

An Optimal Order Interior Penalty Discontinuous Galerkin Discretization of the Compressible Navier–Stokes Equations

Ralf Hartmann

Institute of Aerodynamics and Flow Technology, DLR (German Aerospace Center), Lilienthalplatz 7, 38108 Braunschweig, Germany. Ralf.Hartmann@dlr.de

Paul Houston

School of Mathematical Sciences, University of Nottingham, University Park, Nottingham, NG7 2RD, UK. Paul.Houston@nottingham.ac.uk

Abstract

In this article we propose a new symmetric version of the interior penalty discontinuous Galerkin finite element method for the numerical approximation of the compressible Navier–Stokes equations. Here, particular emphasis is devoted to the construction of an optimal numerical method for the evaluation of certain target functionals of practical interest, such as the lift and drag coefficients of a body immersed in a viscous fluid. With this in mind, the key ingredients in the construction of the method include: (i) An adjoint consistent imposition of the boundary conditions; (ii) An adjoint consistent reformulation of the underlying target functional of practical interest; (iii) Design of appropriate interior–penalty stabilization terms. Numerical experiments presented within this article clearly indicate the optimality of the proposed method when the error is measured in terms of both the L_2 -norm, as well as for certain target functionals. Computational comparisons with other discontinuous Galerkin schemes proposed in the literature, including the second scheme of Bassi & Rebay, cf. [12,13], the standard SIPG method outlined in [27], and an NIPG variant of the new scheme will be undertaken.

Key words: Finite element methods, discontinuous Galerkin methods, adjoint consistency, compressible Navier–Stokes equations

1 Introduction

The ever-increasing range of applications of compressible fluid dynamics is a fertile source of difficult and challenging problems with important implications

Preprint submitted to Elsevier

in engineering design. In recent years there has been tremendous interest in the mathematical development and practical implementation of discontinuous Galerkin finite element methods (DGFEMs, for short) for the discretization of compressible fluid flow problems; see, for example, [8,10,11,14,16–18,27,28,33,36] 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.

The construction of DGFEMs is based on employing a test and trial space comprising discontinuous piecewise polynomial functions of a given degree. Information regarding the numerical solution is then communicated between each element through the introduction of suitable numerical flux functions defined on the faces of each element in the mesh. We remark that boundary conditions are typically imposed in a similar fashion. The choice of the numerical flux functions employed then determines the stability and accuracy properties, etc, of the underlying scheme. For the discretization of transport terms, numerical flux functions employed within the finite volume community are typically employed; in the linear setting, most of these fluxes simply lead to a straight-forward upwinding of the underlying hyperbolic partial differential operator; see, for example, [7], and the references cited therein.

The discretization of second-order partial differential operators by the DGFEM has proved more challenging in many ways, particularly in the context of nonlinear diffusion problems. For a recent review of many of the approaches proposed in the literature for the discretization of linear second-order elliptic partial differential equations, we refer to [1]. In this case, the DGFEM is usually constructed by first rewriting the underlying second-order partial differential equation as a first-order system of equations by introducing additional auxiliary variables; the resulting system is then discretized by restricting the test and trial functions to come from the DGFEM finite element space, together with the introduction of suitable numerical flux functions. The resulting discretized system of equations is often referred to as the *flux* formulation of the DGFEM. The flux formulation involves unknowns which approximate the solution of the original partial differential equation, together with the approximation of the auxiliary variables. Thereby, for computational efficiency, it is usually desirable to eliminate the auxiliary variables introduced en route to the derivation of the discretization scheme. This is typically done through the introduction of suitable *lifting operators* which leads to the so-called primal formulation of the DGFEM, cf. [1]. In general, the computation of the lifting operator present in the primal formulation of the underlying scheme requires the inversion of local mass matrix problems on each face present in the computational mesh; see below for the definition of the second scheme

(BR2) proposed by Bassi & Rebay [12,13].

The computation of the lifting operator present within many DGFEMs is typically quite expensive; indeed, for a nonlinear problem, around 30% of the computational effort required for the evaluation of the nonlinear residual is devoted to the computation of the lifting operator. However, it is worth noting that for one prominent class of DGFEMs, referred to as the interior penalty DGFEMs, the lifting operator may be explicitly evaluated; indeed, here the lifting operator simply reduces to the identity operator. Thus, in principle, the exploitation of the interior penalty DGFEM is particularly appealing for large scale CFD applications. In the context of the laminar compressible Navier–Stokes equations, interior penalty DGFEMs have been developed in our earlier articles [27,28], for example. However, as we shall see later in this article, in the context of duality–based error estimation, the symmetric version of the interior penalty DGFEM proposed in [27] may not lead to optimal rates of convergence as the mesh size tends to zero.

Building on the techniques outlined in our previous articles [25,27,28], the aim of this paper is to propose a new alternative interior penalty DGFEM for the numerical approximation of the compressible Navier–Stokes equations which leads to computationally optimal orders of convergence when the error is computed in terms of both the L_2 -norm, as well as for certain target functionals of the solution of practical interest. Error control in this latter sense is particularly important in engineering applications; indeed, in CFD one is often concerned with calculating the lift and drag coefficients of a body immersed into a viscous fluid whose flow is governed by the Navier–Stokes equations. The lift and drag coefficients are defined as integrals, over the boundary of the body, of the viscous and pressure induced forces normal and tangential to the flow, respectively. With this aim in mind, we note that there are essentially three key ingredients in the design of the underlying interior penalty DGFEM to ensure the optimality of the resulting scheme:

- (1) Adjoint consistent discretization including an adjoint consistent imposition of the boundary conditions;
- (2) Adjoint consistent reformulation of the underlying target functional of practical interest;
- (3) Design of appropriate interior–penalty stabilization terms.

The use of adjoint consistent, or at least asymptotically adjoint consistent, numerical schemes is of paramount importance for duality–based error estimation, cf. [1,23,29]. Indeed, the use of, for example, the nonsymmetric interior penalty method for the numerical approximation of the Poisson’s equation leads to suboptimal orders of convergence when the error is measured in terms of both the L_2 -norm, as well as for target functionals of the solution, cf. [23]. We should point out that the use of adjoint consistent numerical fluxes within

the design of DG methods is a very natural requirement which is already employed within a number of schemes proposed within the literature; see [1], for example, and the references cited therein. In the context of target functional error estimation, it is frequently the case that the functional may be expressed in a variety of different ways which are all equivalent on the continuous level, but may lead to quite different numerical approximations under discretization. Thereby, a suitable adjoint consistent reformulation of the functional under consideration may also be required for purposes of optimality. This was first investigated for DGFEMs in the article [22], though we remark that this idea had been previously developed in the context of conforming finite element methods in the articles [2–4], for example. In this context, we point out that when considering the estimation of, for example, the flux on a portion of the boundary of a second-order elliptic partial differential equation, the adjoint consistent reformulation of the target functional under consideration is performed in a fundamentally different manner depending on whether a DGFEM or a conforming Galerkin finite element method is employed. Indeed, in the former case, the adjoint consistent modification of this target functional still leads to an integral on the corresponding boundary of the computational domain. In contrast, for the case when a continuous Galerkin finite element approximation is employed, the boundary flux must be reformulated as a volume integration over all the elements in the computational mesh which neighbor the boundary of interest, to ensure optimal convergence of the underlying target functional as the mesh is uniformly refined. The extension of these ideas to the discontinuous Galerkin discretization of compressible fluid flows has also been developed in the recent articles [34] and [25]. Finally, in the context of interior-penalty DGFEMs, the absence of the lifting operator, although advantageous from a computational point of view, means that the natural generalization of the scheme from the discretization of the Laplace operator to nonlinear diffusion problems, leads to the introduction of so-called jump terms in the underlying numerical scheme which penalize discontinuities present in the numerical solution subject to a uniform penalty term weighted appropriately by the polynomial degree and the mesh size; cf. [27] for the discretization of the compressible Navier–Stokes equations and [30] for DGFEMs employed in the numerical approximation of second-order quasi-linear elliptic partial differential equations. However, as we shall see later, this choice of the interior penalty parameter may not lead to optimal orders of convergence, when the error is measured in terms of the L_2 -norm as well as for target functionals of the solution. Thereby, in this article we shall propose an alternative choice which is very much inspired by the corresponding term arising in the BR2 scheme, cf. [12,13], for example, but has the advantage of not requiring the computation of a lifting operator; we point out that a similar expression for the interior penalty parameter has also been proposed in the recent article [35]. Indeed, our numerical experiments will demonstrate that the accuracy of the proposed interior penalty DGFEM is comparable to that of the BR2 method, yet the evaluation of the underlying residual of the scheme is significantly

computationally cheaper due to the absence of the lifting operator.

The outline of this article is as follows. In Section 2 we introduce the two-dimensional compressible Navier–Stokes equations. Section 3 considers the construction of the new symmetric interior penalty DGFEM; here, particular attention will be devoted to both the adjoint consistent imposition of the boundary conditions, as well as the design of appropriate penalty terms. The adjoint consistency of the method with respect to a general target functional of practical interest will be demonstrated in Section 4 using the general framework developed within the article [25]. The computational performance of the proposed method will be studied in Section 5 on sequences of both uniformly and adaptively refined computational meshes. Finally, in Section 6 we summarize the work presented in this paper and draw some conclusions.

2 The compressible Navier-Stokes equations

In this section we consider the two-dimensional stationary compressible Navier–Stokes equations

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u})) = 0 \quad \text{in } \Omega, \quad (1)$$

where Ω is an open bounded domain in \mathbb{R}^2 . The vector of conservative variables \mathbf{u} and the convective fluxes $\mathcal{F}^c(\mathbf{u})$ are defined by

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

respectively. Furthermore, the viscous fluxes $\mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = (\mathbf{f}_1^v(\mathbf{u}, \nabla \mathbf{u}), \mathbf{f}_2^v(\mathbf{u}, \nabla \mathbf{u}))$ are defined by

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

Here, ρ , $\mathbf{v} = (v_1, v_2)^\top$, p , E , and T denote the density, velocity vector, pressure, specific total energy, and temperature, respectively. Moreover, \mathcal{K} is the thermal

conductivity coefficient and H is the total enthalpy defined by $H = E + p/\rho$. The pressure is determined by the equation of state of an ideal gas, i.e.,

$$p = (\gamma - 1)\rho(E - \frac{1}{2}\mathbf{v}^2), \quad (2)$$

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, $\gamma = 1.4$. For a Newtonian fluid, the viscous stress tensor is given by

$$\tau = \mu \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^\top - \frac{2}{3}(\nabla \cdot \mathbf{v})I \right),$$

where μ is the dynamic viscosity coefficient; the temperature T is given by $\mathcal{K}T = \frac{\mu\gamma}{Pr} \left(E - \frac{1}{2}\mathbf{v}^2 \right)$, where $Pr = 0.72$ is the Prandtl number.

