed as it exits the computational domain, cf. Figure 3(b). We remark that these differences in the computed solution in the vicinity of the outflow boundary will greatly affect the efficiency of employing residual–based mesh refinement; indeed, employing the boundary condition (19) will lead to the adaptive algorithm unnecessarily refining the computational mesh near the outflow boundary in order to resolve the artificial boundary layer introduced, cf. [20].

Finally, we consider the proposed subsonic boundary condition (14) on the inflow part of the computational domain $\Gamma_{\text{D,sub-in}}$. Given that the numerical solution is virtually constant in the vicinity of the inflow boundary, we observed only very small differences between the numerical solution computed with the condition (14), and the corresponding solution computed by simply imposing (standard) Dirichlet boundary conditions for all the flow variables. However, again adaptive refinement based on the latter boundary conditions did lead to unnecessary refinement of the mesh in the vicinity of $\Gamma_{\text{D,sub-in}}$.



(a) (b)

Figure 3. Mach isolines of the wake near the outflow boundary for: (a) Dirichlet boundary condition; (b) Subsonic outflow boundary condition.

5.1.2. Choice of the preconditioner. In this section we compare the performance of four different preconditioners with the restarted GMRES method (using 200 vectors) for solving the linear problems arising within the Newton algorithm. The first three preconditioners – block-Jacobi, block-Gauss-Seidel and symmetric block-Gauss-Seidel – exploit the block structure of the resulting Jacobian matrix. We remark that this block-structure originates from the discontinuous nature of the test and trial functions employed within the DGFEM: diagonal blocks represent inner element operations, while the off-diagonal blocks stem from the communication between each element and their (edge–wise) neighbours. The block preconditioners employed here are those implemented in the *deal.II* library, [2]. Additionally, we also consider the ILU preconditioner (with no additional fill-in) included within the PETSc library, [26].

In Table 1 we show the number of iterations and the cpu time (in seconds) required to solve the linear system arising within the first Newton step employing each of the above preconditioners, together with GMRES. Here, the discretization is based on employing the mesh shown in Figure 1, with 3072 elements, with the polynomial degree $p = 1$ (bilinear elements), which results in 49152 degrees of freedom. For each case, the starting guess was taken to be the free-stream boundary conditions; here, the ℓ_2-norm of the residual of this linear problem is reduced by a factor of 10^{-4}. As expected from the increasing complexity of the preconditioners, in the given ordering, we observe that the number of iteration steps decreases from one preconditioner to the next; i.e. the ILU preconditioner uses less iterations than the symmetric block-Gauss-Seidel preconditioner, which in turn uses less iterations than the block-Gauss-Seidel preconditioner, and so on. Analogous improvements in the amount of cpu time needed by each preconditioner are also observed; indeed, the ILU preconditioner clearly out performs the block-type preconditioners for this problem. However, a disadvantage of the ILU preconditioner is the amount of additional memory required to store a full copy of the matrix (assuming no fill in is employed, as in this example). On the other hand, the block-type preconditioners only require the storage of the inverse of the diagonal blocks of the underlying

Table 1. Comparison of the preconditioners: number of linear steps and time required for reducing the ℓ_2–norm of the linear residual of the first Newton step by a factor of 10^{-4}.

Preconditioner	# steps	time(s)
block-Jacobi	782	347.3
block-Gauss-Seidel	333	193.5
sym. block-Gauss-Seidel	135	96.0
ILU	78	14.3

matrix, which is clearly significantly less than the full matrix, particularly on nonconforming meshes which contain hanging nodes. Notwithstanding this last point, for the purposes of the current article, all the following computations will be performed using the ILU preconditioner.

5.1.3. Performance of the Newton iteration. In this section we illustrate the performance of the Newton algorithm on successively (globally) refined meshes for $p = 1, 2, 3$. The initial coarsest mesh consists of 3072 elements, cf. Figure 1; the subsequent two additional meshes have 12288 and 49152 elements, respectively. As in the previous section the ℓ_2–norm of the linear residual of each Newton step is reduced by a factor of 10^{-4}. In Figure 4 we show the convergence history of the ℓ_2–norm of the non-linear residuals for $p = 1$. After seven damped Newton steps on the coarsest mesh the computed (approximate) solution is sufficiently close to the actual numerical solution so that the Newton algorithm proceeds without any damping, i.e. the damping parameter $\omega^n = 1$ and we observe quadratic-like Newton convergence. Once the ℓ_2–norm of the non-linear residual is below the prescribed tolerance (10^{-6}), the mesh is globally refined, and the solution is interpolated to the refined mesh. On the second and third meshes only four and two, respectively, undamped Newton steps are required to ensure that the convergence criterion is satisfied.

