Journal of Computational Physics 362 (2018) 305–326

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Analysis and development of adjoint-based h-adaptive direct discontinuous Galerkin method for the compressible Navier–Stokes equations

Jian Cheng [a,b], Huiqiang Yue [b,*], Shengjiao Yu [b], Tiegang Liu [b]

[a] Institute of Applied Physics and Computational Mathematics, Beijing, 100088, PR China
[b] LMIB, School of Mathematics and Systems Science, Beihang University, Beijing, 100091, PR China

ARTICLE INFO

Article history:
Received 2 July 2017
Received in revised form 11 November 2017
Accepted 15 February 2018
Available online 21 February 2018

Keywords:
Direct discontinuous Galerkin method
Adjoint-based h-adaptation
Adjoint consistency
Compressible Navier–Stokes equations

ABSTRACT

In this paper, an adjoint-based high-order h-adaptive direct discontinuous Galerkin method is developed and analyzed for the two dimensional steady state compressible Navier–Stokes equations. Particular emphasis is devoted to the analysis of the adjoint consistency for three different direct discontinuous Galerkin discretizations: including the original direct discontinuous Galerkin method (DDG), the direct discontinuous Galerkin method with interface correction (DDG(IC)) and the symmetric direct discontinuous Galerkin method (SDDG). Theoretical analysis shows the extra interface correction term adopted in the DDG(IC) method and the SDDG method plays a key role in preserving the adjoint consistency. To be specific, for the model problem considered in this work, we prove that the original DDG method is not adjoint consistent, while the DDG(IC) method and the SDDG method can be adjoint consistent with appropriate treatment of boundary conditions and correct modifications towards the underlying output functionals. The performance of those three DDG methods is carefully investigated and evaluated through typical test cases. Based on the theoretical analysis, an adjoint-based h-adaptive DDG(IC) method is further developed and evaluated, numerical experiment shows its potential in the applications of adjoint-based adaptation for simulating compressible flows.

1. Introduction

The past decade has witnessed considerable progress in the analysis, development and application of discontinuous Galerkin (DG) methods [1–16], which is commonly regarded as one of the typical representatives in the community of high order numerical methods for computational fluid dynamics. DG methods combine the advantageous features commonly associated with finite element (FE) methods and finite volume (FV) schemes. As the accuracy is achieved by means of high-order polynomial approximation within an element and the physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces.

DG methods are indeed a natural choice for solving hyperbolic equations, such as the compressible Euler equations. However, the DG formulation seems less certain and advantageous for diffusion equations such as the compressible Navier–Stokes equations, where viscous and heat fluxes exist and require the evaluation of the solution derivatives at the interfaces.

* Corresponding author.
E-mail addresses: chengjian@buaa.edu.cn (J. Cheng), yuehq@buaa.edu.cn (H. Yue), yushj@buaa.edu.cn (S. Yu), liutg@buaa.edu.cn (T. Liu).

https://doi.org/10.1016/j.jcp.2018.02.031
0021-9991/

In order to appropriately address this issue, a variety of different approaches [8–15] have been proposed and developed within the past decade and some of them, such as symmetric IP (SIP) method [12–14] and the second Bassi and Rebay (BR2) scheme [15], have become quite popular among the DG community.

It should be noted that based on the direct weak formulation of DG framework and the analysis of the solution structure of the scalar diffusion equation, a method named the diffusive generalized Riemann problem (dGRP) method was first introduced by Gassner et al. [16]. Later, a direct discontinuous Galerkin (DDG) method was proposed by Liu and Yan [17] to construct the numerical flux for diffusion problems based on DG framework. Compared to the original IP method, the DDG method can be viewed as a multi-term penalty method. The numerical flux defined by the DDG method is simple, compact, conservative, and consistent. Very recently, the DDG method has been successfully extended and applied for solving the both two dimensional and three dimensional compressible Navier–Stokes equations on arbitrary grids by Cheng et al. [24,25]. Owning to its simplicity in implementation, the DDG method shows its potential to deliver comparable accuracy as the widely used BR2 method at a significantly reduced computational cost, thus, becomes an attractive alternative to discretize the compressible Navier–Stokes equations on arbitrary grids. It should be noted that since the original DDG method was first introduced, a variety of new DDG type methods have been proposed and developed. In order to avoid that the jump terms of higher order ($k \geq 4$) derivatives in the original DDG discretization, a new DDG method with interface correction (DDG(IC)) is introduced and further analyzed by Liu et al. [18–20] for solving convection diffusion problems, and then extended to solve the compressible Navier–Stokes equations by Cheng et al. [27]. A maximum-principle-satisfying direct discontinuous Galerkin method for time dependent convection diffusion equations is further developed by Chen et al. [28]. Meanwhile, due to the reason that the formulation of the original DDG method lacks symmetric properties, a symmetric DDG method is developed and analyzed by Yan and his coworkers [21–23,29], then extended for solving the two dimensional compressible Navier–Stokes equations by Yue et al. [26]. The newly developed DDG type methods mentioned above possessed the merit and beauty of the original DDG method, meanwhile offer a tantalizing glimpse of further investigation and evaluation of their full potential for real applications.

Nearly over the same period of time, practical formulations for goal-oriented a posteriori error estimation and adaptation has been successfully developed based on the high-order DG framework [30–32,35–38] and a variety of other numerical methods [33,34]. The advantage of this output-based or adjoint-based error estimation approach is that it provides a well-founded technique for estimating the error targeted for specific functional outputs, such as lift and drag coefficients in computational aerodynamics, and then guides the adaptive refinement strategies for automatically producing accurate simulations at optimal cost with respective to the objective of interest. In the procedure of performing output-based adaptation, an adjoint solution is required for the error estimate and grid adaptation. There are two potential approaches to obtain the adjoint solutions, either through solving a continuous adjoint systems or through solving a discrete adjoint systems. Both of these two approaches above have their pros and cons. In the continuous adjoint approach [35], the adjoint equations have to be analytically derived base on the original governing equations combined with specific boundary conditions, then appropriate discretization needs to be carefully designed and applied to solve the derived adjoint systems in order to get the adjoint solutions. By directly solving the continuous adjoint equations, it can provide better output estimation if the primal and adjoint solutions are both well resolved and also make the preservation of consistency of the numerical discretization for both primal and adjoint equations more simple and straightforward. However, the derivation of linearization can be intricate as well as the imposition of the boundary conditions, which makes it less practical and popular compared to the discrete adjoint approach. For the implementation of discrete adjoint approach [30–32], the discretization of adjoint system is directly based on the linearization of the spatial discretization for the primal equations, no extra analytical derivation is needed and different types of boundary conditions can be automatically included and treated. However, extra efforts becomes necessary, in most of cases, to preserve the so-called adjoint consistency, as only adjoint consistent discretization can deliver the optimal rates of convergence with respect to the computed error of output functional. Plenty of efforts have been devoted to analyze and develop special treatments [34,37–39], which are tailored to the specific spatial discretization and treatment of boundary conditions.

In this paper, an adjoint-based high-order h-adaptive direct discontinuous Galerkin method is developed and further analyzed for the two dimensional compressible steady state Navier–Stokes equations. Particular emphasis is devoted to the analysis of the adjoint consistency for three different direct discontinuous Galerkin discretizations, i.e. the original direct discontinuous Galerkin method (DDG), the direct discontinuous Galerkin method with interface correction (DDG(IC)) and the symmetric direct discontinuous Galerkin method (SDDG). Theoretical analysis shows the extra interface correction term adopted in the DDG(IC) and SDDG methods plays a key role in preserving the adjoint consistency. To be specific, for the model problem considered in this work, we prove that the original DDG method is not adjoint consistent, while the DDG(IC) and SDDG methods can be adjoint consistent with appropriate treatment of boundary conditions and correct modification towards the underlying output functionals. The performance of those three DDG type methods is carefully investigated and evaluated through typical test cases. Based on the theoretical analysis, an adjoint-based h-adaptive DDG(IC) method is further developed and investigated. The adjoint-based error estimator is derived based on the DDG(IC) discretization and its effectivity is carefully verified, which shows its potential in the development and implementation of adjoint-based adaptation for simulating compressible flows.

The rest of the paper is organized as follows. The governing equations are briefly described in Section 2. The spatial discretizations of the DDG method, DDG(IC) method and SDDG method for the compressible Navier–Stokes equations are described in Section 3. In Section 4, adjoint consistent analysis based on the DDG type discretizations is presented and

discussed in detail, the special treatment in preserving adjoint consistency is developed through modified target functional approach. Based on the theoretical analysis, an adjoint-based error estimator for the development of adjoint-based h-adaptive DDG(IC) method is derived and presented. Numerical experiments are reported in Section 5. Concluding remarks are given in Section 6.

2. Governing equations

Let Ω be the computational domain in $\mathbb{R}^2$, the two dimensional steady state compressible Navier–Stokes equations are given by:

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

where $\boldsymbol{u} = (\rho, \rho u, \rho v, E)^{\mathrm{T}}$ is the solution vector consisting of the density ρ, velocity vector $\boldsymbol{v} = (u, v)^{\mathrm{T}}$, and the specific total energy E. The convective flux functions $\boldsymbol{\mathcal{F}}^c(\boldsymbol{u})$ are defined as

$$\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}) = \big(\boldsymbol{\mathcal{F}}_x(\boldsymbol{u}), \boldsymbol{\mathcal{F}}_y(\boldsymbol{u})\big) = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho u v \\ \rho u v & \rho v^2 + p \\ (E+p)u & (E+p)v \end{pmatrix}. \tag{2}$$

Here, p is pressure which is determined by the equation of state of an ideal gas, i.e.,

$$p = (\gamma - 1)\big(E - \frac{1}{2}\rho(u^2 + v^2)\big), \tag{3}$$

where $\gamma = 1.4$ is the heat capacity ratio for air.