Writing G to denote the homogeneity tensor, with $G_{ij}(\mathbf{u}) = \partial \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) / \partial u_{x_j}$, for $i, j = 1, 2$, cf. [27], the viscous fluxes may be written in the form $\mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u}) = G_{ij}(\mathbf{u})\partial \mathbf{u} / \partial x_j$, $i = 1, 2$, or more compactly, we may write $\mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = G(\mathbf{u})\nabla \mathbf{u}$.

Given that $\Omega \subset \mathbb{R}^2$ is a bounded region, with boundary Γ , the system of conservation laws (1) must be supplemented by appropriate boundary conditions. For simplicity of presentation, we assume that Γ may be decomposed as follows

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

where $\Gamma_{D,\text{sup}}$, $\Gamma_{D,\text{sub-in}}$, $\Gamma_{D,\text{sub-out}}$, and Γ_W are distinct subsets of Γ representing Dirichlet (supersonic), Dirichlet (subsonic-inflow), Dirichlet (subsonic-outflow), and solid wall boundaries, respectively, cf. [27]. We remark that as in [27,28], Neumann boundary conditions may also be considered; for clarity of presentation, we neglect this case and refer to our earlier articles for details.

Thereby, we may specify the following boundary conditions:

$$\mathcal{B}(\mathbf{u}) = \mathcal{B}(\mathbf{g}_D) \quad \text{on} \quad \Gamma_{D,\text{sup}} \cup \Gamma_{D,\text{sub-in}} \cup \Gamma_{D,\text{sub-out}},$$

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

For solid wall boundaries, we consider *isothermal* and *adiabatic* conditions; to this end, decomposing $\Gamma_W = \Gamma_{\text{iso}} \cup \Gamma_{\text{adia}}$, we set

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

where T_{wall} is a given wall temperature; see [8,10,15,17,19] and the references cited therein for further details.

3 DGFEM discretization of the compressible Navier-Stokes equations

In this section we propose a new discontinuous Galerkin method with interior penalty for the discretization of the compressible Navier–Stokes equations (1). This is based on two key modifications of the original scheme developed in [27]: (i) Implementation of an adjoint consistent formulation of the numerical fluxes on the boundary of the computational domain, cf. [34,25]; (ii) Modification of the interior penalty parameter, which is very much inspired by the definition of the penalty term involving a suitable lifting operator present in the second scheme proposed by Bassi & Rebay [12,13].

First, we begin by introducing some notation. 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}_κ the unit outward normal vector to the boundary $\partial\kappa$, and by h_κ the elemental diameter. Moreover, h denotes the piecewise constant mesh function defined by $h|_\kappa \equiv h_\kappa = \text{diam}(\kappa)$ for all $\kappa \in \mathcal{T}_h$. We assume that each $\kappa \in \mathcal{T}_h$ is an image of a fixed reference element $\hat{\kappa}$, that is, $\kappa = \sigma_\kappa(\hat{\kappa})$ for all $\kappa \in \mathcal{T}_h$, where $\hat{\kappa}$ is the open unit square in \mathbb{R}^2 and σ_κ is a smooth bijective mapping. For elements in the interior of the domain, namely $\partial\kappa \cap \Gamma = \emptyset$, the mapping σ_κ is bilinear; in order to represent curved boundaries in an accurate manner for boundary elements, i.e., $\partial\kappa \cap \Gamma \neq \emptyset$, the mapping σ_κ is constructed based on employing a higher-order polynomial representation of the computational boundary, see [24,31,37], for example.

On the reference element $\hat{\kappa}$ we define the space of tensor product polynomials of degree $p \geq 0$ as follows:

$$\mathcal{Q}_p(\hat{\kappa}) = \text{span} \{ \hat{x}^\alpha : 0 \leq \alpha_i \leq p, 1 \leq i \leq 2 \},$$

where α denotes a multi-index and $\hat{x}^\alpha = \prod_{i=1}^2 \hat{x}_i^{\alpha_i}$. Finally, we introduce the finite element spaces \mathbf{V}_h^p and $\underline{\Sigma}_h^p$ consisting of discontinuous vector-valued and tensor-valued, respectively, tensor product polynomial functions of degree $p \geq 0$, defined by

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

respectively.

An interior edge of \mathcal{T}_h is the (non-empty) one-dimensional interior of $\partial\kappa^+ \cap$

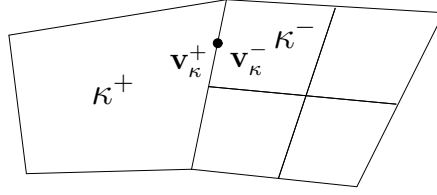


Fig. 1. Definition of the interior and exterior traces \mathbf{v}_κ^\pm with respect to element κ^+ .

$\partial\kappa^-$, 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 . Suppose that $\mathbf{v}|_\kappa \in [H^1(\kappa)]^4$ for each $\kappa \in \mathcal{T}_h$. Let κ^+ and κ^- be two adjacent elements of \mathcal{T}_h and \mathbf{x} an arbitrary point on the interior edge $e = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_{\mathcal{I}}$. By \mathbf{v}_κ^\pm we denote the traces of \mathbf{v} taken from within the interior of κ^\pm , respectively, see Figure 1. Traces of matrix-valued functions $\underline{\mathcal{I}}$ are defined in an analogous manner. Since below it will always be clear from the context which element κ in the subdivision \mathcal{T}_h the quantities \mathbf{v}_κ^+ and \mathbf{v}_κ^- correspond to, for the sake of notational simplicity, we shall suppress the letter κ in the subscript and write, respectively, \mathbf{v}^+ and \mathbf{v}^- , instead.

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

The discontinuous Galerkin discretization of the compressible Navier–Stokes equations (1) is given by: find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$\begin{aligned}
\mathcal{N}(\mathbf{u}_h, \mathbf{v}) \equiv & - \int_{\Omega} \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v} \, d\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) \cdot \mathbf{v}^+ \, ds \\
& + \int_{\Omega} \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) : \nabla_h \mathbf{v} \, d\mathbf{x} - \int_{\Gamma_{\mathcal{I}}} \{\{\mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h)\}\} : \llbracket \mathbf{v} \rrbracket \, ds \\
& - \int_{\Gamma_{\mathcal{I}}} \{\{G^\top(\mathbf{u}_h) \nabla_h \mathbf{v}\}\} : \llbracket \mathbf{u}_h \rrbracket \, ds + \int_{\Gamma_{\mathcal{I}}} \underline{\delta}(\mathbf{u}_h) : \llbracket \mathbf{v} \rrbracket \, ds \\
& + \mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) = 0
\end{aligned} \tag{3}$$

for all \mathbf{v} in \mathbf{V}_h^p . The subscript h on the operator ∇_h is used to denote the broken gradient operator ∇ , defined elementwise. Here, $\mathcal{H}(\cdot, \cdot, \cdot)$ denotes the

(convective) numerical flux function; this may be chosen to be any two-point monotone Lipschitz function which satisfies the following two conditions:

(i) $\mathcal{H}(\cdot, \cdot, \cdot)|_{\partial\kappa}$ is consistent with the flux $\mathcal{F}^c(\cdot) \cdot \mathbf{n}$ for each κ in \mathcal{T}_h , i.e.,

$$\mathcal{H}(\mathbf{v}, \mathbf{v}, \mathbf{n})|_{\partial\kappa} = \mathcal{F}^c(\mathbf{v}) \cdot \mathbf{n} \quad \forall \kappa \in \mathcal{T}_h;$$

(ii) $\mathcal{H}(\cdot, \cdot, \cdot)$ is conservative, i.e., given any two neighboring elements κ^+ and κ^- from the finite element partition \mathcal{T}_h , at each point $\mathbf{x} \in \partial\kappa^+ \cap \partial\kappa^- \neq \emptyset$, noting that $\mathbf{n}_{\kappa^-} = -\mathbf{n}_{\kappa^+}$, we have

$$\mathcal{H}(\mathbf{v}, \mathbf{w}, \mathbf{n}_{\kappa^+}) = -\mathcal{H}(\mathbf{w}, \mathbf{v}, -\mathbf{n}_{\kappa^+}).$$

For the purposes of the numerical experiments presented in Section 5, we employ the Vijayasundaram flux. In this case, $\mathcal{H}(\cdot, \cdot, \cdot)$ is defined by

$$\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa)|_{\partial\kappa} = \mathcal{A}^+(\hat{\mathbf{u}}_h, \mathbf{n}_\kappa)\mathbf{u}_h^+ + \mathcal{A}^-(\hat{\mathbf{u}}_h, \mathbf{n}_\kappa)\mathbf{u}_h^- \quad \text{for } \kappa \in \mathcal{T}_h,$$

where $\mathcal{A}^+(\hat{\mathbf{u}}_h, \mathbf{n}_\kappa)$ and $\mathcal{A}^-(\hat{\mathbf{u}}_h, \mathbf{n}_\kappa)$ denote the positive and negative parts of the Jacobi matrix $\mathcal{A}(\hat{\mathbf{u}}_h, \mathbf{n}_\kappa)$, respectively, evaluated at some average state $\hat{\mathbf{u}}_h$ between \mathbf{u}_h^+ and \mathbf{u}_h^- . Here,

$$\mathcal{A}(\mathbf{u}_h, \mathbf{n}_\kappa) = \mathcal{F}_{\mathbf{u}}^c(\mathbf{u}_h) \cdot \mathbf{n}_\kappa \equiv \sum_{i=1}^2 A_i(\mathbf{u}_h) (\mathbf{n}_\kappa)_i,$$

where A_i , $i = 1, 2$, are the Jacobi matrices of the fluxes \mathbf{f}_i^c , respectively, cf. [32,38].