In Table 2 we summarize the number of Newton steps required to satisfy the nonlinear stopping criterion for both bilinear elements, as well as for higher–order elements with $p = 2, 3$. For higher–order elements, we use the numerical solution computed with bilinear elements on the coarsest mesh as the starting guess for the Newton algorithm. This then results in only 4 iterations being needed for both $p = 2, 3$ on the coarsest mesh; on the finer two meshes only two steps are required for both biquadratic and bicubic elements.

5.1.4. Convergence of force coefficients. Finally, in this section we demonstrate the accuracy of the proposed interior penalty DGFEM for the computation of force coefficients of the underlying viscous flow under both global and local mesh refinement.

In particular, we consider the estimation of the drag and lift coefficients, c_{d} and c_{l}, respectively, which, in the case of a viscous flow, are defined by

$$\begin{aligned} J_{c_{\mathrm{d}}}(\mathbf{u}) &= J_{c_{\mathrm{dp}}}(\mathbf{u}) + J_{c_{\mathrm{df}}}(\mathbf{u}), \\ J_{c_{\mathrm{l}}}(\mathbf{u}) &= J_{c_{\mathrm{lp}}}(\mathbf{u}) + J_{c_{\mathrm{lf}}}(\mathbf{u}), \end{aligned}$$

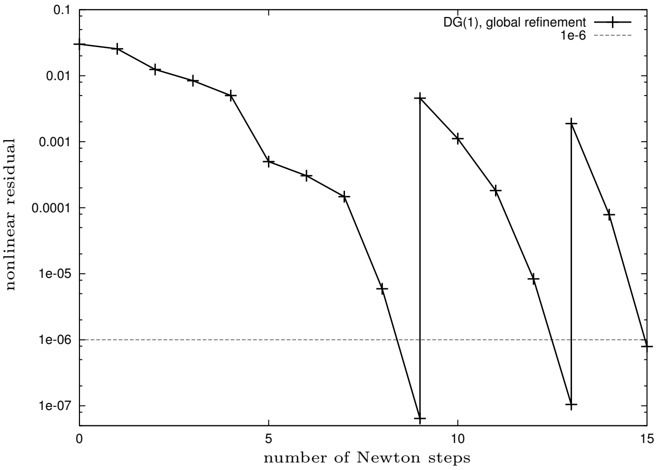


Figure 4. Convergence of the non-linear residual under global refinement with $p = 1$.

Table 2. Number of Newton steps on each global refinement level for DG(p), $1 \leq p \leq 3$. (* pre-iteration on DG(1))

mesh	DG(1)	DG(2)	DG(3)
1	9	4^*	4^*
2	4	2	2
3	2	2	

respectively, where $c_{\rm dp}$ and $c_{\rm lp}$ are the pressure induced force coefficients given by

$$J_{c_{\rm dp}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_d)\,\mathrm{d}s, \qquad J_{c_{\rm lp}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S p\,(\mathbf{n}\cdot\psi_l)\,\mathrm{d}s,$$

respectively, and $c_{\rm df}$ and $c_{\rm lf}$ are the viscous force coefficients, defined by

$$J_{c_{\rm df}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_d\,\mathrm{d}s, \qquad J_{c_{\rm lf}}(\mathbf{u}) = \frac{2}{\bar{l}\bar{\rho}|\bar{\mathbf{v}}|^2}\int_S (\underline{\tau}\,\mathbf{n})\cdot\psi_l\,\mathrm{d}s,$$

respectively. Here, S denotes the surface of the airfoil, $\bar{l}$ its chord length (equal to one), $\bar{\mathbf{v}}$ and $\bar{\rho}$ are the reference (or free-stream) velocity and density, respectively, $(\underline{\tau}\,\mathbf{n})\cdot\psi = \tau_{ij}n_j\psi_i$, where τ is the viscous stress tensor defined in (5) and

$$\psi_d = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \psi_l = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Given that this flow is symmetric about the x-axis the lift coefficients $c_{\rm lp}$ and $c_{\rm lf}$ are both zero. On the basis of fine grid computations, reference values of the drag coefficients are given by $J_{c_{\rm dp}}(\mathbf{u}) \approx 0.0222875$ and $J_{c_{\rm df}}(\mathbf{u}) \approx 0.032535$.