The viscous flux functions $\boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u})$ are defined as

$$\begin{aligned} \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u}) &= \big(\boldsymbol{\mathcal{G}}_x(\boldsymbol{u}, \nabla \boldsymbol{u}), \boldsymbol{\mathcal{G}}_y(\boldsymbol{u}, \nabla \boldsymbol{u})\big) \\ &= \begin{pmatrix} 0 & 0 \\ \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \\ u\tau_{xx} + v\tau_{xy} + \mathcal{K}T_x & u\tau_{yx} + v\tau_{yy} + \mathcal{K}T_y \end{pmatrix}, \end{aligned} \tag{4}$$

where $\mathcal{K}$ is the thermal conductivity coefficient. The viscous stress tensor can be computed from

$$\boldsymbol{\tau} = \mu\big(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^{\mathrm{T}} - \frac{2}{3}(\nabla \cdot \boldsymbol{v})\mathrm{I}\big). \tag{5}$$

Finally, the temperature T is defined by

$$\mathcal{K}T = \frac{\mu\gamma}{Pr}\frac{E - \frac{1}{2}\rho \boldsymbol{v}^2}{\rho}. \tag{6}$$

In the above formulas, Pr is the Prandtl number, which is taken as 0.72 for sea-level air, and μ is the dynamic viscosity coefficient which can be computed by Sutherland's law as follows:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}}\frac{T_0 + S}{T + S}, \tag{7}$$

where μ_0 and T_0 denote the reference dynamic viscosity coefficient and reference temperature respectively, and S is assumed as a constant with a value of $110.5K$.

In order to clearly describe the spatial discretization of DDG type methods, we rewrite the viscous flux functions $\boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u})$ as follows:

$$\begin{aligned} \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u}) &= \big(\boldsymbol{\mathcal{G}}_x(\boldsymbol{u}, \nabla \boldsymbol{u}), \boldsymbol{\mathcal{G}}_y(\boldsymbol{u}, \nabla \boldsymbol{u})\big) \\ &= \big((\boldsymbol{\mathcal{G}}_{xx}\ \boldsymbol{\mathcal{G}}_{xy}) \cdot \nabla \boldsymbol{u}, (\boldsymbol{\mathcal{G}}_{yx}\ \boldsymbol{\mathcal{G}}_{yy}) \cdot \nabla \boldsymbol{u}\big), \end{aligned} \tag{8}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{G}}_{xx} &= \frac{\partial \boldsymbol{\mathcal{G}}_x}{\partial \boldsymbol{u}_x} \\ &= \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}u & \frac{4}{3} & 0 & 0 \\ -v & 0 & 1 & 0 \\ -(\frac{4}{3}u^2 + v^2 + \frac{\gamma}{Pr}(\frac{E}{\rho} - \boldsymbol{v}^2)) & (\frac{4}{3} - \frac{\gamma}{Pr})u & (1 - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{pmatrix} \end{aligned}$$

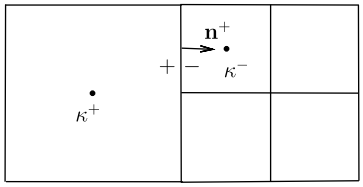


Fig. 1. Definition of interior and exterior trace.

$$\mathcal{G}_{xy} = \frac{\partial \mathcal{G}_x}{\partial \boldsymbol{u}_y} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v & 0 & -\frac{2}{3} & 0 \\ -u & 1 & 0 & 0 \\ -\frac{1}{3}uv & v & -\frac{2}{3}u & 0 \end{pmatrix} \qquad \mathcal{G}_{yx} = \frac{\partial \mathcal{G}_y}{\partial \boldsymbol{u}_x} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 \\ \frac{2}{3}u & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3}uv & -\frac{2}{3}v & u & 0 \end{pmatrix}$$

$$\mathcal{G}_{yy} = \frac{\partial \mathcal{G}_y}{\partial \boldsymbol{u}_y} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ -\frac{4}{3}v & 0 & \frac{4}{3} & 0 \\ -(u^2 + \frac{4}{3}v^2 + \frac{\gamma}{Pr}(\frac{E}{\rho} - \boldsymbol{v}^2)) & (1 - \frac{\gamma}{Pr})u & (\frac{4}{3} - \frac{\gamma}{Pr})v & \frac{\gamma}{Pr} \end{pmatrix}.$$

For the sake of simplification, we introduce the following definition:

$$\boldsymbol{G}(\boldsymbol{u})\nabla \boldsymbol{u} := \big((\mathcal{G}_{xx}\ \mathcal{G}_{xy}) \cdot \nabla \boldsymbol{u}, (\mathcal{G}_{yx}\ \mathcal{G}_{yy}) \cdot \nabla \boldsymbol{u}\big). \tag{9}$$

Finally, we obtain the equivalent form of the primal compressible Navier–Stokes equations given in Eq. (1) as follows:

$$\nabla \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{u}) - \nabla \cdot (\boldsymbol{G}(\boldsymbol{u})\nabla \boldsymbol{u}) = 0 \quad \text{in} \quad \Omega. \tag{10}$$

3. Direct DG discretizations for the compressible Navier–Stokes equations

3.1. Spatial discretization of direct DG methods

Let us begin this section with introducing some basic notations. The shape-regular subdivision of domain Ω is denoted by $\mathcal{T}_h = \{\kappa\}$. For $\kappa \in \mathcal{T}_h$, $\partial\kappa \cap \partial\Omega \neq \varnothing$ is the boundary face, we denote the set of all boundary faces by Γ. An interior edge of $\mathcal{T}_h$ is the one-dimensional interior of $\partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$, correspondingly, we denote the set of all internal faces by Γ_I. We denote the outward unit normal vector of $\partial\kappa$ by $\boldsymbol{n}_\kappa := (n_x, n_y)^{\mathrm{T}}$. We use $\boldsymbol{u}^+$ and $\boldsymbol{u}^-$ to denote the traces of $\boldsymbol{u}$ evaluated from the interior and the exterior of $\partial\kappa$, respectively, as shown in Fig. 1. Then, we define the average and jump of $\boldsymbol{u}$ on $\partial\kappa$ as follows:

$$\overline{\boldsymbol{u}} = \frac{1}{2}(\boldsymbol{u}^- + \boldsymbol{u}^+), \quad [\boldsymbol{u}] = \boldsymbol{u}^- - \boldsymbol{u}^+, \quad [\![\boldsymbol{u}]\!] = (\boldsymbol{u}^- - \boldsymbol{u}^+) \otimes \boldsymbol{n}^+, \tag{11}$$

where $\boldsymbol{n}^+$ is the outward unit normal vector of $\partial\kappa^+$.

For matrices $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^{m\times n}$, we define the matrix multiplication operator $\boldsymbol{a} : \boldsymbol{b} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij}$ and for vectors $\boldsymbol{c} \in \mathbb{R}^m$, $\boldsymbol{d} \in \mathbb{R}^n$ the dyadic tensor $\boldsymbol{c} \otimes \boldsymbol{d} \in \mathbb{R}^{m\times n}$ is defined by $(\boldsymbol{c} \otimes \boldsymbol{d})_{ij} = c_i d_j$. Additionally, we introduce the following broken Sobolev space

$$\mathcal{V}_{h,p} = \left\{ \boldsymbol{u} \in [L_2(\Omega)]^4 : \boldsymbol{u}|_\kappa \in [\mathcal{P}^p]^4 \quad \forall \kappa \in \mathcal{T}_h \right\}, \tag{12}$$

where $\mathcal{P}^p$ denotes the space of polynomials with degree not higher than p.

In order to obtain the DG discretization of Eq. (10), we multiply them by a test function $\boldsymbol{v}_h \in \mathcal{V}_{h,p}$ and integrate by parts over element k, then, we get the following weak formulation: find $\boldsymbol{u}_h \in \mathcal{V}_{h,p}$, such that

$$\begin{aligned} &-\int_\kappa \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega + \int_{\partial\kappa} (\boldsymbol{n}_\kappa \cdot \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h)) \cdot \boldsymbol{v}_h \mathrm{d}s \\ &+ \int_\kappa (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega - \int_{\partial\kappa} (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{u}_h) : \boldsymbol{n}_\kappa \otimes \boldsymbol{v}_h \mathrm{d}s \\ &= 0 \qquad \forall \boldsymbol{v}_h \in \mathcal{V}_{h,p}. \end{aligned} \tag{13}$$

Summing over all the elements $\kappa \in \mathcal{T}_h$ and approximating the inviscid flux and viscous flux functions at each cell interface and boundary face by suitable numerical flux functions, then the spacial discretization can be given in a face-based form as follows:

$$\begin{aligned}\mathcal{N}(\boldsymbol{u}_h, \boldsymbol{v}_h) \equiv & -\int_{\Omega} \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega + \sum_{k\in\mathcal{T}_h} \int_{\partial k\setminus\Gamma} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)\cdot \boldsymbol{v}_h^+ \mathrm{d}s \\ & + \int_{\Omega} (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega \\ & + \int_{\Gamma_I} \boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h : [\![\boldsymbol{v}_h]\!] \mathrm{d}s + \int_{\Gamma_I} \boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h : [\![\boldsymbol{u}_h]\!] \mathrm{d}s \\ & + \mathcal{N}_{\Gamma}(\boldsymbol{u}_h, \boldsymbol{v}_h) = 0 \qquad \forall \boldsymbol{v}_h \in \mathcal{V}_{h,p},\end{aligned} \tag{14}$$

where $\boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h$ and $\boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h$ are the viscous numerical fluxes defined for the DDG type methods, to be specific, can be given as follows

$$\begin{aligned}\boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h &= \{\boldsymbol{G}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{u}_h]\!] + \{\boldsymbol{G}(\boldsymbol{u}_h)\nabla\boldsymbol{u}_h\} + \{\boldsymbol{G}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{u}_h\cdot\boldsymbol{n}^+)], \\ \boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h &= \lambda\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{v}_h]\!] + \omega\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{v}_h\} + \lambda\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{v}_h\cdot\boldsymbol{n}^+)],\end{aligned} \tag{15}$$

and $\mathcal{N}_{\Gamma}(\boldsymbol{u}_h, \boldsymbol{v}_h)$ contains all the boundary terms. All the three DDG methods can be specified by the different choices of the parameters λ and ω, i.e. $\lambda = 0$, $\omega = 0$ represents the original DDG method, $\lambda = 0$, $\omega = 1$ represents the DDG(IC) method, and $\lambda = 1$, $\omega = 1$ represents the SDDG method.