For an interior edge $e \in \Gamma_{\mathcal{T}}$ the penalization function $\underline{\delta}(\cdot)$ arising in the DGFEM (3) may be defined in a number of ways. For our newly proposed interior penalty DGFEM, we denote $\underline{\delta}(\cdot)$ by $\underline{\delta}^{\text{SIPG}}(\cdot)$, since this represents a generalization of the symmetric interior penalty method developed for linear diffusion problems, cf. [1], and the references cited therein. Indeed, on this basis we shall use the short-hand notation SIPG to refer to this scheme. In Section 5 computational comparisons between our newly proposed SIPG scheme and, what we shall subsequently refer to as the standard interior penalty method proposed in [27], as well as the second DGFEM proposed by Bassi & Rebay (see Remark 3.2 for further details) will be undertaken; in the sequel we shall refer to these two latter schemes as STSIPG and BR2, respectively. Thereby, we shall also introduce the corresponding penalty functions $\underline{\delta}^{\text{STSIPG}}(\cdot)$ and $\underline{\delta}^{\text{BR2}}(\cdot)$, respectively. With this notation we write

$$\underline{\delta}(\mathbf{u}_h) \equiv \begin{cases} \underline{\delta}^{\text{SIPG}}(\mathbf{u}_h) &= C_{\text{IP}} \frac{p^2}{h_e} \{ \{ G(\mathbf{u}_h) \} \} \llbracket \mathbf{u}_h \rrbracket, & \text{for the (new) SIPG scheme,} \\ \underline{\delta}^{\text{STSIPG}}(\mathbf{u}_h) &= C_{\text{IP}} \mu \frac{p^2}{h_e} \llbracket \mathbf{u}_h \rrbracket, & \text{for the STSIPG scheme,} \\ \underline{\delta}^{\text{BR2}}(\mathbf{u}_h) &= C_{\text{BR2}} \{ \{ \underline{L}_0^e(\mathbf{u}_h) \} \}, & \text{for the BR2 scheme,} \end{cases}$$

where C_{IP} and C_{BR2} are positive constants. For reasons of stability, C_{IP} must be chosen sufficiently large. Similarly, C_{BR2} must be chosen to be at least as large as the total number of edges a given element possesses, i.e., in the case of quadrilateral meshes, $C_{\text{BR2}} \geq 4$. We remark that this theoretical minimum is based on the analysis of Poisson's equation, though such a requirement is also likely to be necessary for more complex problems. The local mesh size h_e , defined by

$$h_e = \min(\text{meas}(\kappa^+), \text{meas}(\kappa^-)) / \text{meas}(e),$$

represents the element dimension orthogonal to the edge $e = \partial\kappa^+ \cap \partial\kappa^-$, cf. [20]. Here, we point out that the BR2 scheme requires, for every edge $e \in \Gamma_{\mathcal{T}}$, the computation of the local lifting $\underline{L}_0^e(\mathbf{u}_h) \in \underline{\Sigma}_h^p$ which is defined by

$$\int_{\Omega} \underline{L}_0^e(\mathbf{u}_h) : \underline{\mathcal{T}} \, d\mathbf{x} = \int_e \llbracket \mathbf{u}_h \rrbracket : \{ \{ G^\top(\mathbf{u}_h) \underline{\mathcal{T}} \} \} \, ds \quad \forall \underline{\mathcal{T}} \in \underline{\Sigma}_h^p.$$

Note, that the support of $\underline{L}_0^e(\mathbf{u}_h)$ is $\Omega_e = \kappa_e^+ \cup \kappa_e^-$ with $e = \partial\kappa_e^+ \cap \partial\kappa_e^- \neq \emptyset$.

Finally, it remains to define the boundary terms present in the semilinear form $\mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v})$. To this end, we define

$$\begin{aligned} \mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) &= \int_{\Gamma} \mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}^+) \cdot \mathbf{v}^+ \, ds + \int_{\Gamma} \underline{\delta}_\Gamma(\mathbf{u}_h^+) : \mathbf{v} \otimes \mathbf{n} \, ds, \\ &\quad - \int_{\Gamma} \mathbf{n} \cdot \widehat{\mathcal{F}}^v(\mathbf{u}_\Gamma(\mathbf{u}_h^+), \nabla_h \mathbf{u}_h^+) \mathbf{v}^+ \, ds \\ &\quad - \int_{\Gamma} (\widehat{G}^\top(\mathbf{u}_\Gamma(\mathbf{u}_h^+)) \nabla_h \mathbf{v}_h^+) : (\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \otimes \mathbf{n} \, ds. \end{aligned} \quad (4)$$

Here, the viscous fluxes $\widehat{\mathcal{F}}^v(\mathbf{u}, \nabla \mathbf{u})$ are defined as follows

$$\widehat{\mathcal{F}}^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{cases} \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) & \text{on } \Gamma \setminus \Gamma_{\text{adia}}, \\ \mathcal{F}^{v, \text{adia}}(\mathbf{u}, \nabla \mathbf{u}) & \text{on } \Gamma_{\text{adia}}, \end{cases}$$

where $\mathcal{F}^{v, \text{adia}}(\mathbf{u}, \nabla \mathbf{u})$ is defined so that

$$\mathcal{F}^{v, \text{adia}}(\mathbf{u}, \nabla \mathbf{u}) \cdot \mathbf{n} = (0, \tau_{1j} n_{x_j}, \tau_{2j} n_{x_j}, \tau_{ij} v_j n_{x_i})^\top;$$

thereby, enforcing the adiabatic solid wall boundary condition $\mathbf{n} \cdot \nabla T = 0$ on Γ_{adia} . Here, the summation convention has been assumed, where $\mathbf{n} = (n_{x_1}, n_{x_2})^\top$. Furthermore, the corresponding homogeneity tensor \widehat{G} is defined so that $\widehat{\mathcal{F}}^v(\mathbf{u}, \nabla \mathbf{u}) = \widehat{G}(\mathbf{u}) \nabla \mathbf{u}$.

On the boundary of the computational domain the penalization function $\underline{\delta}_\Gamma(\cdot)$ is defined as follows

$$\underline{\delta}_\Gamma(\mathbf{u}_h) \equiv \begin{cases} \underline{\delta}_\Gamma^{\text{SIPG}}(\mathbf{u}_h) &= C_{\text{IP}} \frac{p^2}{h_e} \widehat{G}(\mathbf{u}_\Gamma(\mathbf{u}_h^+)) (\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)) \otimes \mathbf{n}, \\ \underline{\delta}_\Gamma^{\text{STSIPG}}(\mathbf{u}_h) &= C_{\text{IP}} \mu \frac{p^2}{h_e} (\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)) \otimes \mathbf{n}, \\ \underline{\delta}_\Gamma^{\text{BR2}}(\mathbf{u}_h) &= C_{\text{BR2}} \underline{L}_\Gamma^e(\mathbf{u}_h), \end{cases}$$

for the (new) SIPG, STSIPG, and BR2 schemes, respectively. For a boundary edge $e \subset \Gamma$ the local lifting operator $\underline{L}_\Gamma^e(\mathbf{u}_h) \in \underline{\Sigma}_h^p$ is defined by:

$$\int_\Omega \underline{L}_\Gamma^e(\mathbf{u}_h) : \underline{\mathcal{I}} \, d\mathbf{x} = \int_e (\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)) \otimes \mathbf{n} : \left(\widehat{G}^\top(\mathbf{u}_\Gamma(\mathbf{u}_h)) \underline{\mathcal{I}} \right) \, ds \quad \forall \underline{\mathcal{I}} \in \underline{\Sigma}_h^p.$$

Here, note that the support of $\underline{L}_\Gamma^e(\mathbf{u}_h)$ is κ_e , where κ_e is the element in \mathcal{T}_h such that $\partial\kappa_e \cap \Gamma = e$.

The convective numerical flux function $\mathcal{H}_\Gamma(\cdot, \cdot, \cdot)$ is defined as follows

$$\mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) = \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_\Gamma(\mathbf{u}_h^+)). \quad (5)$$

Finally, the boundary function $\mathbf{u}_\Gamma(\mathbf{u})$ is given according to the type of boundary condition imposed. To this end, we set $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_D$ on $\Gamma_{D,\text{sup}}$, $\mathbf{u}_\Gamma(\mathbf{u}) = ((g_D)_1, (g_D)_2, (g_D)_3, \frac{p(\mathbf{u})}{\gamma-1} + ((g_D)_2^2 + (g_D)_3^2)/(2(g_D)_1))^\top$ on $\Gamma_{D,\text{sub-in}}$, and $\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, u_2, u_3, \frac{p_{\text{out}}}{\gamma-1} + (u_2^2 + u_3^2)/(2u_1))^\top$ on $\Gamma_{D,\text{sub-out}}$. Here, $p \equiv p(\mathbf{u})$ denotes the pressure evaluated using the equation of state (2). On Γ_{iso} , we set $\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_1 c_v T_{\text{wall}})^\top$, while $\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_4)^\top$ on Γ_{adia} .

Remark 3.1 We note that the non-symmetric variant of the interior penalty method, which we will subsequently refer to as the NIPG scheme, is obtained by changing the sign in front of the fifth and fourth terms arising in the semi-linear forms $\mathcal{N}(\cdot, \cdot)$ and $\mathcal{N}_\Gamma(\cdot, \cdot)$, respectively, defined, respectively, in (3) and (4).

Remark 3.2 We point out that the formulation of the second DGFEM proposed by Bassi & Rebay defined above is based on exploiting the so-called flux-based formulation of the scheme which has been developed for laminar flows in the articles [12,13]; this terminology is used to highlight the fact that the underlying lifting operator represents a contribution to the viscous flux function. An alternative gradient-based formulation, where now the associated lifting operator represents a contribution to the gradient of the conservative variables, may be defined by employing the following the penalty function on interior edges

$$\underline{\delta}^{\text{BR2}'}(\mathbf{u}_h) = C_{\text{BR2}'} \{ \{ G(\mathbf{u}_h) \underline{L}_0^{e'}(\mathbf{u}_h) \} \},$$

where, for every edge $e \in \Gamma_{\mathcal{I}}$, the local lifting operator $\underline{L}_0^{e'}(\mathbf{u}_h) \in \underline{\Sigma}_h^p$ is defined by

$$\int_\Omega \underline{L}_0^{e'}(\mathbf{u}_h) : \underline{\mathcal{I}} \, d\mathbf{x} = \int_e \llbracket \mathbf{u}_h \rrbracket : \{ \{ \underline{\mathcal{I}} \} \} \, ds \quad \forall \underline{\mathcal{I}} \in \underline{\Sigma}_h^p;$$

on boundary edges we write

$$\underline{\delta}_\Gamma^{\text{BR2}'}(\mathbf{u}_h) = C_{\text{BR2}} \widehat{G}(\mathbf{u}_\Gamma(\mathbf{u}_h)) \underline{L}_\Gamma^{e'}(\mathbf{u}_h),$$

where

$$\int_\Omega \underline{L}_\Gamma^{e'}(\mathbf{u}_h) : \underline{\tau} \, d\mathbf{x} = \int_e (\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)) \otimes \mathbf{n} : \underline{\tau} \, ds \quad \forall \underline{\tau} \in \underline{\Sigma}_h^p.$$

This latter gradient-based formulation has been exploited for turbulent compressible flows in the article [9]; see also [14] for earlier work on this scheme. We point out, cf. Section 5 below, that for the numerical approximation of the laminar compressible Navier–Stokes equations (1) both the flux-based and gradient-based schemes proposed by Bassi & Rebay are almost identical in terms of accuracy, however, the latter scheme is substantially more expensive, in terms of computational effort, than the flux-based method. Thereby, in the sequel we shall primarily focus on making comparisons between the flux-based scheme (BR2) and the SIPG and STSIPG methods.