In Figure 5 we plot the Mach isolines of the computed numerical approximation using the proposed interior penalty DGFEM on both the coarsest mesh and the mesh once refined for $p = 1, 2, 3$. Here, we clearly see that as the mesh is refined

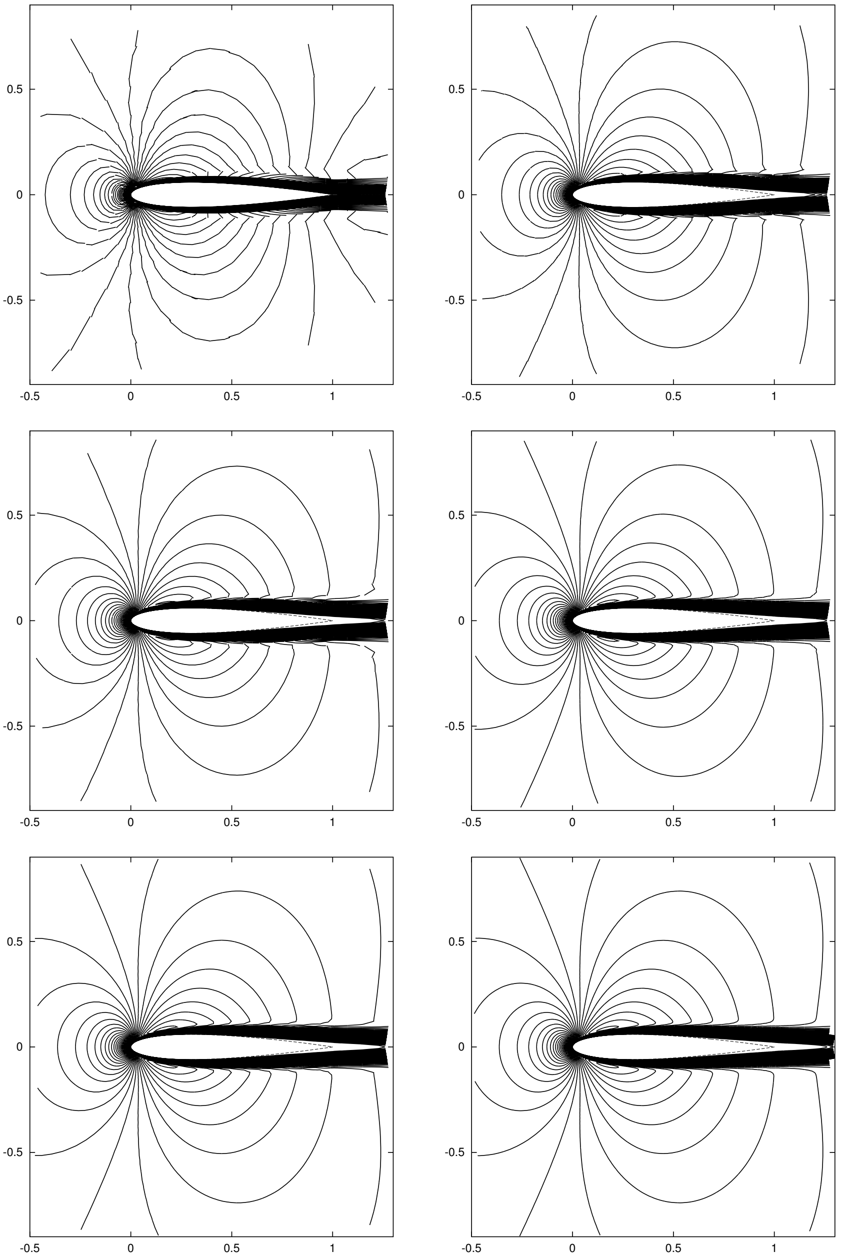
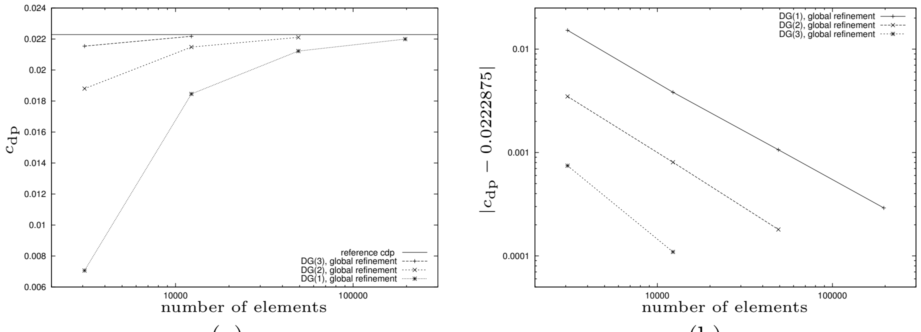