The above spatial discretization can be also written in an element-based form, which can be given as follows:

$$\begin{aligned}\mathcal{N}(\boldsymbol{u}_h, \boldsymbol{v}_h) \equiv & -\int_{\Omega} \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega + \sum_{k\in\mathcal{T}_h} \int_{\partial k\setminus\Gamma} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)\cdot \boldsymbol{v}_h^+ \mathrm{d}s \\ & + \int_{\Omega} (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{u}_h) : \nabla \boldsymbol{v}_h \mathrm{d}\Omega \\ & - \sum_{k\in\mathcal{T}_h} \int_{\partial k\setminus\Gamma} \boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h : (\boldsymbol{v}_h^+ \otimes \boldsymbol{n}^+) \mathrm{d}s + \sum_{k\in\mathcal{T}_h} \int_{\partial k\setminus\Gamma} \boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h : [\![\boldsymbol{u}_h]\!] \mathrm{d}s \\ & + \mathcal{N}_{\Gamma}(\boldsymbol{u}_h, \boldsymbol{v}_h) = 0 \qquad \forall \boldsymbol{v}_h \in \mathcal{V}_{h,p},\end{aligned} \tag{16}$$

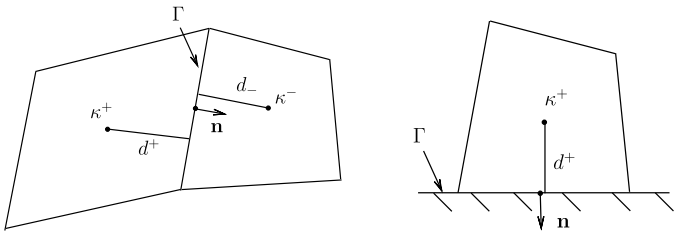
similarly, here $\boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h$ and $\boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h$ in Eq. (16) are the equivalent element-based version of viscous numerical fluxes, which can be given as follows:

$$\begin{aligned}\boldsymbol{G}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{u}_h &= \{\boldsymbol{G}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{u}_h]\!] + \{\boldsymbol{G}(\boldsymbol{u}_h)\nabla\boldsymbol{u}_h\} + \{\boldsymbol{G}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{u}_h\cdot\boldsymbol{n}^+)], \\ \boldsymbol{G}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)\nabla}\boldsymbol{v}_h &= \lambda\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}(-\boldsymbol{v}_h^+ \otimes \boldsymbol{n}^+) + \omega\frac{1}{2}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{v}_h^+ + \lambda\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\beta_1 h(-\nabla(\nabla\boldsymbol{v}_h^+\cdot\boldsymbol{n}^+)),\end{aligned} \tag{17}$$

where $\lambda = 0$, $\omega = 0$ represents the original DDG method, $\lambda = 0$, $\omega = 1$ represents the DDG(IC) method, and $\lambda = 1$, $\omega = 1$ represents the SDDG method.

The choice of the parameters (β_0, β_1) is based on both the theoretical analysis of the DDG methods for the convection diffusion equations, cf. [18–20] and our numerical experiments for solving practical problems of the compressible Navier–Stokes equations [24]. The numerical experiments indicate that choosing the β_0 as roughly the same scale of k^2 for P_k polynomials will be good enough to stabilize the scheme meanwhile without bringing any detrimental effects to the order of accuracy. In specific, for DDG and DDG(IC) methods, we choose β_0 equal to 4 for both DG(p_1) and DG(p_2) discretizations and for the second parameter β_1, we set it equal to 1/12, for the SDDG method, we note that the ratio of β_0/β_1 has to be sufficient large in order to be numerical stable. As for the definition of the characteristic length h, the general principle is that the characteristic length h should be associated with the target face where the viscous flux is evaluated and also should be orthogonal to the target face. A variety of different definitions had been tested in our previous work [24], and the one gives the best performance currently is given as follows:

$$h := \begin{cases} \min(d^+,\ d^-), & \text{at internal face,} \\ d^+, & \text{at boundary face,} \end{cases} \tag{18}$$



(a) For internal face (b) For boundary face

Fig. 2. Definition of characteristic length h in the DDG flux.

where d^+ and d^- are the distance from the cell center to the target face as shown in Fig. 2. We note that in some cases, especially on curved meshes, calculating the distance exactly might be troublesome, as the common face or the boundary face might be a curve either presented by high-order polynomial or spline. Thus, we borrow the idea from [14], which the characteristic size h can also be measured as follows

$$h := \frac{1}{2}\min(vol(k^+), vol(k^-))/len(\Gamma), \tag{19}$$

where $vol(k)$ denotes the volume of cell k and $len(\Gamma)$ is the length of the target face. This definition will simplify the calculation of the characteristic length under curved meshes, as the volume and face length can be easily obtained.

3.2. Boundary treatment

Note that the numerical flux functions defined in Eq. (17) involve the first and second derivatives of conservative variables at ghost states of the boundary faces. In general, these derivatives are not available for practical problems. Therefore on boundary faces, we drop the jump terms of the second order derivatives and only use the internal first order derivatives to define the viscous numerical flux functions.

To be specific, e.g. on no-slip wall boundary Γ_w, the boundary discretization in Eq. (14) can be written as follows:

$$\begin{aligned}\mathcal{N}_{\Gamma_w}(\boldsymbol{u}_h, \boldsymbol{v}_h) = &\int_{\Gamma_w} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_{\Gamma_w}^-(\boldsymbol{u}_h^+), \boldsymbol{n}^+)\cdot \boldsymbol{v}_h^+ \mathrm{d}s - \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}\widehat{(\boldsymbol{u}_h)}\nabla \boldsymbol{u}_h : \boldsymbol{v}_h^+ \otimes \boldsymbol{n}^+ \mathrm{d}s \\ &+ \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)}\nabla \boldsymbol{v}_h : (\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+)\otimes \boldsymbol{n}^+ \mathrm{d}s\end{aligned} \tag{20}$$

where the ghost state value $\boldsymbol{u}_{\Gamma_w}^-(\boldsymbol{u}_h^+)$ and the boundary viscous flux can be given as

$$\begin{aligned}\boldsymbol{G}_{\Gamma_w}\widehat{(\boldsymbol{u}_h)}\nabla \boldsymbol{u}_h &= \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+)\otimes \boldsymbol{n}^+) + \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla \boldsymbol{u}_h^+, \\ \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}\widehat{(\boldsymbol{u}_h)}\nabla \boldsymbol{v}_h &= \lambda \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+)\frac{\beta_0}{h}(-\boldsymbol{v}_h^+\otimes \boldsymbol{n}^+) + \omega \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla \boldsymbol{v}_h^+,\end{aligned} \tag{21}$$

and $\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)$ is the consistent treatment of boundary function based on the specific type of boundary conditions, e.g. along no-slip adiabatic wall boundaries the no-slip boundary condition enforces $\boldsymbol{v} = (u, v) = \boldsymbol{0}$ which leads to

$$\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) = (\rho^+, 0, 0, E^+)^{\mathrm{T}}. \tag{22}$$

Furthermore, $\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)$ is the corresponding homogeneity tensor which is modified based on specific type of boundary conditions imposed on boundary face Γ_w. To be specific, one consistent treatment of the corresponding modified homogeneity tensor for the no-slip adiabatic wall boundary, which is currently adopted in this work, can be given as follows:

$$\begin{aligned}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla \boldsymbol{u}_h^+ &= \left(\mathcal{G}_{\Gamma x}(\boldsymbol{u}_h^+, \nabla \boldsymbol{u}_h^+), \mathcal{G}_{\Gamma y}(\boldsymbol{u}_h^+, \nabla \boldsymbol{u}_h^+)\right) \\ &= \left((\mathcal{G}_{\Gamma xx}\ \mathcal{G}_{\Gamma xy})\cdot \nabla \boldsymbol{u}_h^+, (\mathcal{G}_{\Gamma yx}\ \mathcal{G}_{\Gamma yy})\cdot \nabla \boldsymbol{u}_h^+\right),\end{aligned} \tag{23}$$

where

$$\mathcal{G}_{\Gamma xx} = \frac{\mu^+}{\rho^+}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}u^+ & \frac{4}{3} & 0 & 0 \\ -v^+ & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathcal{G}_{\Gamma xy} = \frac{\mu^+}{\rho^+}\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v^+ & 0 & -\frac{2}{3} & 0 \\ -u^+ & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{G}_{\Gamma yx} = \frac{\mu^+}{\rho^+}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v^+ & 0 & 1 & 0 \\ \frac{2}{3}u^+ & -\frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathcal{G}_{\Gamma yy} = \frac{\mu^+}{\rho^+}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -u^+ & 1 & 0 & 0 \\ -\frac{4}{3}v^+ & 0 & \frac{4}{3} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

It can be easily verified that the modified homogeneity tensor guarantees that the no-slip adiabatic wall boundary condition is imposed along wall boundary Γ_w as the fourth component of the viscous flux vanishes for the no-slip adiabatic wall boundary. Finally, the farfield boundary Γ_f is treated through characteristic boundary conditions. More details about the treatments for the other types of wall boundary conditions, we refer the readers to [12,14] and references cited therein.