Remark 3.3 We remark that for purposes of stability the constant C_{IP} appearing in the definition of the interior penalty terms must be chosen sufficiently large, cf. [1], for example. Computational experience indicates that a value of around 10–20 is sufficient to guarantee stability of the method, without being so large as to adversely affect the conditioning of the resulting system of nonlinear equations. However, in terms of accuracy, the method is relatively insensitive to the magnitude of C_{IP} , provided that C_{IP} is chosen large enough to guarantee stability.

Remark 3.4 As a final remark, we point out that a similar definition for the interior penalty terms for the newly proposed SIPG scheme has also been considered in the recent article [35]. In that paper, the jumps in the numerical solution are scaled by the homogeneity tensor G evaluated at the average of the numerical solution (the averaging being computed between the two states of the solution on neighboring elements), rather than computing the average state of G on a given edge as proposed here. However, the scheme developed in [35] does not impose the boundary conditions in an adjoint consistent manner which is essential for optimality.

4 Consistency and adjoint consistency

In the context of duality-based error estimation, one of the key ingredients needed to ensure that the underlying numerical approximation converges optimally, as the computational mesh is refined, is the so-called *adjoint consistency* property of the discretization scheme employed (assuming, of course, sufficient

regularity of both the underlying primal and dual problems). In the context of DGFEM approximations to Poisson's equation, adjoint consistency has been investigated in [1] for a wide variety of commonly used DGFEMs; see also [22] for related work developed in the context of functional error estimation. The extension of these ideas to DGFEM approximations of compressible fluid flows has been considered in the articles [34,25]. As noted in the Section 1, in the context of target functional error estimation, adjoint consistency of the underlying scheme must also be accompanied by an adjoint consistent reformulation of the underlying target functional of practical interest, cf. [22,34,25]. Indeed, as we shall see in Section 5.2, the accuracy of the underlying scheme may seriously deteriorate if the underlying target functional considered is not reformulated in a suitable manner. However, we remark that this reformulation of the target functional is not essential in all cases, cf. Section 5.1, where the weighted mean-value of the density is considered.

The purpose of this section is to demonstrate that the new SIPG scheme proposed in the previous section is adjoint consistent. Here, the analysis will be pursued in the context of estimating the general target functional

$$J(\mathbf{u}) = J_\Omega(\mathbf{u}) + J_\Gamma(\mathbf{u}), \quad (6)$$

where

$$J_\Omega(\mathbf{u}) = \int_\Omega \mathbf{u} \cdot \boldsymbol{\psi}_\Omega \, d\mathbf{x}$$

and

$$J_\Gamma(\mathbf{u}) = \int_\Gamma (p \mathbf{n} - \boldsymbol{\tau} \mathbf{n}) \cdot \boldsymbol{\psi}_\Gamma \, ds. \quad (7)$$

Here, $\boldsymbol{\psi}_\Omega \in [L^2(\Omega)]^4$ and $\boldsymbol{\psi}_\Gamma \in [L^2(\Gamma)]^2$ are given weighting functions defined on the interior and boundary of Ω , respectively. We remark that $J_\Omega(\cdot)$ represents the weighted mean-value functional, while the boundary functional $J_\Gamma(\cdot)$ is of vital importance in the context of aerodynamic flows. Indeed, in this latter case, setting

$$\boldsymbol{\psi}_\Gamma = \begin{cases} \frac{1}{C_\infty} \boldsymbol{\psi} & \text{on } \Gamma_W, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $\boldsymbol{\psi}$ is given by $\boldsymbol{\psi}_d = (\cos(\alpha), \sin(\alpha))^\top$ or $\boldsymbol{\psi}_l = (-\sin(\alpha), \cos(\alpha))^\top$, $J_\Gamma(\cdot)$ defines the drag and lift coefficient, respectively, of a body immersed in a viscous fluid with inlet flow at the angle of attack α . Here, $C_\infty = \frac{1}{2} \gamma p_\infty M_\infty^2 \bar{l} = \frac{1}{2} \gamma \frac{|\mathbf{v}_\infty|^2}{c_\infty^2} p_\infty \bar{l} = \frac{1}{2} \rho_\infty |\mathbf{v}_\infty|^2 \bar{l}$, where M_∞ denotes the Mach number at free-stream conditions, c_∞ is the free-stream speed of sound defined by $c_\infty^2 = \gamma p_\infty / \rho_\infty$, where p_∞ and ρ_∞ denote the freestream pressure and density, respectively, and \bar{l} denotes the reference (chord) length of the body.

Following [25], we note that the continuous adjoint equations to (1) are given

by

$$-(\mathcal{F}_{\mathbf{u}}^c - \mathcal{F}_{\mathbf{u}}^v)^\top \nabla \mathbf{z} - \nabla \cdot ((\mathcal{F}_{\nabla \mathbf{u}}^v)^\top \nabla \mathbf{z}) = \boldsymbol{\psi}_\Omega. \quad (9)$$

subject to the boundary conditions on $\Gamma_W = \Gamma_{\text{iso}} \cup \Gamma_{\text{adia}}$,

$$(\mathbf{n} \cdot (\mathcal{F}_{\mathbf{u}}^c - \mathcal{F}_{\mathbf{u}}^v))^\top \mathbf{z} + \mathbf{n} \cdot ((\mathcal{F}_{\nabla \mathbf{u}}^v)^\top \nabla \mathbf{z}) = \frac{1}{C_\infty} (p_{\mathbf{u}} \mathbf{n} - \underline{\mathcal{I}}_{\mathbf{u}} \mathbf{n}) \cdot \boldsymbol{\psi}, \quad (10)$$

$$(\mathbf{n} \cdot \mathcal{F}_{\nabla \mathbf{u}}^v)^\top \mathbf{z} = \frac{1}{C_\infty} (\underline{\mathcal{I}}_{\nabla \mathbf{u}} \mathbf{n}) \cdot \boldsymbol{\psi}, \quad (11)$$

see [25]. At wall boundaries Γ_W where $\mathbf{v} = (v_1, v_2)^\top = 0$, the normal viscous flux reduces to $\mathbf{n} \cdot \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = (0, (\tau \mathbf{n})_1, (\tau \mathbf{n})_2, \mathbf{n} \cdot \nabla T)^\top$. Thereby, cf. [25], the continuous adjoint boundary conditions of the adjoint problem (9) to the compressible Navier-Stokes equations are given by

$$z_2 = \frac{1}{C_\infty} \psi_1, \quad z_3 = \frac{1}{C_\infty} \psi_2 \quad \text{on } \Gamma_W, \quad z_4 = 0 \quad \text{on } \Gamma_{\text{iso}}, \quad \mathbf{n} \cdot \nabla z_4 = 0 \quad \text{on } \Gamma_{\text{adia}}. \quad (12)$$

We begin by first demonstrating that the proposed SIPG scheme is consistent. To this end, by employing integration by parts in (3) we obtain the primal residual form given by, cf. [25]: find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$\begin{aligned} \int_\Omega \mathbf{R}(\mathbf{u}_h) \cdot \mathbf{v} \, dx + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial \kappa \setminus \Gamma} (\mathbf{r}(\mathbf{u}_h) \cdot \mathbf{v}^+ + \underline{\rho}(\mathbf{u}_h) : \nabla \mathbf{v}^+) \, ds \\ + \int_\Gamma (\mathbf{r}_\Gamma(\mathbf{u}_h) \cdot \mathbf{v}^+ + \underline{\rho}_\Gamma(\mathbf{u}_h) : \nabla \mathbf{v}^+) \, ds = 0 \quad \forall \mathbf{v} \in \mathbf{V}_h^p. \end{aligned} \quad (13)$$

Here, the primal residuals are given by

$$\begin{aligned} \mathbf{R}(\mathbf{u}_h)|_\kappa &= -\nabla \cdot \mathcal{F}^c(\mathbf{u}_h) + \nabla \cdot \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h), \\ \mathbf{r}(\mathbf{u}_h)|_{\partial \kappa \setminus \Gamma} &= \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) - \frac{1}{2} \llbracket \mathcal{F}^v(\mathbf{u}_h, \nabla_h \mathbf{u}_h) \rrbracket - \mathbf{n} \cdot \underline{\delta}(\mathbf{u}_h), \\ \underline{\rho}(\mathbf{u}_h)|_{\partial \kappa \setminus \Gamma} &= \frac{1}{2} \left(G(\mathbf{u}_h) \llbracket \mathbf{u}_h \rrbracket \right)^\top, \\ \mathbf{r}_\Gamma(\mathbf{u}_h)|_{\partial \kappa \cap \Gamma} &= \mathbf{n} \cdot \left(\mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{F}^c(\mathbf{u}_\Gamma(\mathbf{u}_h^+)) - \mathcal{F}^v(\mathbf{u}_h^+, \nabla \mathbf{u}_h^+) \right. \\ &\quad \left. + \widehat{\mathcal{F}}^v(\mathbf{u}_\Gamma(\mathbf{u}_h^+), \nabla \mathbf{u}_h^+) \right) - \mathbf{n} \cdot \underline{\delta}_\Gamma(\mathbf{u}_h), \\ \underline{\rho}_\Gamma(\mathbf{u}_h)|_{\partial \kappa \cap \Gamma} &= \left(G^\top(\mathbf{u}_\Gamma(\mathbf{u}_h^+)) : (\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)) \otimes \mathbf{n} \right)^\top, \end{aligned}$$

for $\kappa \in \mathcal{T}_h$. Exploiting the consistency of the (convective) numerical flux function $\mathcal{H}(\cdot, \cdot, \cdot)$, namely, $\mathcal{H}(\mathbf{u}, \mathbf{u}, \mathbf{n}) = \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u})$, the consistency of the boundary function, i.e., $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$ on Γ and hence $\underline{\delta}(\mathbf{u}) = \underline{\delta}_\Gamma(\mathbf{u}) = 0$, assuming the analytical solution \mathbf{u} to (1) is sufficiently smooth ($\mathbf{u} \in [H^2(\Omega)]^4$, for example), we deduce that

$$\mathbf{R}(\mathbf{u}) = 0, \quad \mathbf{r}(\mathbf{u}) = 0, \quad \underline{\rho}(\mathbf{u}) = 0, \quad \mathbf{r}_\Gamma(\mathbf{u}) = 0, \quad \underline{\rho}_\Gamma(\mathbf{u}) = 0.$$

Thereby, we deduce that the proposed SIPG discretization given in (3) and (4) is consistent.