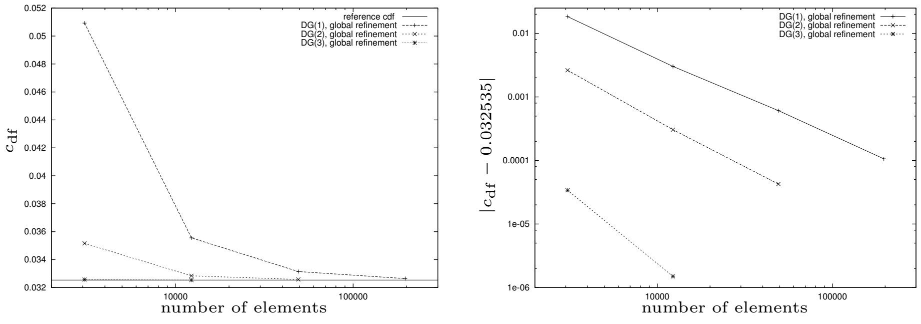
Figure 5. Mach isolines of the DG(p) solutions for (top) $p = 1$, (middle) $p = 2$, (bottom) $p = 3$ on (left) the coarsest mesh with 3072 elements and (right) on the mesh once refined with 12288 elements.

and the polynomial degree is increased, the quality of the numerical approximation significantly improves. Indeed, from Figures 6 and 7, we observe that the error



(a) (b)

FIGURE 6. Convergence of c_{dp} under global refinement for DG(p), $p = 1, 2, 3$: (a) c_{dp} versus number of elements; (b) Error in c_{dp} (reference $c_{\mathrm{dp}} - c_{\mathrm{dp}}$) versus number of elements.



(a) (b)

FIGURE 7. Convergence of c_{df} under global refinement for DG(p), $p = 1, 2, 3$: (a) c_{df} versus number of elements; (b) Error in c_{df} ($c_{\mathrm{df}}-$ reference c_{df}) versus number of elements.

in the approximation to both c_{dp} and c_{df}, respectively, decreases when either the mesh is globally refined, or the polynomial degree is uniformly increased.

Finally, in Figure 8 we compare the error in the approximations to c_{dp} using global mesh refinement with an adaptive mesh refinement strategy employing residual-based error indicators together with the fixed fraction strategy (with refinement and derefinement fractions set to 20% and 10%, respectively), cf. [20], for details. We see from Figure 8 that the gain in accuracy in the approximation of c_{dp} using local refinement versus global refinement significantly increases as the polynomial degree is increased, which is expected for this smooth problem. Indeed, by employing local variation of the polynomial degree, depending on the local smoothness of the solution, in addition to local mesh refinement, i.e., hp-refinement, should be extremely efficient for this problem; this will be investigated as part of our programme of future research.

5.2. Flow over a flat plate. As we saw in the previous section, an increase in the polynomial degree leads to a dramatic improvement in the accuracy of the

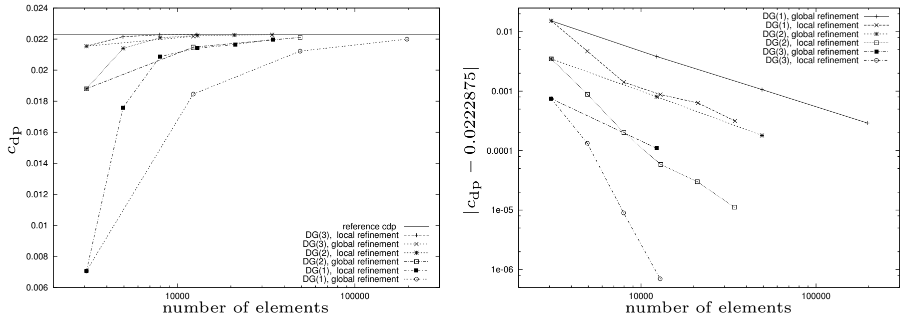


FIGURE 8. Convergence of c_{dp} under global mesh refinement and local mesh refinement using residual-based indicators.