3.3. Pseudo-transient continuation

For temporal discretization, we add a pseudo time term to Eq. (10) and rewrite the steady state compressible Navier–Stokes equations as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}) - \nabla \cdot (\boldsymbol{G}(\boldsymbol{u})\nabla \boldsymbol{u}) = 0 \quad \text{in} \quad \Omega. \tag{24}$$

Spatial discretization of Eq. (24) leads to a system of ordinary differential equations:

$$\mathcal{M}\frac{d\boldsymbol{u}_h}{dt} = \mathcal{R}_h(\boldsymbol{u}_h), \tag{25}$$

where $\mathcal{M}$ is mass matrix and $\mathcal{R}_h$ is right hand side residual. In this work, as we are only interested in stationary problems, thus, the implicit backward Euler scheme is applied. After temporal discretization and linearizing, we get a linear system written as

$$\left(\frac{\mathcal{M}}{\Delta t^n}\mathrm{I} - \frac{\partial \mathcal{R}_h}{\partial \boldsymbol{u}_h}\right)\Delta \boldsymbol{u}_h^n = \mathcal{R}_h(\boldsymbol{u}_h^n), \tag{26}$$

where Δt^n is time increment, and $\Delta \boldsymbol{u}_h^n = \boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n$. Then, the linear system is solved by GMRES method with ILU0 preconditioning in this work.

4. Adjoint consistent analysis of DDG type discretizations

In this section, we will present the analysis of adjoint consistency for the spatial discretizations of DDG method, DDG(IC) method and SDDG method. In order to complete the adjoint consistency analysis, first we give a brief description of the model problem and the derivation of the continuous adjoint compressible Navier–Stokes equations. Then a proof of adjoint consistency for those three DDG methods based on discrete adjoint approach will be provided and discussed in detail. In the end, we give the error representation formula for the implementation of adjoint-based h-adaptation based on the DDG(IC) method.

4.1. The model problem

We consider a model problem, i.e., the laminar flow over an airfoil with typical no-slip wall boundary Γ_w and the far-field boundary Γ_f.

The governing equations are the steady state compressible Navier–Stokes equations in Eq. (10) with the following output functional defined along the no-slip adiabatic wall boundary Γ_w:

$$J(\boldsymbol{u}) = \int_{\Gamma_w} (p\boldsymbol{n} - \boldsymbol{\tau}\boldsymbol{n}) \cdot \boldsymbol{\psi} \mathrm{d}s \quad \text{on} \quad \Gamma_w, \tag{27}$$

where p is pressure, $\boldsymbol{\tau}$ is viscous stress tensor, $\boldsymbol{n}$ is unit normal vector, $\boldsymbol{\psi} = (\psi_1, \psi_2)^{\mathrm{T}} := \frac{1}{C_\infty}(\cos(\alpha), \sin(\alpha))^{\mathrm{T}}$ and $\boldsymbol{\psi} = (\psi_1, \psi_2)^{\mathrm{T}} := \frac{1}{C_\infty}(-\sin(\alpha), \cos(\alpha))^{\mathrm{T}}$ for drag coefficient C_d and lift coefficient C_l, respectively, and C_∞ is equal to $\frac{1}{2}\rho_\infty \boldsymbol{v}_\infty^2 l_{ref}$. ρ_∞, $\boldsymbol{v}_\infty$ and l_∞ are the freestream density, freestream velocity vector and the reference length, respectively.

4.2. The continuous adjoint formulation

We give a brief description of the derivation for the continuous adjoint Navier–Stokes equations, the derivation of continuous adjoint Navier–Stokes equations can be obtained first by multiplying the left hand side of Eq. (10) with $\boldsymbol{z} \in \mathcal{V}$, then integrating by parts

$$\begin{aligned}(\nabla \cdot (\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}) - \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u})), \boldsymbol{z})_\Omega = \\ -(\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}) - \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u}), \nabla \boldsymbol{z})_\Omega + (\boldsymbol{n} \cdot (\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}) - \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla \boldsymbol{u})), \boldsymbol{z})_{\partial\Omega},\end{aligned} \tag{28}$$

where $\mathcal{V}$ is some suitable chosen functional space such that $\mathcal{V}_{h,p} \subset \mathcal{V}$. Then we linearize the above formula about $\boldsymbol{u}$ in the direction $\boldsymbol{w}$, which leads to:

$$\begin{aligned}&(\nabla \cdot (\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}}\boldsymbol{w} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}}\boldsymbol{w} - \boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}}\nabla\boldsymbol{w}), \boldsymbol{z})_\Omega = \\ &\quad -((\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}})\boldsymbol{w} - \boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}}\nabla\boldsymbol{w}, \nabla\boldsymbol{z})_\Omega + (\boldsymbol{n}\cdot((\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}})\boldsymbol{w} - \boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}}\nabla\boldsymbol{w}), \boldsymbol{z})_{\partial\Omega},\end{aligned} \tag{29}$$

where $\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} := \partial_{\boldsymbol{u}}\boldsymbol{\mathcal{F}}^c(\boldsymbol{u})$ is the Fréchet derivative of $\boldsymbol{\mathcal{F}}^c$ with respect to $\boldsymbol{u}$, similarly, $\boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}} := \partial_{\boldsymbol{u}}\boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla\boldsymbol{u}) = \boldsymbol{G}'[\boldsymbol{u}]\nabla\boldsymbol{u}$ and $\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}} := \partial_{\nabla\boldsymbol{u}}\boldsymbol{\mathcal{F}}^v(\boldsymbol{u}, \nabla\boldsymbol{u}) = \boldsymbol{G}(\boldsymbol{u})$ are the Fréchet derivatives of $\boldsymbol{\mathcal{F}}^v$ with respect to $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$, respectively. Use integration by parts again, we can obtain the following variational formulation for the continuous adjoint problem considered in this work: find $\boldsymbol{z} \in \mathcal{V}$ such that

$$\begin{aligned}&-(\boldsymbol{w}, (\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z})_\Omega - (\boldsymbol{w}, \nabla\cdot((\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z}))_\Omega + (\boldsymbol{w}, (\boldsymbol{n}\cdot(\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z}))_{\partial\Omega} \\ &\quad + (\boldsymbol{w}, (\boldsymbol{n}\cdot(\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}}))^{\mathrm{T}}\boldsymbol{z})_{\partial\Omega} - (\nabla\boldsymbol{w}, (\boldsymbol{n}\cdot\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\boldsymbol{z})_{\partial\Omega} = J'[\boldsymbol{u}](\boldsymbol{w}) \quad \forall\boldsymbol{w} \in \mathcal{V},\end{aligned} \tag{30}$$

where $J'[\boldsymbol{u}](\boldsymbol{w})$ is the linearization of the output functional with respect to $\boldsymbol{u}$, to be specific, it can be given as follows

$$\begin{aligned}J'[\boldsymbol{u}](\boldsymbol{w}) &= \frac{1}{C_\infty}\int_\Gamma (p_{\boldsymbol{u}}\boldsymbol{n} - \boldsymbol{\tau}_{\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi}\,\boldsymbol{w} - (\boldsymbol{\tau}_{\nabla\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi}\nabla\boldsymbol{w}\,\mathrm{d}s \\ &= (\boldsymbol{w}, \frac{1}{C_\infty}(p_{\boldsymbol{u}}\boldsymbol{n} - \boldsymbol{\tau}_{\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi})_{\Gamma_w} - (\nabla\boldsymbol{w}, \frac{1}{C_\infty}(\boldsymbol{\tau}_{\nabla\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi})_{\Gamma_w},\end{aligned} \tag{31}$$

where $p_{\boldsymbol{u}}$, $\boldsymbol{\tau}_{\boldsymbol{u}}$ and $\boldsymbol{\tau}_{\nabla\boldsymbol{u}}$ are the Fréchet derivatives of p and $\boldsymbol{\tau}$ with respect to $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$, respectively.

From comparing the corresponding cell integral terms and face integral terms on both sides of Eq. (30), the continuous adjoint equations can be obtained and given as the following equations

$$-(\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z} - \nabla\cdot((\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z}) = 0 \quad \text{on} \quad \Omega, \tag{32}$$

subject to the boundary conditions on Γ_w,

$$\begin{aligned}(\boldsymbol{n}\cdot(\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}}))^{\mathrm{T}}\boldsymbol{z} + \boldsymbol{n}\cdot((\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z}) &= \frac{1}{C_\infty}(p_{\boldsymbol{u}}\boldsymbol{n} - \boldsymbol{\tau}_{\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi}, \\ (\boldsymbol{n}\cdot\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\boldsymbol{z} &= \frac{1}{C_\infty}(\boldsymbol{\tau}_{\nabla\boldsymbol{u}}\boldsymbol{n})\cdot\boldsymbol{\psi},\end{aligned} \tag{33}$$

and the boundary conditions on Γ_f,

$$\begin{aligned}(\boldsymbol{w}, (\boldsymbol{n}\cdot(\boldsymbol{\mathcal{F}}^c_{\boldsymbol{u}} - \boldsymbol{\mathcal{F}}^v_{\boldsymbol{u}}))^{\mathrm{T}}\boldsymbol{z} + \boldsymbol{n}\cdot((\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\nabla\boldsymbol{z})) &= 0, \\ -(\nabla\boldsymbol{w}, (\boldsymbol{n}\cdot\boldsymbol{\mathcal{F}}^v_{\nabla\boldsymbol{u}})^{\mathrm{T}}\boldsymbol{z}) &= 0 \quad \forall\boldsymbol{w} \in \mathcal{V}.\end{aligned} \tag{34}$$

Furthermore, assume the no-slip adiabatic wall boundary is imposed along Γ_w, which indicates that $\boldsymbol{v} = (u, v)$ equals to zero and $\boldsymbol{n}\cdot\nabla T = 0$. Substitute the no-slip adiabatic wall boundary conditions into the second equation of Eq. (33), the adjoint solution at boundary Γ_w for the model problem satisfies the following boundary conditions:

$$\boldsymbol{z}_{\Gamma_w} = (0, \frac{1}{C_\infty}\psi_1, \frac{1}{C_\infty}\psi_2, z_4)^{\mathrm{T}}, \qquad \boldsymbol{n}\cdot\nabla z_4 = 0. \tag{35}$$

For more details in the derivation of the continuous adjoint Navier–Stokes equations with other types of boundary conditions, one can refer to the work [38].