We now proceed to demonstrate the adjoint consistency of the SIPG scheme (3). To this end, the target functional $J_\Gamma(\cdot)$ defined on the boundary Γ of the computational domain Ω must be replaced by the following modified version, cf. [25]:

$$\tilde{J}_\Gamma(\mathbf{u}_h) = J_\Gamma(\mathbf{u}_\Gamma(\mathbf{u}_h)) + \int_\Gamma \underline{\delta}_\Gamma(\mathbf{u}_h) : \mathbf{z}_\Gamma \otimes \mathbf{n} \, ds, \quad (14)$$

where

$$\mathbf{z}_\Gamma = \frac{1}{C_\infty}(0, \psi_1, \psi_2, 0)^\top,$$

represents the boundary values of the adjoint solution \mathbf{z} . Noting that $\underline{\delta}_\Gamma(\mathbf{u}) = 0$ holds for the analytical solution \mathbf{u} , assuming \mathbf{u} is sufficiently regular, we have $\tilde{J}_\Gamma(\mathbf{u}) = J_\Gamma(\mathbf{u})$, i.e., $\tilde{J}_\Gamma(\cdot)$ in (14) is a consistent modification of $J_\Gamma(\cdot)$.

Rewriting $\mathcal{N}(\mathbf{u}_h, \mathbf{v})$ in (3) in terms of the homogeneity tensor G , recalling (4) and using integration by parts, we see that the discrete adjoint problem: find $\mathbf{z}_h \in \mathbf{V}_h^p$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}, \mathbf{z}_h) = \tilde{J}'[\mathbf{u}_h](\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}_h^p, \quad (15)$$

can be given in adjoint residual form as follows, see [25]: find $\mathbf{z}_h \in \mathbf{V}_h^p$ such that

$$\begin{aligned} \int_\Omega \mathbf{w} \cdot \mathbf{R}^*[\mathbf{u}_h](\mathbf{z}_h) \, dx + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} (\mathbf{w} \cdot \mathbf{r}^*[\mathbf{u}_h](\mathbf{z}_h) + \nabla \mathbf{w} : \underline{\rho}^*[\mathbf{u}_h](\mathbf{z}_h)) \, ds \\ + \int_\Gamma (\mathbf{w} \cdot \mathbf{r}_\Gamma^*[\mathbf{u}_h](\mathbf{z}_h) + \nabla \mathbf{w} : \underline{\rho}_\Gamma^*[\mathbf{u}_h](\mathbf{z}_h)) \, ds = 0 \quad \forall \mathbf{w} \in \mathbf{V}_h^p. \end{aligned} \quad (16)$$

Here, the adjoint residuals are given by

$$\begin{aligned} \mathbf{R}^*[\mathbf{u}_h](\mathbf{z}_h)|_\kappa &= (\mathcal{F}_\mathbf{u}^c(\mathbf{u}_h) - G'[\mathbf{u}_h] \nabla \mathbf{u}_h)^\top \nabla_h \mathbf{z}_h + \nabla_h \cdot (G^\top(\mathbf{u}_h) \nabla_h \mathbf{z}_h), \\ \mathbf{r}^*[\mathbf{u}_h](\mathbf{z}_h)|_{\partial\kappa \setminus \Gamma} &= - (\mathcal{H}'_{\mathbf{u}^+}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+))^\top \llbracket \mathbf{z}_h \rrbracket \cdot \mathbf{n} - \frac{1}{2} \llbracket G^\top(\mathbf{u}_h) \nabla \mathbf{z}_h \rrbracket \\ &\quad - (\underline{\delta}'[\mathbf{u}_h])^\top \llbracket \mathbf{z}_h \rrbracket + \frac{1}{2} (G'[\mathbf{u}_h] \nabla \mathbf{u}_h)^\top \llbracket \mathbf{z}_h \rrbracket \\ &\quad + \frac{1}{2} (G'[\mathbf{u}_h] \llbracket \mathbf{u}_h \rrbracket)^\top \nabla_h \mathbf{z}_h, \\ \underline{\rho}^*[\mathbf{u}_h](\mathbf{z}_h)|_{\partial\kappa \setminus \Gamma} &= \frac{1}{2} G^\top[\mathbf{u}_h] \llbracket \mathbf{z}_h \rrbracket, \end{aligned}$$

for $\kappa \in \mathcal{T}_h$. Recalling that $\mathcal{F}_\mathbf{u}^v = G'[\mathbf{u}] \nabla \mathbf{u}$ and $\mathcal{F}_{\nabla \mathbf{u}}^v = G(\mathbf{u})$ we see that the analytical solution \mathbf{z} to the continuous adjoint problem (9) satisfies $\mathbf{R}^*[\mathbf{u}](\mathbf{z}) = 0$. Furthermore, the adjoint solution \mathbf{z} satisfies $\mathbf{r}^*[\mathbf{u}](\mathbf{z}) = 0$ and $\underline{\rho}^*[\mathbf{u}](\mathbf{z}) = 0$.

The adjoint boundary residuals in (16) on Γ_W are given by

$$\begin{aligned}\mathbf{r}_\Gamma^*[\mathbf{u}_h](\mathbf{z}_h) &= \frac{1}{C_\infty} (p_{\mathbf{u}} \mathbf{n} - \mathcal{L}_{\mathbf{u}} \mathbf{n}) \cdot \boldsymbol{\psi} - \left(\mathbf{n} \cdot \left(\mathcal{F}_{\mathbf{u}}^c(\mathbf{u}_\Gamma(\mathbf{u}_h)) - \widehat{\mathcal{F}}_{\mathbf{u}}^v(\mathbf{u}_\Gamma(\mathbf{u}_h), \nabla \mathbf{u}_h) \right) \right)^\top \mathbf{z}_h \\ &\quad - \mathbf{n} \cdot \left(\widehat{G}^\top(\mathbf{u}_\Gamma(\mathbf{u}_h)) \nabla \mathbf{z}_h \right), \\ \boldsymbol{\rho}_\Gamma^*[\mathbf{u}_h](\mathbf{z}_h) &= - \frac{1}{C_\infty} (\mathcal{L}_{\nabla \mathbf{u}} \mathbf{n}) \cdot \boldsymbol{\psi} + \left(\mathbf{n} \cdot \widehat{\mathcal{F}}_{\nabla \mathbf{u}}^v(\mathbf{u}_\Gamma(\mathbf{u}_h), \nabla \mathbf{u}_h) \right)^\top \mathbf{z}_h.\end{aligned}$$

Recalling (10) and (11) we see that the analytical solutions \mathbf{u} and \mathbf{z} to the primal problem (1) and the continuous adjoint problem (9)-(12), respectively, satisfy $\mathbf{r}_\Gamma^*[\mathbf{u}](\mathbf{z}) = 0$ and $\boldsymbol{\rho}_\Gamma^*[\mathbf{u}](\mathbf{z}) = 0$.

Thereby, in summary we conclude that the proposed SIPG scheme defined in (3) is adjoint consistent with respect to the general target functional $\widetilde{J}(\cdot)$ which is defined in a similar manner to $J(\cdot)$ in (6) with the boundary functional $J_\Gamma(\cdot)$ replaced by $\widetilde{J}_\Gamma(\cdot)$, cf. (14).

Remark 4.1 *We point out that the NIPG variant of the proposed DGFEM, cf. Remark 3.1, is not adjoint consistent, cf. [25]. As we shall see in Section 5 the lack of adjoint consistency of the NIPG method leads to a loss of accuracy when the error is measured in terms of both the $L_2(\Omega)$ -norm, as well as in terms of the given target functional $J(\cdot)$; analogous behavior is also observed in the context of linear advection–diffusion problems in the article [23].*

Remark 4.2 *We remark that although the STSIPG scheme, cf. above, has been shown to be adjoint consistent in the article [25], the penalty terms present in the method penalize all components of the system in an equal manner, which computationally leads to a loss in accuracy. The newly proposed SIPG scheme weights the jump terms in a ‘natural’ fashion by scaling them according to the size of the homogeneity tensor G . In this way, we shall observe computationally in the next section that the resulting scheme converges optimally under mesh refinement when the error is measured in terms of both the L_2 -norm, as well as for certain target functionals.*

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.

5.1 Example 1: Flow in a square domain

In this first example, we consider a simple model problem in order to examine the experimental order of convergence of the proposed interior penalty DGFEM. To this end, we let $\Omega = (0, \pi)^2$, and supplement the compressible Navier–Stokes equations (1) with an inhomogeneous forcing function \mathbf{f} , which is chosen so that the analytical solution to (1) is given by

$$\mathbf{u} = (\sin(2(x+y))+4, \sin(2(x+y))/5+4, \sin(2(x+y))/5+4, (\sin(2(x+y))+4)^2)^\top,$$

where the dynamic viscosity coefficient μ has been set to $1/10$. This represents a modification of the (unsteady) test problem employed in the article [36]. In this section we shall be interested in measuring the discretization error in terms of both the $L_2(\Omega)$ -norm, denoted by $\|\cdot\|_0$, as well as in terms of a given target functional $J(\cdot)$. In the latter case, we consider the weighted mean-value of the density, i.e.,

$$J(\mathbf{u}) \equiv J_\Omega(\mathbf{u}) = \int_\Omega u_1 \psi \, d\mathbf{x},$$

where $\psi = \sin(\pi x) \sin(\pi y)$; thereby, the true value of the functional is given by $J(\mathbf{u}) = 1.168587648689877$.