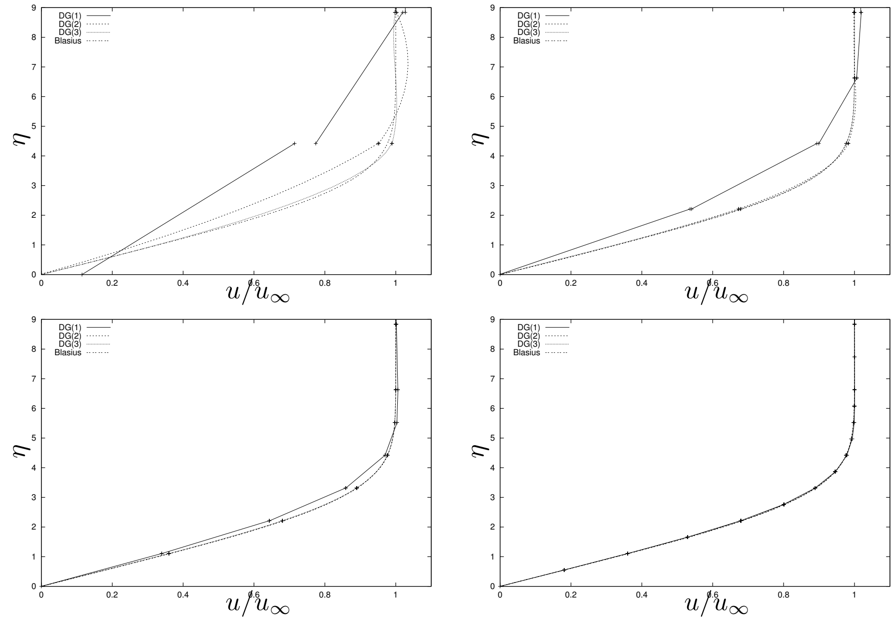


FIGURE 9. DG(p), $1 \le p \le 3$, solutions in comparison with the Blasius solution ($\eta = y\sqrt{u_\infty/(\nu x)} = \frac{y}{x}\sqrt{Re_x}$ versus u/u_∞) on a sequence of meshes with an increasing number of elements.

computed force coefficients c_{dp} and (even more pronounced for) c_{df}, cf. Figures 6 and 7, respectively. To a large extent this improvement in accuracy for these force coefficients can be attributed to the discretization's ability to accurately resolve the viscous boundary layer in the vicinity of the airfoil profile.

Thereby, in order to gain additional insight into how well the proposed interior penalty DGFEM can approximate boundary layers as the polynomial degree is increased, in this example we consider the flow over a flat plate. To this end, we consider a Mach 0.01 flow with Reynolds number 10000 horizontally passing over a flat plate of length $l = 2$. The boundary layer solution to this problem can be

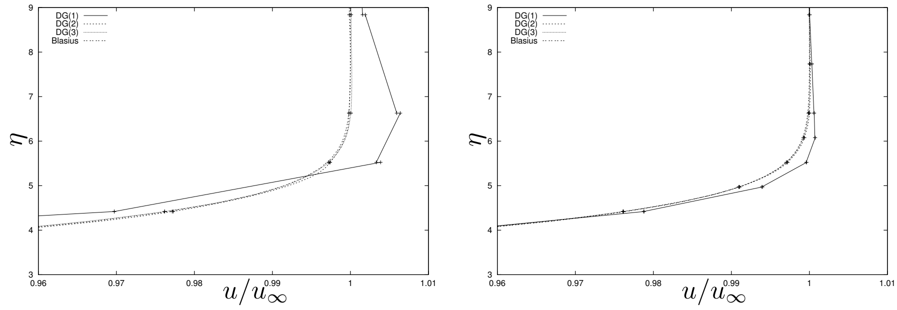


FIGURE 10. Zoom of the DG(p), $1 \leq p \leq 3$, solutions with the Blasius solution on the two finest grids.

TABLE 3. Number of elements and degrees of freedom in the boundary layer required by DG(p), $1 \leq p \leq 3$, discretizations for approximating the viscous force up to 5%.