4.3. Analysis of adjoint consistency for DDG type discretizations

The discrete adjoint problem is defined by: find $\boldsymbol{z}_h \in \mathcal{V}_{h,p}$ such that

$$\mathcal{N}'[\boldsymbol{u}_h](\boldsymbol{w}, \boldsymbol{z}_h) = J'[\boldsymbol{u}_h](\boldsymbol{w}) \qquad \forall\boldsymbol{w} \in \mathcal{V}_{h,p}, \tag{36}$$

where $\mathcal{N}'[\boldsymbol{u}_h](\boldsymbol{w}, \boldsymbol{z}_h)$ and $J'[\boldsymbol{u}_h](\boldsymbol{w})$ are the Fréchet derivatives of $\mathcal{N}(\boldsymbol{u}_h, \boldsymbol{z}_h)$ and $J(\boldsymbol{u}_h)$ about $\boldsymbol{u}_h$ in the direction $\boldsymbol{w}$, respectively. With the definition of the discrete adjoint equations above, we say the discretization of the compressible Navier–Stokes equations is adjoint consistent for the model problem considered in this work if the exact primal solution $\boldsymbol{u}$ of Eq. (10) and the exact adjoint solution $\boldsymbol{z}$ of Eq. (32) satisfy the above discrete adjoint equations.

Based on the concept of adjoint consistency, we provide the following theorem about the adjoint consistency for the three specific DDG discretizations described in Eq. (14) and Eq. (15).

Theorem 1. *Consider the model problem described in Section 4.1 and the discrete adjoint problem given in Eq. (36), the DDG discretization is not adjoint consistent, while the DDG(IC) discretization ($\lambda = 0$) and the SDDG discretization ($\lambda = 1$) can be adjoint consistent with the modified output functional given as follows:*

$$\begin{aligned}\tilde{J}(\boldsymbol{u}_h) = &\int_{\Gamma_w} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_{\Gamma_w}^-(\boldsymbol{u}_h^+), \boldsymbol{n}) \cdot \hat{\boldsymbol{z}}_{\Gamma_w} \mathrm{d}s - \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+) \nabla \boldsymbol{u}_h : \hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n} \mathrm{d}s \\ &- \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+) \frac{\beta_0}{h} (\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+) : \hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n} \mathrm{d}s \\ &+ \lambda \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+) \frac{\beta_0}{h} (-\hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n}) : (\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+) \otimes \boldsymbol{n} \mathrm{d}s,\end{aligned} \tag{37}$$

where $\hat{\boldsymbol{z}_{\Gamma_w}} = \frac{1}{C_\infty}(0, \psi_1, \psi_2, 0)^{\mathrm{T}}$.

The framework for the analysis and proof of the adjoint consistency commonly contains two steps, which is introduced by Hartmann for the analysis of SIP method [37]. The first step is to analyze the consistency of the spatial discretization for the primal equations, i.e. the compressible Navier–Stokes equations. For the second step, firstly, we need to derive the discrete adjoint equations defined in Eq. (36) through linearizing both the spatial discretization of primal equations and the modified output functional. Then, the rest of the proof becomes to substitute the exact primal solutions $\boldsymbol{u}$ and the exact adjoint solution $\boldsymbol{z}$ into the derived discrete adjoint equations to verity whether the left hand side and the right hand side are equal with each other. Since the proof of consistency of the spatial discretization for primal equations based on the DDG type methods can be trivial, thus, we omit the first step, and only present the analysis and proof for the second step in details in the rest of this section.

In order to give a clear view of the proof for the second step, we rewrite the discrete adjoint equations given in Eq. (36) into an equivalent formulation, which is named as the adjoint residual formulation, and it can be given as follows: find $\boldsymbol{z}_h \in \mathcal{V}_{h,p}$ such that

$$\begin{aligned}&\int_\Omega \boldsymbol{w} \cdot \boldsymbol{R}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) \mathrm{d}\Omega + \sum_{k \in \mathcal{T}_h} \int_{\partial k \setminus \Gamma} \boldsymbol{w} \cdot \boldsymbol{r}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) + \nabla \boldsymbol{w} : \boldsymbol{\rho}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) + \nabla(\nabla \boldsymbol{w} \cdot \boldsymbol{n}) : \boldsymbol{\sigma}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) \mathrm{d}s \\ &\qquad + \int_\Gamma \boldsymbol{w} \cdot \boldsymbol{r}_\Gamma^*[\boldsymbol{u}_h](\boldsymbol{z}_h) + \nabla \boldsymbol{w} : \boldsymbol{\rho}_\Gamma^*[\boldsymbol{u}_h](\boldsymbol{z}_h) \mathrm{d}s = 0 \qquad \forall \boldsymbol{w} \in \mathcal{V}_{h,p},\end{aligned} \tag{38}$$

where $\boldsymbol{R}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ is the cell residual term, $\boldsymbol{r}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$, $\boldsymbol{\rho}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ and $\boldsymbol{\sigma}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ are the internal face residual terms, $\boldsymbol{r}_\Gamma^*[\boldsymbol{u}](\boldsymbol{z})$ and $\boldsymbol{\rho}_\Gamma^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ are the boundary face residual terms. Then, in order to complete the second step, we need to show that the exact solution $\boldsymbol{u}$ from the primal equation (10) and the exact solution $\boldsymbol{z}$ from the adjoint solution Eq. (32) satisfy

$$\begin{aligned}&\int_\Omega \boldsymbol{w} \cdot \boldsymbol{R}^*[\boldsymbol{u}](\boldsymbol{z}) \mathrm{d}\Omega + \sum_{k \in \mathcal{T}_h} \int_{\partial k \setminus \Gamma} \boldsymbol{w} \cdot \boldsymbol{r}^*[\boldsymbol{u}](\boldsymbol{z}) + \nabla \boldsymbol{w} : \boldsymbol{\rho}^*[\boldsymbol{u}](\boldsymbol{z}) + \nabla(\nabla \boldsymbol{w} \cdot \boldsymbol{n}) : \boldsymbol{\sigma}^*[\boldsymbol{u}](\boldsymbol{z}) \mathrm{d}s \\ &\qquad + \int_\Gamma \boldsymbol{w} \cdot \boldsymbol{r}_\Gamma^*[\boldsymbol{u}](\boldsymbol{z}) + \nabla \boldsymbol{w} : \boldsymbol{\rho}_\Gamma^*[\boldsymbol{u}](\boldsymbol{z}) \mathrm{d}s = 0 \qquad \forall \boldsymbol{w} \in \mathcal{V}_{h,p}.\end{aligned} \tag{39}$$

Now, it is clear that for the rest of the proof, we need to first derive the specific formulations of all the adjoint residual terms and then verify whether the above equality is satisfied or not with the exact primal solution $\boldsymbol{u}$ and the exact adjoint solution $\boldsymbol{z}$, which is given by the following two lemmas.

Lemma 1. *The cell residual term and internal face residual terms defined in the adjoint residual formulation Eq. (38) can be given as*

$$\boldsymbol{R}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) = (\mathcal{F}_{\boldsymbol{u}}^c[\boldsymbol{u}_h] - \boldsymbol{G}'[\boldsymbol{u}_h]\nabla \boldsymbol{u}_h)^{\mathrm{T}} \nabla \boldsymbol{z}_h + \nabla \cdot (\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h) \nabla \boldsymbol{z}_h) \qquad on \quad k \in \mathcal{T}_h, \tag{40}$$

and

$$
\begin{aligned}
\boldsymbol{r}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) &= (\mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}^+,\boldsymbol{u}^-,\boldsymbol{n}^+))^{\mathrm{T}}[\boldsymbol{z}_h] - \frac{1}{2}\frac{\beta_0}{h}(\boldsymbol{G}'[\boldsymbol{u}_h][\![\boldsymbol{u}_h]\!])^{\mathrm{T}}[\![\boldsymbol{z}_h]\!] \\
&\quad + \frac{\beta_0}{h}\{\boldsymbol{G}(\boldsymbol{u}_h)\}^{\mathrm{T}}[\![\boldsymbol{z}_h]\!]\cdot\boldsymbol{n} - \frac{1}{2}(\boldsymbol{G}'[\boldsymbol{u}_h]\nabla\boldsymbol{u}_h)^{\mathrm{T}}[\![\boldsymbol{z}_h]\!] \\
&\quad - \frac{1}{2}\beta_1 h(\boldsymbol{G}'[\boldsymbol{u}_h][\nabla(\nabla\boldsymbol{u}_h\cdot\boldsymbol{n})])^{\mathrm{T}}[\![\boldsymbol{z}_h]\!] \\
&\quad - \frac{1}{2}\lambda\frac{\beta_0}{h}((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h][\![\boldsymbol{z}_h]\!])^{\mathrm{T}}[\![\boldsymbol{u}_h]\!] + \lambda\frac{\beta_0}{h}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}[\![\boldsymbol{z}_h]\!]\cdot\boldsymbol{n} \\
&\quad - \frac{1}{2}\omega(\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\nabla\boldsymbol{z}_h)^{\mathrm{T}}[\![\boldsymbol{u}_h]\!] + \frac{1}{2}\omega[\![\boldsymbol{G}^{\mathrm{T}}[\boldsymbol{u}_h]\nabla\boldsymbol{z}_h]\!] - (1-\omega)\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h\cdot\boldsymbol{n} \\
&\quad - \frac{1}{2}\lambda\beta_1 h((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h][\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})])^{\mathrm{T}}[\![\boldsymbol{u}_h]\!] \\
&\quad + \lambda\beta_1 h\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}[\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})]\cdot\boldsymbol{n} \\
\boldsymbol{\rho}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) &= -\frac{1}{2}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)[\![\boldsymbol{z}_h]\!], \\
\boldsymbol{\sigma}^*[\boldsymbol{u}_h](\boldsymbol{z}_h) &= \beta_1 h\{\boldsymbol{G}(\boldsymbol{u}_h)\}^{\mathrm{T}}[\![\boldsymbol{z}_h]\!] \qquad on \quad \partial k\backslash\Gamma,\ k\in\mathcal{T}_h,
\end{aligned}
\tag{41}
$$

based on the spatial discretizations of the DDG method ($\omega=0,\lambda=0$), the DDG(IC) method ($\omega=1,\lambda=0$), and the SDDG method ($\omega=1,\lambda=1$), respectively.