In Figure 2(a) we present a comparison of the error in the $L_2(\Omega)$ -norm with the (square root of the) number of elements for $p = 1, 2, 3, 4$, employing both the SIPG method with $C_{\text{IP}} = 10$ and the (flux-based) Bassi–Rebay method (BR2) with $C_{\text{BR2}} = 4$. In both cases, we observe that $\|\mathbf{u} - \mathbf{u}_h\|_0$ converges to zero at the expected optimal rate $\mathcal{O}(h^{p+1})$ as the mesh is refined for each fixed p . Moreover, from Figure 2(b) we observe that the error in the computed target functional $J(\cdot)$ behaves (approximately) like $\mathcal{O}(h^{2p})$, for each fixed p , as the mesh is uniformly refined for both of the discretization schemes considered. These rates of convergence for both the $L_2(\Omega)$ -norm of the error and the error in the computed target functional $J(\cdot)$ are in complete agreement with the corresponding convergence behavior we would expect for the SIPG and BR2 methods when applied to a linear convection–diffusion problem; see [23], for example, for the analysis of general interior penalty DGFEMs for second–order partial differential equations with non-negative characteristic form. We remark that in terms of accuracy, for a given number of elements, or equivalently, for a fixed number of degrees of freedom, both the newly proposed SIPG scheme and the BR2 method perform in a comparable manner, with the latter scheme being, in general, slightly more accurate. However, in terms of computational resources, the time required to assemble the residual vector of the BR2 method, which is the most computationally intensive part of the flow solver, when explicit time-stepping schemes are employed, is significantly more expensive than the computation of the corresponding quantity when the SIPG scheme is employed. More precisely, for (bi)-linear, elements, i.e., $p = 1$, the BR2 method is around 38% more expensive than the SIPG scheme; this overhead increases

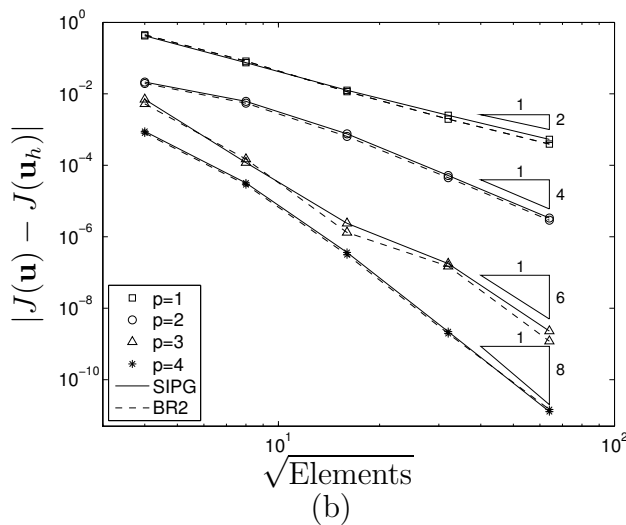
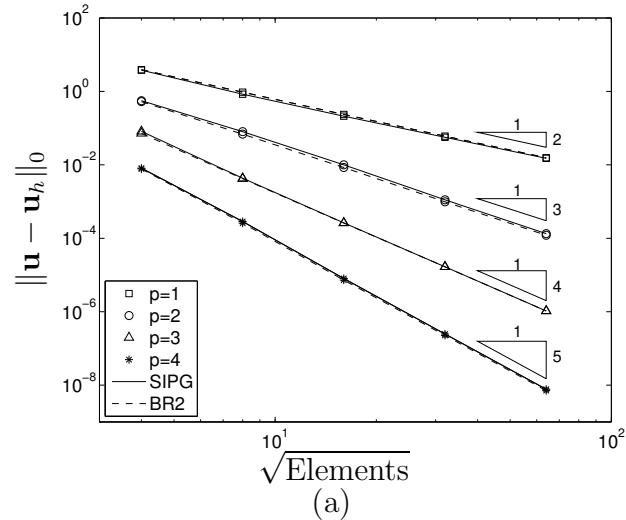


Fig. 2. Example 1. Comparison of the SIPG and BR2 methods when the error is measured in terms of: (a) $L_2(\Omega)$ -norm; (b) Weighted mean-value functional $J(\cdot)$.

as the underlying polynomial degree is enriched. Indeed, for $p = 2$, the BR2 method is approximately 47% more expensive, and for $p = 3$ and $p = 4$ the additional work rises to around 55%. This increase in the cpu times when the BR2 method is employed is attributed to the computation of the lifting operator on each face of the computational mesh. We point that the gradient-based formulation of the BR2 method, cf. Remark 3.2, is almost identical in terms of accuracy when compared to the flux-based formulation employed here, however, this former scheme is significantly less efficient in terms of computational effort. Indeed, comparing with SIPG scheme, the computation of the residual vector of the gradient-based BR2 method is around 78% more expensive than the SIPG method for $p = 1$; for $p = 2$, this overhead increases to 86%, for $p = 3$, it is approximately 107% more expensive, and for $p = 4$, the additional work rises to around 115%.

Finally, in this section we compare the performance of the proposed SIPG

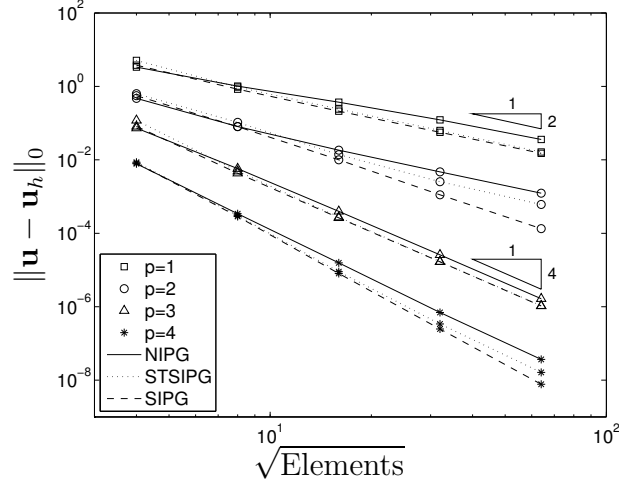


Fig. 3. Example 1. Comparison of the SIPG, NIPG, and STSIPG methods when the error is measured in terms of the $L_2(\Omega)$ -norm.

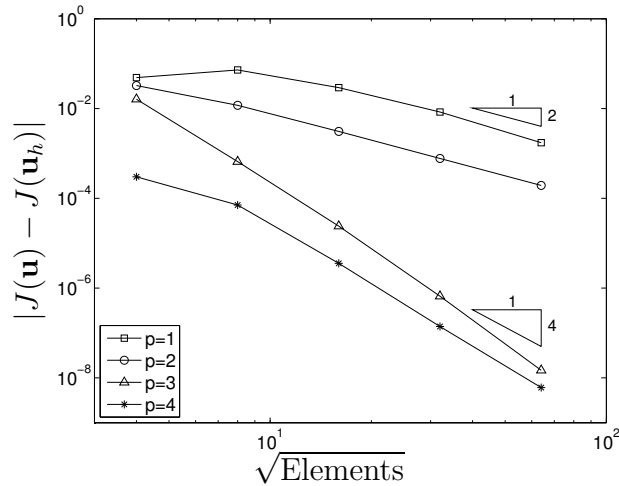


Fig. 4. Example 1. Convergence of the NIPG scheme with respect to $J(\cdot)$ with h -refinement.

method with both the corresponding NIPG formulation of the underlying scheme (cf. Remark 3.1), together with the interior penalty method outlined in our previous article [27]; as in Section 3, we shall refer to this latter scheme as the standard SIPG (STSIPG) method. To this end, in Figure 3 we plot the $L_2(\Omega)$ -norm of the error against the (square root of the) number of elements for $p = 1, 2, 3, 4$ using each of the above schemes. In contrast to the SIPG and BR2 methods, we now observe that $\|\mathbf{u} - \mathbf{u}_h\|_0$ behaves like $\mathcal{O}(h^{p+1})$ for odd p and like $\mathcal{O}(h^p)$ for even p when either the NIPG method or the STSIPG scheme are employed. The sub-optimal convergence observed when employing these two schemes is attributed to the lack of smoothness in the resulting dual problems, cf. [23]. Moreover, the same behavior is also observed in the functional setting; indeed, for the NIPG scheme, from Figure 4 we see that $|J(\mathbf{u}) - J(\mathbf{u}_h)|$ tends to zero at (approximately) the rate $\mathcal{O}(h^{p+1})$ for odd p and $\mathcal{O}(h^p)$ for even p , as the mesh is uniform refined. Analogous behavior is also

observed when the error in the computed target functional $J(\cdot)$ is evaluated using the STSIPG method; for brevity, these numerics have been omitted.

5.2 Example 2: Mach 0.5 flow at $Re = 5000$ and $\alpha = 0^\circ$ around a NACA0012

In this 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

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

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. 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, i.e. $\alpha = 0^\circ$, 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 test case which has been investigated by many other authors, cf. [10,27], for example. The solution to this problem consists of a strictly subsonic flow which is symmetric about the x -axis.