	DG(1)	DG(2)	DG(3)
elements	36	5	3
DoFs	72	15	12

approximated using Blasius' solution, see [27], for example. In Figure 9, we compare the numerical solution computed with our DGFEM for $1 \leq p \leq 3$, at $x = \frac{l}{2} = 1$ and a local Reynolds number $Re_x = 5000$, with the Blasius solution ($\eta = y\sqrt{u_\infty/(\nu x)} = \frac{y}{x}\sqrt{Re_x}$ versus u/u_∞, cf. [27]) on a sequence of rather coarse computational meshes. On the coarsest mesh, which has about one or two elements within the boundary layer, we see that the DGFEM solution computed with $p = 1, 2$ are not very close to the Blasius solution; increasing the polynomial order to $p = 3$ clearly yields a dramatic improvement in the underlying computed numerical solution. On the next finer mesh, where three elements are placed within the boundary layer, the bilinear approximation is still not very accurate, though now both the computed solution with $p = 2, 3$ are in excellent agreement with the Blasius solution. On the subsequent two meshes we clearly observe that the DGFEM approximation with bilinear elements ($p = 1$) finally starts to coincide with the Blasius solution, at least on a macroscopic level. A more detailed view of the numerical solution on these latter two finer meshes is shown in the zoom depicted in Figure 10. Here, we see that there is still a significant difference between the Blasius solution and the computed DGFEM solution with $p = 1$. Indeed, these figures clearly highlight the substantial gains in accuracy attained when higher–order polynomial degrees are employed with the proposed DGFEM. This is further highlighted in Table 3, where we summarize the number of elements and the number of degrees of freedom, orthogonal to the wall, which are required by the DGFEM for each polynomial degree in order to resolve the boundary layer to a sufficient accuracy that the error in computed viscous stress forces exerted on the wall are within 5% of that computed with the Blasius solution.

6. Concluding Remarks

In this article we have formulated a discontinuous Galerkin finite element discretization of the compressible Navier–Stokes equations, based on employing the generalization of the symmetric version of the interior penalty method for the numerical approximation of the viscous terms. The resulting nonlinear equations are solved by exploiting a damped Newton–GMRES algorithm using a suitable approximation of the Jacobian matrix. Comparisons between different preconditioning strategies and different polynomial degrees have been presented. In particular, we have demonstrated that by employing higher–order elements, remarkably accurate solutions may be computed on relatively coarse meshes. Indeed, by increasing the polynomial order, boundary layers can be approximated extremely well with significantly fewer degrees of freedom in the orthogonal direction to the boundary, than in the case when the lowest–order bilinear elements are employed. In the companion–article [20], we consider the design of automatic adaptive mesh refinement algorithms based on exploiting goal–oriented *a posteriori* error estimation; the extension of this work to the *hp*–version of the finite element method will be considered as part of our programme of future research.

Acknowledgments

Paul Houston acknowledges the financial support of the EPSRC (GR/R76615). All computations have been performed using a DG flow solver based on the `deal.II` library [2].

References

[1] D. Arnold, F. Brezzi, B. Cockburn and D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems.* SIAM J. Namer. Anal. 39(5):1749–1779, 2002.

[2] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II` *Differential Equations Analysis Library, Technical Reference.* http://www.dealii.org/, 5.1 edition, Dec. 2004. first edition 1999.

[3] C.E. Baumann and J.T. Oden, *A discontinuous hp finite element method for the solution of the Euler and Navier–Stokes equations.* Int. J. Numer. Meth. Fluids 31:79–95, 1999.

[4] F. Bassi and S. Rebay, *High–order accurate discontinuous finite element solution of the numerical solution of the compressible Navier–Stokes equations.* J. Comput. Phys. 131:267–279, 1997.

[5] F. Bassi and S. Rebay, *High-order accurate discontinuous finite element solution of the* 2*D Euler equations.* J. Comput. Phys., 138:251–285, 1997.

[6] R. Becker and R. Rannacher, *An optimal control approach to a-posteriori error estimation in finite element methods.* In A. Iserles, editor, Acta Numerica, pp. 1–102, CUP, 2001.

[7] P.J. Capon, *Adaptive Stable Finite Element Methods for the Compressible Navier–Stokes Equations.* Ph.D. Thesis, University of Leeds, 1995.

[8] B. Cockburn and C.-W. Shu, *The Runge–Kutta discontinuous Galerkin finite element method for conservation laws* V: *Multidimensional systems.* J. Comput. Phys., 141:199–224, 1998.