Proof. The proof of this lemma can be achieved by the derivation of the left hand side of the discrete adjoint equations, as for the model problem considered in this work, the linearization of the output functional only involves boundary face values, thus, does not contribute to the cell residual term or the internal face residual terms.

The linearization of Eq. (14) about $\boldsymbol{u}_h$ in the direction of $\boldsymbol{w}$ for the DDG type spatial discretizations can be given as follows:

$$
\begin{aligned}
&\mathcal{N}'[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h) = \\
&-\int_\Omega(\mathcal{F}_u^c[\boldsymbol{u}_h]\boldsymbol{w}):\nabla\boldsymbol{z}_h\mathrm{d}\Omega - \sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}\mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}_h^+,\boldsymbol{u}_h^-,\boldsymbol{n}^+)\boldsymbol{w}[\boldsymbol{z}_h]\mathrm{d}s \\
&+\int_\Omega(\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\nabla\boldsymbol{u}_h):\nabla\boldsymbol{z}_h\mathrm{d}\Omega + \int_\Omega(\boldsymbol{G}(\boldsymbol{u}_h)\nabla\boldsymbol{w}):\nabla\boldsymbol{z}_h\mathrm{d}\Omega \\
&+\int_{\Gamma_I}\{\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\}\frac{\beta_0}{h}[\![\boldsymbol{u}_h]\!]:[\![\boldsymbol{z}_h]\!]\mathrm{d}s + \int_{\Gamma_I}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{w}]\!]:[\![\boldsymbol{z}_h]\!]\mathrm{d}s \\
&+\int_{\Gamma_I}\{\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\nabla\boldsymbol{u}_h\}:[\![\boldsymbol{z}_h]\!]\mathrm{d}s + \int_{\Gamma_I}\{\boldsymbol{G}(\boldsymbol{u}_h)\nabla\boldsymbol{w}\}:[\![\boldsymbol{z}_h]\!]\mathrm{d}s \\
&+\int_{\Gamma_I}\{\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\}\beta_1 h[\nabla(\nabla\boldsymbol{u}_h\cdot\boldsymbol{n})]:[\![\boldsymbol{z}_h]\!]\mathrm{d}s + \int_{\Gamma_I}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{w}\cdot\boldsymbol{n})]:[\![\boldsymbol{z}_h]\!]\mathrm{d}s \\
&+\lambda\int_{\Gamma_I}\{(\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}\}\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!]:[\![\boldsymbol{u}_h]\!]\mathrm{d}s + \lambda\int_{\Gamma_I}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!]:[\![\boldsymbol{w}]\!]\mathrm{d}s \\
&+\omega\int_{\Gamma_I}\{(\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}\nabla\boldsymbol{z}_h\}:[\![\boldsymbol{u}_h]\!]\mathrm{d}s + \omega\int_{\Gamma_I}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h\}:[\![\boldsymbol{w}]\!]\mathrm{d}s \\
&+\lambda\int_{\Gamma_I}\{(\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}\}\beta_1 h[\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})]:[\![\boldsymbol{u}_h]\!]\mathrm{d}s + \lambda\int_{\Gamma_I}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})]:[\![\boldsymbol{w}]\!]\mathrm{d}s \\
&+\mathcal{N}'_\Gamma[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h).
\end{aligned}
\tag{42}
$$

It should be noted that the linearization of the inviscid numerical flux term involves the linearization with respect to both its first argument $\boldsymbol{u}^+$ and its second argument $\boldsymbol{u}^-$, denoted as e.g. $\mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}^+,\boldsymbol{u}^-,\boldsymbol{n}^+)$ and $\mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}^+,\boldsymbol{u}^-,\boldsymbol{n}^+)$, respectively. In the above derivation, we use the following identity introduced in [37]:

$$\begin{aligned}\int_{\Gamma_I} \mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)\boldsymbol{w}^- \boldsymbol{z}_h^+ \mathrm{d}s &= -\int_{\Gamma_I} \mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}_h^-, \boldsymbol{u}_h^+, \boldsymbol{n}^-)\boldsymbol{w}^- \boldsymbol{z}_h^+ \mathrm{d}s \\ &= -\int_{\Gamma_I} \mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)\boldsymbol{w}^+ \boldsymbol{z}_h^- \mathrm{d}s.\end{aligned} \tag{43}$$

Rewrite the face-based terms of Eq. (42) into an equivalent element-based formulation, which leads to

$$\begin{aligned}&\mathcal{N}'[\boldsymbol{u}_h](\boldsymbol{w}, \boldsymbol{v}_h) = \\ &-\int_\Omega (\mathcal{F}_u^c[\boldsymbol{u}_h]\boldsymbol{w}) : \nabla \boldsymbol{z}_h \mathrm{d}\Omega - \sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} \mathcal{H}'_{u^+}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)\boldsymbol{w}[\boldsymbol{z}_h]\mathrm{d}s \\ &+\int_\Omega (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\nabla \boldsymbol{u}_h) : \nabla \boldsymbol{z}_h \mathrm{d}\Omega + \int_\Omega (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{w}) : \nabla \boldsymbol{z}_h \mathrm{d}\Omega \\ &+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w})\frac{\beta_0}{h}[\![\boldsymbol{u}_h]\!] : [\![\boldsymbol{z}_h]\!]\mathrm{d}s - \sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}(\boldsymbol{w}\otimes\boldsymbol{n}) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s \\ &+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\nabla \boldsymbol{u}_h) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s + \frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{w}) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s \\ &+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w})\beta_1 h[\nabla(\nabla \boldsymbol{u}_h\cdot\boldsymbol{n})] : [\![\boldsymbol{z}_h]\!]\mathrm{d}s \\ &-\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\beta_1 h(\nabla(\nabla \boldsymbol{w}\cdot\boldsymbol{n})) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s \\ &+\frac{1}{2}\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} ((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w})\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!] : [\![\boldsymbol{u}_h]\!]\mathrm{d}s - \lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!] : (\boldsymbol{w}\otimes\boldsymbol{n})\mathrm{d}s \\ &+\frac{1}{2}w\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} ((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}\nabla \boldsymbol{z}_h) : [\![\boldsymbol{u}_h]\!]\mathrm{d}s - w\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla \boldsymbol{z}_h\} : (\boldsymbol{w}\otimes\boldsymbol{n})\mathrm{d}s \\ &+\frac{1}{2}\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma} ((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w})\beta_1 h[\nabla(\nabla \boldsymbol{z}_h\cdot\boldsymbol{n})] : [\![\boldsymbol{u}_h]\!]\mathrm{d}s \\ &-\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla \boldsymbol{z}_h\cdot\boldsymbol{n})] : (\boldsymbol{w}\otimes\boldsymbol{n})\mathrm{d}s \\ &+\mathcal{N}'_\Gamma[\boldsymbol{u}_h](\boldsymbol{u}_h, \boldsymbol{z}_h).\end{aligned} \tag{44}$$

Then we apply integral by parts for Eq. (44), it can be rewritten as follows:

$$\begin{aligned}&\mathcal{N}'[\boldsymbol{u}_h](\boldsymbol{w}, \boldsymbol{z}_h) = \\ &-\int_\Omega \boldsymbol{w}(\mathcal{F}_u^c[\boldsymbol{u}_h])^{\mathrm{T}}\nabla \boldsymbol{z}_h \mathrm{d}\Omega - \sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma} \boldsymbol{w}(\mathcal{H}'_{\boldsymbol{u}^+}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+))^{\mathrm{T}}[\boldsymbol{z}_h]\mathrm{d}s \\ &+\int_\Omega \boldsymbol{w}(\boldsymbol{G}'[\boldsymbol{u}_h]\nabla \boldsymbol{u}_h)^{\mathrm{T}}\nabla \boldsymbol{z}_h \mathrm{d}\Omega - \int_\Omega \boldsymbol{w}\nabla\cdot(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla \boldsymbol{z}_h)\mathrm{d}\Omega \\ &+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma} (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w})\frac{\beta_0}{h}[\![\boldsymbol{u}_h]\!] : [\![\boldsymbol{z}_h]\!]\mathrm{d}s - \sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}(\boldsymbol{w}\otimes\boldsymbol{n}) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s \\ &+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma} (\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w}\nabla \boldsymbol{u}_h) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s + \frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma} (\boldsymbol{G}(\boldsymbol{u}_h)\nabla \boldsymbol{w}) : [\![\boldsymbol{z}_h]\!]\mathrm{d}s\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{2}\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}(\boldsymbol{G}'[\boldsymbol{u}_h]\boldsymbol{w})\beta_1 h[\nabla(\nabla\boldsymbol{u}_h\cdot\boldsymbol{n})]:[\![\boldsymbol{z}_h]\!]\mathrm{d}s\\
&-\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}\{\boldsymbol{G}(\boldsymbol{u}_h)\}\beta_1 h(\nabla(\nabla\boldsymbol{w}\cdot\boldsymbol{n})):[\![\boldsymbol{z}_h]\!]\mathrm{d}s\\
&+\frac{1}{2}\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w})\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!]:[\![\boldsymbol{u}_h]\!]\mathrm{d}s-\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\frac{\beta_0}{h}[\![\boldsymbol{z}_h]\!]:(\boldsymbol{w}\otimes\boldsymbol{n})\mathrm{d}s\\
&+\frac{1}{2}\omega\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}\nabla\boldsymbol{z}_h):[\![\boldsymbol{u}_h]\!]\mathrm{d}s-\frac{1}{2}\omega\sum_{k\in\mathcal{T}_h}\int_{\partial k\backslash\Gamma}\boldsymbol{w}\cdot[\![\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h]\!]\mathrm{d}s\\
&+(1-\omega)\int_{\partial k\backslash\Gamma}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h)\mathrm{d}s+\int_{\Gamma}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h)\mathrm{d}s\\
&+\frac{1}{2}\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}((\boldsymbol{G}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w})\beta_1 h[\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})]:[\![\boldsymbol{u}_h]\!]\mathrm{d}s\\
&-\lambda\sum_{k\in\mathcal{T}_h}\int_{\partial k/\Gamma}\{\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\}\beta_1 h[\nabla(\nabla\boldsymbol{z}_h\cdot\boldsymbol{n})]:(\boldsymbol{w}\otimes\boldsymbol{n})\mathrm{d}s\\
&+\mathcal{N}'_{\Gamma}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h).
\end{aligned}
\tag{45}
$$