Here, we consider the estimation of the drag coefficient c_d ; i.e., the target functional of interest is given by

$$J(\mathbf{u}) \equiv J_\Gamma(\mathbf{u}) = \int_\Gamma (p \mathbf{n} - \underline{\tau} \mathbf{n}) \cdot \boldsymbol{\psi}_\Gamma ds,$$

where

$$\boldsymbol{\psi}_\Gamma = \begin{cases} \frac{1}{C_\infty} \boldsymbol{\psi}_d & \text{on } \Gamma_W, \\ 0 & \text{otherwise,} \end{cases}$$

and $\boldsymbol{\psi}_d = (\cos(\alpha), \sin(\alpha))^\top$, cf. (7), (8). We remark that the adjoint consistency of the proposed SIPG scheme is based on the consistent reformulation of $J(\cdot)$ defined in (14). With this in mind, in Figure 5(a) we present a comparison of the error in the computed target functional with the (square root of the) number of elements for $p = 1, 2, 3$, employing both the SIPG method with $C_{\text{IP}} = 10$ and the (flux-based) Bassi-Rebay method (BR2) with $C_{\text{BR2}} = 4$. In both cases, we observe that, asymptotically, at least, $|J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h)|$ converges to zero at the expected optimal rate $\mathcal{O}(h^{2p})$ as the mesh is refined for each fixed p , cf. Section 5.1 above. Moreover, as before, we note that in terms of accuracy, for a given number of elements, or equivalently, for a fixed number of degrees of freedom, both the newly proposed SIPG scheme and the BR2 method perform in a comparable manner, though as already noted, the SIPG scheme requires less computational effort to attain the computed solution. As in the previous

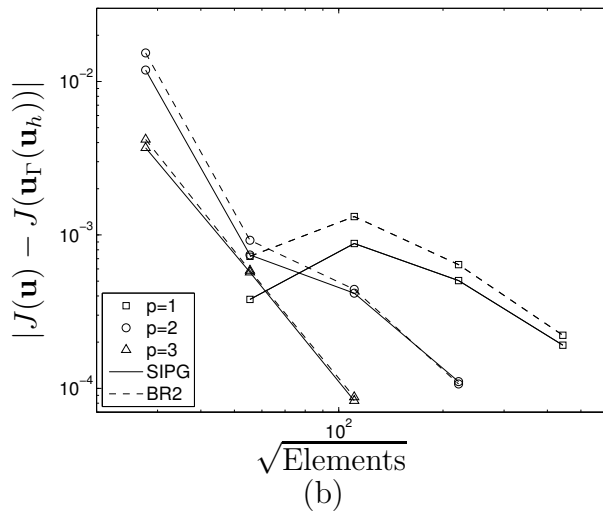
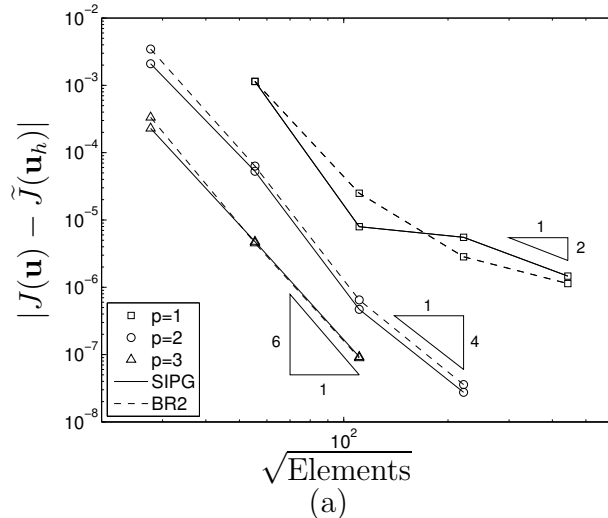
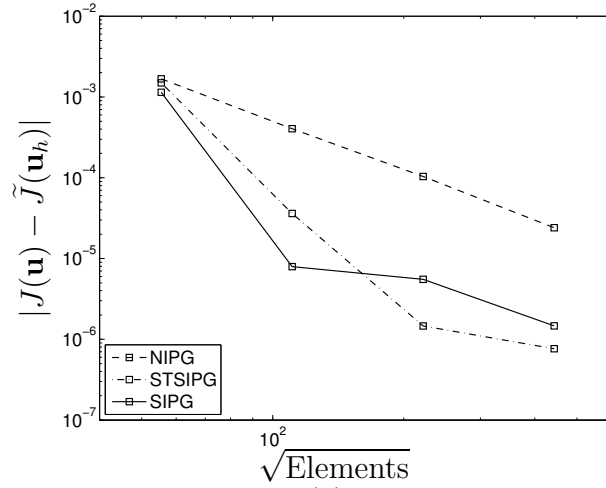
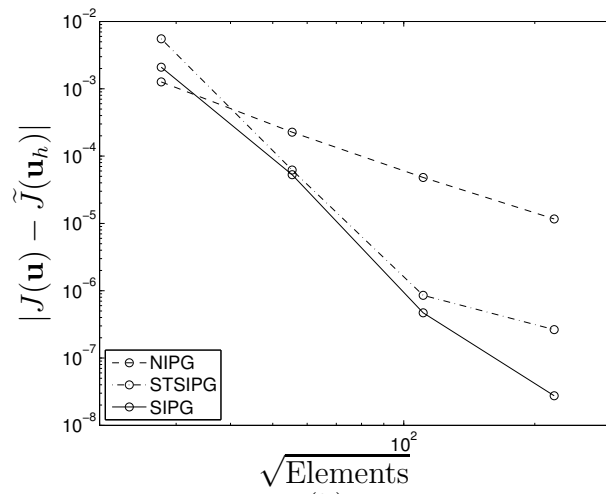


Fig. 5. Example 2. Comparison of the SIPG and BR2 methods employing: (a) Adjoint consistent reformulation of the drag functional; (b) Adjoint consistent reformulation of the drag functional excluding the penalty terms.

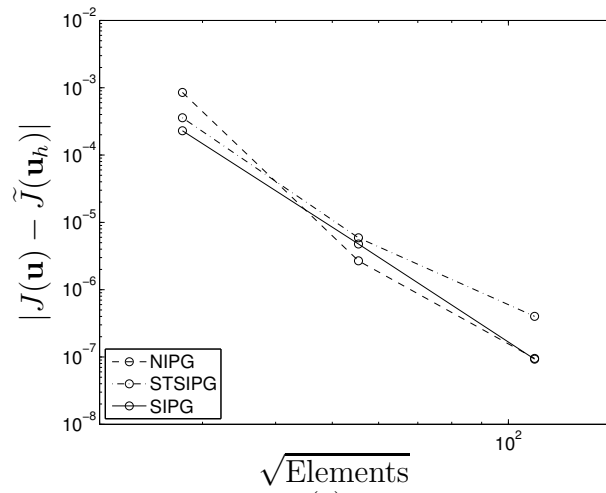
example, we note that the gradient-based formulation of the BR2 method is almost identical in terms of accuracy when compared to the flux-based formulation employed here; though, again it is significantly less efficient in terms of computational effort. To highlight the necessity of the consistent reformulation of the original target functional $J(\cdot)$ through the additional of the term involving the penalty function $\underline{\delta}_\Gamma(\cdot)$, cf. (14) for the definition of $\tilde{J}_\Gamma(\cdot)$, in Figure 5(b) we present a comparison of $|J(\mathbf{u}) - J(\mathbf{u}_\Gamma(\mathbf{u}_h))|$ with the (square root of the) number of elements for $p = 1, 2, 3$ employing both the SIPG and BR2 schemes. In this case, we now observe that there is a significant deterioration of the error for a given mesh size and polynomial order when compared to the corresponding results when the penalty function modification of the target functional has been included. Indeed, comparing Figures 5(a) and 5(b), we see that the inclusion of the penalty function modification in the definition of $\tilde{J}(\cdot) \equiv \tilde{J}_\Gamma(\cdot)$ leads to around 2–3 orders of magnitude improvement in the



(a)



(b)



(c)

Fig. 6. Example 2. Comparison of the SIPG, NIPG, and STSIPG methods: (a) $p = 1$; (b) $p = 2$; (c) $p = 3$.

computed error in the drag.

Finally, in Figure 6 we present a comparison of the newly proposed SIPG method with both the NIPG and STSIPG schemes for $p = 1, 2, 3$. In this case, we see that for $p = 1$ and $p = 2$, the NIPG scheme is inferior to both the SIPG and STSIPG methods; indeed, in this case, the error in the underlying target functional computed using the SIPG method is between 1–2 orders of magnitude smaller than the corresponding quantity evaluated using the NIPG method. For $p = 1$ we observe that the STSIPG method is marginally more accurate than the SIPG method on the two finest meshes employed, though for $p = 2$ this method is inferior to the SIPG scheme. For $p = 3$, the three methods lead to a very similar error in the computed drag coefficient, for a given number of elements.

5.3 Example 3: Mach 0.5 flow at $Re = 5000$ and $\alpha = 2^\circ$ around a NACA0012

In this final example, we investigate the performance of the newly proposed SIPG method on a sequence of adaptively refined meshes generated based on the goal-oriented dual-weighted-residual error indicators $\hat{\eta}_\kappa$ derived in [27]. We remark that although the STSIPG scheme was employed in the article [27], the extension of the *a posteriori* error bound derived in [27] to the application of the DGFEM developed within the current article follows in a similar fashion; for brevity, we omit the details.

To this end, we consider a variant of the problem studied in the previous example. Indeed, here we consider the subsonic viscous flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 2^\circ$, and Reynolds number $Re = 5000$; on the walls of the airfoil geometry, we impose a zero heat flux (adiabatic) no-slip boundary condition. In Tables 1 & 2 we demonstrate the performance of the adaptive algorithm for the numerical approximation of the drag (c_d) and lift (c_l) coefficients, respectively. In each case, we show the number of elements and degrees of freedom (DoF) in \mathbf{V}_h^1 , the true error in the functional $J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h)$, the computed error representation formula $\sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$, and the corresponding effectivity index $\theta = \sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa / (J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h))$. We see that even on the initial very coarse meshes the quality of the computed error representation formula $\sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$ is relatively good, in the sense that θ is close to one; however, as the mesh is refined, we observe that the effectivity indices θ improve by slowly tending towards unity.

Finally, in Figure 7 we show the meshes generated for both the approximation of the drag and lift coefficients after five adaptive refinements, with 5155 and 5236 elements, respectively. As noted in [27], refinement is mainly concentrated within in the vicinity of the airfoil, with the mesh generated for the

Elements	DoF	$J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$	θ
400	6400	1.179e-02	1.385e-02	1.17
655	10480	-1.345e-03	-2.113e-03	1.57
1096	17536	-8.976e-04	-7.871e-04	0.88
1795	28720	-4.282e-04	-3.803e-04	0.89
3028	48448	-2.422e-04	-2.275e-04	0.94
5155	82480	-1.152e-04	-1.104e-04	0.96
8653	138448	-5.481e-05	-4.781e-05	0.87
14584	233344	-3.484e-05	-3.458e-05	0.99

Table 1

Example 3: Adaptive algorithm for the numerical approximation of c_d .