[9] V. Dolejší, *On the discontinuous Galerkin method for the numerical solution of the Euler and the Navier–Stokes equations.* Int. J. Numer. Methods Fluids (submitted).

[10] V. Dolejší and M. Feistauer, *Discontinuous Galerkin finite element method for convection–diffusion problems and the compressible Navier–Stokes equations.* Appl. Numer. Math. (submitted).

[11] V. Dolejší, M. Feistauer and Ch. Schwab, *On discontinuous Galerkin methods for nonlinear convection–diffusion problems and compressible flow.* Mathematica Bohemica, 127(2):163–179, 2002.

[12] M. Feistauer, J. Felcman and I. Straškraba, *Mathematical and Computational Methods for Compressible Flow.* Clarendon Press, Oxford, 2003.

[13] E.H. Georgoulis and E. Süli, *hp–Version interior penalty discontinuous Galerkin finite element methods on anisotropic meshes.* Submitted for publication, 2004.

[14] K. Harriman, P. Houston, B. Senior and E. Süli, *hp–Version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form.* In C.-W. Shu, T. Tang, and S.-Y. Cheng, editors, *Proceedings of the International Conference on Scientific Computing and Partial Differential Equations.* AMS Contemporary Mathematics 330:89–119, 2003.

[15] R. Hartmann *The role of the Jacobian in the adaptive discontinuous Galerkin method for the compressible Euler equations.* In G. Warnecke, editor, Analysis and Numerics for Conservation Laws, Springer, 2005. To appear.

[16] R. Hartmann and P. Houston, *Adaptive Discontinuous Galerkin Finite Element Methods for Nonlinear Hyperbolic Conservation Laws.* SIAM J. Sci. Comp., 24(3):979–1004, 2002.

[17] R. Hartmann and P. Houston, *Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations.* J. Comp. Phys., 183:508–532, 2002.

[18] R. Hartmann and P. Houston, *Goal–oriented a posteriori error estimation for multiple target functionals.* In T.Y. Hou and E. Tadmor, editors, Hyperbolic Problems: Theory, Numerics, Applications, pp. 579-588, Springer-Verlag, 2003.

[19] R. Hartmann and P. Houston, *Adaptive discontinuous Galerkin finite element methods with interior penalty for the compressible Navier–Stokes equations.* In M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, editors, Numerical Mathematics and Advanced Applications, ENUMATH 2003, pp. 410-419, Springer-Verlag, 2004.

[20] R. Hartmann and P. Houston, *Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation.* In preparation.

[21] P. Houston, R. Hartmann and E. Süli, *Adaptive discontinuous Galerkin finite element methods for compressible fluid flows.* In M. Baines, editor, Numerical Methods for Fluid Dynamics VII, ICFD, pp. 347-353, 2001.

[22] P. Houston, Ch. Schwab and E. Süli, *Discontinuous hp–finite element methods for advection–diffusion–reaction problems.* SIAM J. Numer. Anal., 39(6):2133–2163, 2002.

[23] D. Kröner, *Numerical Schemes for Conservation Laws*, Wiley-Teubner, Stüttgart, 1997.

[24] M.G. Larson and T.J. Barth, *A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems.* In Discontinuous Galerkin Methods, B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, Lecture Notes in Comput. Sci. Engrg., 11, Springer, Berlin, 2000, pp. 363–368.

[25] J.T. Oden, I. Babuška, and C. Baumann, *A discontinuous hp-FEM for diffusion problems*, J. Comput. Phys., 146:491–519, 1998.

[26] 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.

[27] H. Schlichting and K. Gersten, *Boundary-Layer Theory*, Springer 2003.

[28] E. Süli and P. Houston, *Adaptive finite element approximation of hyperbolic problems.* In T. Barth and H. Deconinck, editors, Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics. Lecture Notes in Computational Science and Engineering, Volume 25, pp. 269-344, Springer-Verlag, 2002.

[29] E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, Berlin, 1997.

Institute of Aerodynamics and Flow Technology, German Aerospace Center, Lilienthalplatz 7, 38108 Braunschweig, Germany.
E-mail: `Ralf.Hartmann@dlr.de`

School of Mathematical Sciences, University of Nottingham, University Park, Nottingham, NG7 2RD, UK.
E-mail: `Paul.Houston@nottingham.ac.uk`