The rest of the proof for Lemma 1 becomes trivial, as we just need to combine the cell integral and the internal face integral terms derived in Eq. (45), respectively. It should be noted that the last two terms are the boundary face residual terms, which will be considered later in Lemma 2. Then, one can rewrite the above formula according to the adjoint residual equation (38), which gives the final formulation of cell residual term $\boldsymbol{R}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ as presented in Eq. (40) and internal face terms $\boldsymbol{r}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$, $\boldsymbol{\rho}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ and $\boldsymbol{\sigma}^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ as defined in Eq. (41) for the adjoint residual formulation in Lemma 1. □

From Eq. (40) and Eq. (41), we can see that with the exact solution $\boldsymbol{u}$ of the primal problem and the exact solution $\boldsymbol{z}$ of the adjoint problem, the cell residual term $\boldsymbol{R}^*[\boldsymbol{u}](\boldsymbol{z})=0$. Whereas all the internal face terms $\boldsymbol{r}^*[\boldsymbol{u}](\boldsymbol{z})$, $\boldsymbol{\rho}^*[\boldsymbol{u}](\boldsymbol{z})$ and $\boldsymbol{\sigma}^*[\boldsymbol{u}](\boldsymbol{z})$ are equal to zero, only if $\omega=1$, which verifies that the original DDG method is not adjoint consistent with respect to the discrete adjoint problem defined in Eq. (36), while the DDG(IC) method and the SDDG method show the potential to be adjoint consistent with appropriate boundary treatment, which is described in Lemma 2.

Lemma 2. *The boundary face residual terms defined in the adjoint residual formulation Eq. (38) can be given as*

$$
\begin{aligned}
\boldsymbol{r}^*_{\Gamma}[\boldsymbol{u}_h](\boldsymbol{z}_h)=&(\mathcal{H}'_{\boldsymbol{u}^+}+\mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}^-_{\Gamma_w})'[\boldsymbol{u}^+_h])^{\mathrm{T}}(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\\
&-\big(\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}^+_h]\nabla\boldsymbol{u}_h\big)^{\mathrm{T}}:(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\\
&-\big(\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}^+_h](\frac{\beta_0}{h}(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}^+_h)-\boldsymbol{u}^+_h)\otimes\boldsymbol{n})\big)^{\mathrm{T}}:(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\\
&-\big(\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}^+_h):\frac{\beta_0}{h}(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}^+_h]-\boldsymbol{I})\otimes\boldsymbol{n})\big)^{\mathrm{T}}(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\\
&-\lambda\big((\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w})'[\boldsymbol{u}^+_h](\frac{\beta_0}{h}(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n})\big)^{\mathrm{T}}:(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}^+_h)-\boldsymbol{u}^+_h)\otimes\boldsymbol{n}\\
&-\lambda\big(\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}^+_h):(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}^+_h]-\boldsymbol{I})\otimes\boldsymbol{n}\big)^{\mathrm{T}}\frac{\beta_0}{h}(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\\
\boldsymbol{\rho}^*_{\Gamma}[\boldsymbol{u}_h](\boldsymbol{z}_h)=&-\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w}(\boldsymbol{u}^+_h)(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\qquad on\qquad\Gamma_w,
\end{aligned}
\tag{46}
$$

based on the spatial discretizations of the DDG(IC) method ($\lambda=0$) and the SDDG method ($\lambda=1$), respectively.

Proof. It is clear that by substituting the exact primal solution $\boldsymbol{u}$ and exact adjoint solution $\boldsymbol{z}$ into the modified output functional, the modified output functional is consistent with the output functional defined in Eq. (27). Thus, based on the previous derivation of Eq. (45), the rest of the proof can be achieved by further deriving the boundary linearization $\mathcal{N}'_{\Gamma}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)$ based on the spatial discretization of the DDG(IC) method and SDDG method in combination with the linearization of the modified output functional proposed in Eq. (37). To be specific, the linearization of the boundary terms

can be further separated into the linearization of the no-slip wall boundary terms on Γ_w and the farfield boundary terms on Γ_f, which can be given as follows:

$$\begin{aligned}
&\mathcal{N}'_{\Gamma}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)+\int_{\Gamma}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h)\nabla\boldsymbol{z}_h)\mathrm{d}s\\
=&\mathcal{N}'_{\Gamma_w}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)+\int_{\Gamma_w}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h)\mathrm{d}s\\
+&\mathcal{N}'_{\Gamma_f}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)+\int_{\Gamma_f}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h)\mathrm{d}s.
\end{aligned}\tag{47}$$

Here, we ignore the analysis of the farfield boundary terms and only focus on the derivation of the linearization for the no-slip wall boundary terms. As farfield boundary terms is not involved in the output functional and we can always assume that the farfield boundary is far enough and the influence towards the calculation of the output functional can be neglectable. The linearization of the no-slip wall boundary terms can be further given as follows:

$$\begin{aligned}
&\mathcal{N}'_{\Gamma_w}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)+\int_{\Gamma_w}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h)\mathrm{d}s\\
=&\int_{\Gamma_w}(\mathcal{H}'_{\boldsymbol{u}^+}+\mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}_{\Gamma_w}^-)'[\boldsymbol{u}_h^+])\boldsymbol{w}\boldsymbol{z}_h\mathrm{d}s\\
-&\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h^+]\boldsymbol{w}(\frac{\beta_0}{h}(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}):\boldsymbol{z}_h\otimes\boldsymbol{n}\mathrm{d}s\\
-&\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h^+]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}):\boldsymbol{z}_h\otimes\boldsymbol{n}\mathrm{d}s\\
-&\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h^+]\boldsymbol{w}\nabla\boldsymbol{u}_h:\boldsymbol{z}_h\otimes\boldsymbol{n}\mathrm{d}s-\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla\boldsymbol{w}:\boldsymbol{z}_h\otimes\boldsymbol{n}\mathrm{d}s\\
+&\lambda\int_{\Gamma_w}(\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w})'[\boldsymbol{u}_h^+]\boldsymbol{w}(\frac{\beta_0}{h}(-\boldsymbol{z}_h)\otimes\boldsymbol{n}):(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}\mathrm{d}s\\
+&\lambda\int_{\Gamma_w}\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(-\boldsymbol{z}_h)\otimes\boldsymbol{n}):(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h^+]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}\mathrm{d}s\\
+&\int_{\Gamma_w}(\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w})'[\boldsymbol{u}_h^+]\boldsymbol{w}\nabla\boldsymbol{z}_h:(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}\mathrm{d}s\\
+&\int_{\Gamma_w}\boldsymbol{G}^{\mathrm{T}}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h:(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h^+]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}\mathrm{d}s\\
+&\int_{\Gamma_w}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h)\mathrm{d}s.
\end{aligned}\tag{48}$$

Similarly, we get the linearization of the modified output functional defined in Eq. (37) as follows:

$$\begin{aligned}
\tilde{J}'[\boldsymbol{u}_h](\boldsymbol{w})=&\int_{\Gamma_w}(\mathcal{H}'_{\boldsymbol{u}^+}+\mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}_{\Gamma_w}^-)'[\boldsymbol{u}_h^+])\boldsymbol{w}\hat{\boldsymbol{z}}_{\Gamma_w}\mathrm{d}s\\
&-\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h^+]\boldsymbol{w}\nabla\boldsymbol{u}_h:\hat{\boldsymbol{z}}_{\Gamma_w}\otimes\boldsymbol{n}\mathrm{d}s-\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla\boldsymbol{w}:\hat{\boldsymbol{z}}_{\Gamma_w}\otimes\boldsymbol{n}\mathrm{d}s\\
&-\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h^+]\boldsymbol{w}(\frac{\beta_0}{h}(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}):\hat{\boldsymbol{z}}_{\Gamma_w}\otimes\boldsymbol{n}\mathrm{d}s
\end{aligned}$$

$$
\begin{aligned}
&-\int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h^+]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}):\hat{\boldsymbol{z}}_{\Gamma_w}\otimes\boldsymbol{n}\mathrm{d}s \\
&+\lambda\int_{\Gamma_w}(\boldsymbol{G}_{\Gamma_w}^{\mathrm{T}})'[\boldsymbol{u}_h^+]\boldsymbol{w}(\frac{\beta_0}{h}(-\hat{\boldsymbol{z}}_{\Gamma_w})\otimes\boldsymbol{n}):(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}\mathrm{d}s \\
&+\lambda\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(-\hat{\boldsymbol{z}}_{\Gamma_w})\otimes\boldsymbol{n}):(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h^+]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}\mathrm{d}s.
\end{aligned}
\tag{49}
$$