Elements	DoF	$J(\mathbf{u}) - \tilde{J}(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \hat{\eta}_\kappa$	θ
400	6400	-1.175e-01	-5.867e-02	0.50
658	10528	6.548e-03	6.841e-03	1.04
1108	17728	-1.292e-03	-1.159e-03	0.90
1861	29776	-1.784e-03	-1.891e-03	1.06
3118	49888	-1.239e-03	-1.266e-03	1.02
5236	83776	-6.504e-04	-6.704e-04	1.03
8746	139936	-2.623e-04	-2.622e-04	1.00

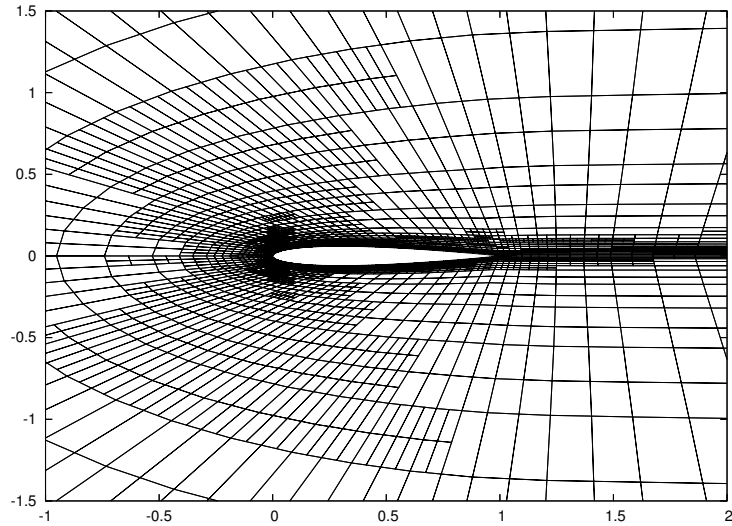
Table 2

Example 3: Adaptive algorithm for the numerical approximation of c_l .

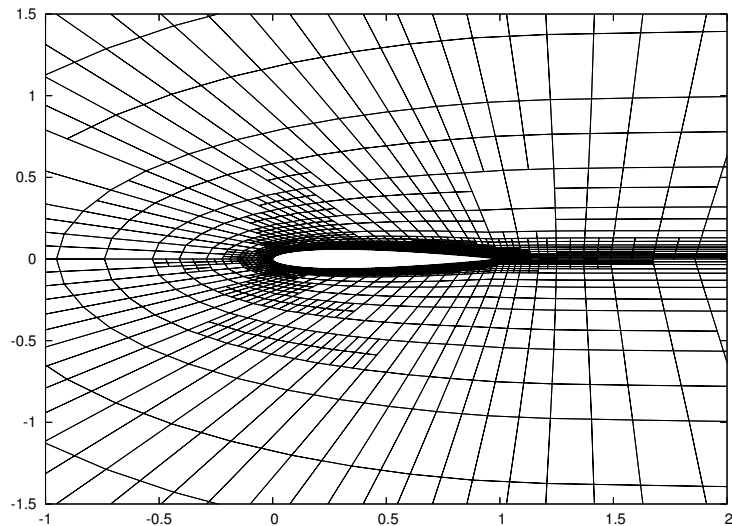
computation of the lift coefficient being more concentrated around the the airfoil, than the corresponding mesh generated for the accurate computation of the drag coefficient.

6 Concluding Remarks

In this article we have considered the formulation of a new symmetric version of the interior penalty method for the numerical approximation of the compressible Navier-Stokes equations. Indeed, in the goal-oriented setting we have shown that the new method is adjoint consistent with respect to certain target functionals of practical interest, based on exploiting the analysis developed in the article [25]. Experimentally, the newly proposed scheme has been shown to yield optimal rates of convergence, when the error is measured in terms of both the L_2 -norm, as well as for certain target functionals of the



(a)



(b)

Fig. 7. Example 3: Adaptively refined computational mesh generated for the accurate approximation of: (a) c_d ; (b) c_l .

solution of practical interest. Future work will be devoted to the application of the scheme to both three-dimensional laminar and turbulent flows.

Acknowledgments

Ralf Hartmann acknowledges the financial support by the President's Initiative and Networking Fund of the Helmholtz Association of German Research Centres. Paul Houston acknowledges the financial support of both the EPSRC, under grant GR/R76615, and the European Union, under the ADIGMA

project. Computations have been performed using the `AptoFEM` software package (see [21] for details) and the DGFEM flow solver `PADGE` [26] based on the `deal.II` library [5,6]. Finally, the authors wish to thank the Aircraft Research Association for providing the initial coarse mesh employed in Example 3.

References

- [1] D. ARNOLD, F. BREZZI, B. COCKBURN AND D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*. SIAM J. Numer. Anal. 39(5):1749–1779, 2002.
- [2] I. BABUŠKA AND A. MILLER, *The post processing approach in the finite element method, Part 1: Calculation of displacements, stresses and other higher derivatives of the displacements*. Int. J. Numer. Methods. Engrg., 34:1085–1109, 1984.
- [3] I. BABUŠKA AND A. MILLER, *The post processing approach in the finite element method, Part 2: The calculation of stress intensity factors*. Int. J. Numer. Methods. Engrg., 34:1111–1129, 1984.
- [4] I. BABUŠKA AND A. MILLER, *The post processing approach in the finite element method, Part 3: A posteriori estimates and adaptive mesh selection*. Int. J. Numer. Methods. Engrg., 34:1131–1151, 1984.
- [5] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II Differential Equations Analysis Library, Technical Reference`. <http://www.dealii.org/>, 6.0 edition, Sept. 2007. first edition 1999.
- [6] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II – A general purpose object oriented finite element library`. *ACM Transactions on Mathematical Software*, 33(4), Aug. 2007.
- [7] T.J. Barth. *An introduction to upwind finite volume and finite element methods: Some unifying and contrasting themes*. VKI Lecture Series on High Order Discretization Methods in CFD, von Karman Institute for Fluid Dynamics, pp. 1-64, 2006.
- [8] 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.
- [9] F. BASSI, A. CRIVELLINI, S. REBAY, M. SAVINI, *Discontinuous Galerkin solution of the Reynolds–averaged Navier–Stokes and k – ω turbulence model equations* Computers & Fluids, 34:507–540, 2005.
- [10] 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.

- [11] F. BASSI AND S. REBAY, *High-order accurate discontinuous finite element solution of the 2D Euler equations*. J. Comput. Phys., 138:251–285, 1997.
- [12] F. BASSI AND S. REBAY, *GMRES Discontinuous Galerkin Solution of the Compressible Navier-Stokes Equations*. In Discontinuous Galerkin Methods, B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, Lecture Notes in Comput. Sci. Engrg., 11, Springer, Berlin, 2000, pp. 197–208.
- [13] F. BASSI AND S. REBAY, *Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations*. Int. J. Numer. Meth. Fluids, 40:197–207, 2002.
- [14] F. BASSI, S. REBAY, G. MARIOTTI, S. PEDINOTTI, AND M. SAVINI, *A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows*. In R. Decuyper and G. Dibelius, editors, *2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, March 5–7, 1997*, pages 99–108. Technologisch Instituut, 1997.
- [15] P.J. CAPON, *Adaptive Stable Finite Element Methods for the Compressible Navier–Stokes Equations*. Ph.D. Thesis, University of Leeds, 1995.
- [16] 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.
- [17] V. DOLEJŠÍ, *On the discontinuous Galerkin method for the numerical solution of the the Navier–Stokes equations*. Int. J. Numer. Methods Fluids, 45:1083–1106, 2004.
- [18] 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.
- [19] M. FEISTAUER, J. FELCMAN AND I. STRAŠKRABA, *Mathematical and Computational Methods for Compressible Flow*. Clarendon Press, Oxford, 2003.
- [20] E.H. GEORGOULIS, E. HALL, AND P. HOUSTON, *Discontinuous Galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes*. SIAM J. Sci. Comp., 30(1):246–271, 2007.
- [21] S. GIANI, E. HALL AND P. HOUSTON, *AptoFEM Users Manual Version 1.0*. Technical report, University of Nottingham, (in preparation).
- [22] K. HARRIMAN, D.J. GAVAGHAN, AND E. SÜLI. The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method. Technical Report NA04/18, Oxford University Computing Laboratory, 2004.
- [23] 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.

- [24] R. Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations*. PhD thesis, University of Heidelberg, 2002.
- [25] R. Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6): 2671–2696, 2007.
- [26] R. Hartmann *et al.* *PADGE Parallel Adaptive Discontinuous Galerkin Environment. Technical Reference*. DLR Braunschweig, 2007. In preparation.
- [27] R. HARTMANN AND P. HOUSTON, *Symmetric Interior Penalty DG Methods for the Compressible Navier–Stokes Equations I: Method formulation*. *Int. J. Num. Anal. Model.*, 3(1):1–20, 2006.
- [28] R. HARTMANN AND P. HOUSTON, *Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal-oriented a posteriori error estimation*. *Int. J. Num. Anal. Model.*, 3(2):141–162, 2006.
- [29] P. HOUSTON, R. RANNACHER, AND E. SÜLI, *A posteriori error analysis for stabilised finite element approximations of transport problems*, *Comput. Methods Appl. Mech. Engrg.*, 190:1483–1508, 2000.
- [30] P. Houston, J. Robson, and E. Süli. Discontinuous Galerkin finite element approximation of quasilinear elliptic boundary value problems I: The scalar case. *IMA J. Numer. Anal.*, 25:726–749, 2005.
- [31] G.E. KARNIADAKIS AND S. SHERWIN, *Spectral/hp Finite Element Methods in CFD*. Oxford University Press, 1999.
- [32] D. KRÖNER, *Numerical Schemes for Conservation Laws*, Wiley-Teubner, Stuttgart, 1997.
- [33] 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.
- [34] J. LU AND D. L. DARMOFAL, *Dual-consistency analysis and error estimation for discontinuous Galerkin discretization: application to first-order conservation laws*, *IMA Journal of Numerical Analysis*, 2006, submitted.
- [35] G. GASSNER, F. LÖRCHER, AND C.-D. MUNZ, *A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes*. *J. Comp. Phys.*, 224(2):1049–1063, 2007.
- [36] G. GASSNER, F. LÖRCHER, AND C.-D. MUNZ, *A discontinuous Galerkin scheme based on a space-time expansion II. Viscous flow equations in multi-dimensions*. *J. Sci. Comp.*, 34(3):260–286, 2008.

- [37] P. ŠOLÍN, K. SEGETH, AND I. DOLEŽEL, *Higher-Order Finite Element Methods*. Chapman & Hall/CRC, 2004.
- [38] E. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, Berlin, 1997.