We note that with the exact primal solution $\boldsymbol{u}$ and the exact adjoint solution $\boldsymbol{z}$ derived on the no-slip wall boundary, the last three terms of Eq. (48) vanish. Then, from the rest of the boundary terms in Eq. (48) and the boundary terms derived from the linearization of the modified output functional in Eq. (49), we can obtain

$$
\begin{aligned}
&\tilde{J}'[\boldsymbol{u}_h](\boldsymbol{w})-\mathcal{N}'_{\Gamma_w}[\boldsymbol{u}_h](\boldsymbol{w},\boldsymbol{z}_h)-\int_{\Gamma_w}(\boldsymbol{w}\otimes\boldsymbol{n}):(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{u}_h^+)\nabla\boldsymbol{z}_h)\mathrm{d}s \\
&=\int_{\Gamma_w}(\mathcal{H}'_{\boldsymbol{u}^+}+\mathcal{H}'_{\boldsymbol{u}^-}(\boldsymbol{u}_{\Gamma_w}^-)'[\boldsymbol{u}_h^+])\boldsymbol{w}(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\mathrm{d}s \\
&-\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h]\boldsymbol{w}\nabla\boldsymbol{u}_h:(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\mathrm{d}s-\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)\nabla\boldsymbol{w}:(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\mathrm{d}s \\
&-\int_{\Gamma_w}\boldsymbol{G}'_{\Gamma_w}[\boldsymbol{u}_h]\boldsymbol{w}(\frac{\beta_0}{h}(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}):(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\mathrm{d}s \\
&-\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}):(\hat{\boldsymbol{z}}_{\Gamma_w}-\boldsymbol{z}_h)\otimes\boldsymbol{n}\mathrm{d}s \\
&+\lambda\int_{\Gamma_w}(\boldsymbol{G}_{\Gamma_w}^{\mathrm{T}})'[\boldsymbol{u}_h]\boldsymbol{w}(\frac{\beta_0}{h}(\boldsymbol{z}_h-\hat{\boldsymbol{z}}_{\Gamma_w})\otimes\boldsymbol{n}):(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)-\boldsymbol{u}_h^+)\otimes\boldsymbol{n}\mathrm{d}s \\
&+\lambda\int_{\Gamma_w}\boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+)(\frac{\beta_0}{h}(\boldsymbol{z}_h-\hat{\boldsymbol{z}}_{\Gamma_w})\otimes\boldsymbol{n}):(\boldsymbol{u}'_{\Gamma_w}[\boldsymbol{u}_h]-\boldsymbol{I})\boldsymbol{w}\otimes\boldsymbol{n}\mathrm{d}s.
\end{aligned}
\tag{50}
$$

From Eq. (50) above, the boundary face residual term $\boldsymbol{r}_\Gamma^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ and $\boldsymbol{\rho}_\Gamma^*[\boldsymbol{u}_h](\boldsymbol{z}_h)$ can be given as shown in Eq. (46), which completes the proof of Lemma 2. □

From the Lemma 2, it is easy to verify that with the exact primal solution $\boldsymbol{u}$ and exact adjoint solution $\boldsymbol{z}$ on the wall boundary, the adjoint boundary residuals $\boldsymbol{r}_\Gamma^*[\boldsymbol{u}](\boldsymbol{z})$ and $\boldsymbol{\rho}_\Gamma^*[\boldsymbol{u}](\boldsymbol{z})$ are all equal to zero.

Finally, based on the conclusions from Lemma 1 and Lemma 2, the proof of the Theorem 1 becomes obvious. By substituting the exact primal solution $\boldsymbol{u}$ and exact adjoint solution $\boldsymbol{z}$ into the adjoint cells residual term, adjoint internal face residual terms and adjoint boundary residual terms defined in Lemma 1 and Lemma 2, all these adjoint residual terms are equal to zero, thus, Eq. (39) is satisfied, if they are derived based on the linearization of the spatial discretization of the DDG(IC) method or SDDG method together with the modified output functional, respectively. Thus, we have proved that for the spatial discretization defined in Eq. (14) and Eq. (15), the original DDG method ($\omega=0$, $\lambda=0$) is not adjoint consistent, while the DDG(IC) method ($\omega=1$, $\lambda=0$) and the SDDG method ($\omega=1$, $\lambda=1$) can be adjoint consistent with the modified output functionals given in Eq. (37), respectively, for the model problem.

Remark 1. The reason why adjoint consistency is important lies in that adjoint consistency guarantees that the spatial discretization, either by global refinement or during adjoint adaptation, delivers optimal order of convergence with respect to the computed error of the output functional, which equals to $O(h^{2p})$ for the compressible Navier–Stokes equations, if the adjoint solution is smooth enough.

Remark 2. The modification towards the target functional is closely related with the specific treatment of both inviscid and viscous fluxes in spatial discretization. For example, the inviscid boundary flux term defined in Eq. (20) can also be treated in an alternative way as follows:

$$
\int_{\Gamma_w}\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+))\cdot\boldsymbol{n}\cdot\boldsymbol{v}_h^+\mathrm{d}s,
\tag{51}
$$

which will lead to a different formulation of modification towards the original output functional $J(\boldsymbol{u}_h)$ as follows:

$$\begin{aligned}\tilde{J}(\boldsymbol{u}_h) = &\int_{\Gamma_w} \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+)) \cdot \boldsymbol{n} \cdot \hat{\boldsymbol{z}}_{\Gamma_w} \mathrm{d}s - \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+) \nabla \boldsymbol{u}_h : \hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n} \mathrm{d}s \\ &- \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}(\boldsymbol{u}_h^+) \frac{\beta_0}{h} (\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+) : \hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n} \mathrm{d}s \\ &+ \lambda \int_{\Gamma_w} \boldsymbol{G}_{\Gamma_w}^{\mathrm{T}}(\boldsymbol{u}_h^+) \frac{\beta_0}{h} (-\hat{\boldsymbol{z}}_{\Gamma_w} \otimes \boldsymbol{n}) : (\boldsymbol{u}_{\Gamma_w}(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+) \otimes \boldsymbol{n} \mathrm{d}s.\end{aligned} \tag{52}$$

Thus, in order to preserve adjoint consistency, the modification to the target functional has to be adjusted according to the specific treatment of numerical fluxes adopted at boundary faces Γ_w, and any acceptable modification should be consistent with the original output functional.

4.4. Error representation formula

For the proceeding of error analysis and h-adaption, we only present the error representation formula for the adjoint-based h-adaptive DDG(IC) method. The specific error representation formula is given in the following theorem.

Theorem 2. *Let $\boldsymbol{u}$ and $\boldsymbol{u}_h$ denote the solutions of Eq. (10) and Eq. (14), respectively, and suppose the dual problem is well posed. Then,*

$$J(\boldsymbol{u}) - J(\boldsymbol{u}_h) = \boldsymbol{\varepsilon}_\Omega(\boldsymbol{u}, \boldsymbol{u}_h; \boldsymbol{z} - \boldsymbol{z}_h) := \sum_{k \in \mathcal{T}_h} \eta_k, \tag{53}$$

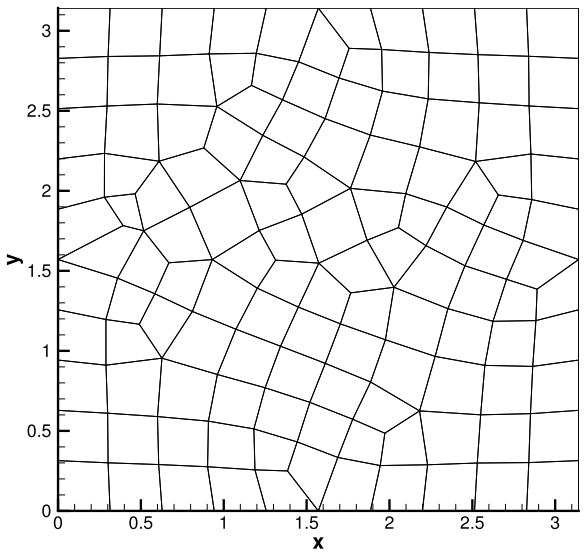
where

$$\begin{aligned}\eta_k = &- \int_k \nabla \cdot \boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h) \cdot \boldsymbol{w}_h \mathrm{d}\Omega + \int_k \nabla \cdot \boldsymbol{\mathcal{F}}^v(\boldsymbol{u}_h, \nabla \boldsymbol{u}_h) \cdot \boldsymbol{w}_h \mathrm{d}\Omega \\ &+ \int_{\partial k \backslash \Gamma} (\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h^+) \cdot \boldsymbol{n} - \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}^+)) \cdot \boldsymbol{w}_h \mathrm{d}s \\ &+ \int_{\partial k \backslash \Gamma} \{\boldsymbol{G}(\boldsymbol{u}_h)\} \frac{\beta_0}{h} [\![\boldsymbol{u}_h]\!] : \boldsymbol{w}_h \otimes \boldsymbol{n} \mathrm{d}s + \frac{1}{2} \int_{\partial k \backslash \Gamma} [\![\boldsymbol{G}(\boldsymbol{u}_h) \nabla \boldsymbol{u}_h]\!] \cdot \boldsymbol{w}_h \mathrm{d}s \\ &+ \int_{\partial k \backslash \Gamma} \{\boldsymbol{G}(\boldsymbol{u}_h)\} \beta_1 h [\nabla (\nabla \boldsymbol{u}_h \cdot \boldsymbol{n})] : \boldsymbol{w}_h \otimes \boldsymbol{n} \mathrm{d}s \\ &- \int_{\partial k \backslash \Gamma} \boldsymbol{G}^{\mathrm{T}}(\widehat{\boldsymbol{u}_h) \nabla} \boldsymbol{w}_h : [\![\boldsymbol{u}_h]\!] \mathrm{d}s \\ &+ \int_{\Gamma} (\boldsymbol{\mathcal{F}}^c(\boldsymbol{u}_h^+) \cdot \boldsymbol{n} - \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_\Gamma^-(\boldsymbol{u}_h^+), \boldsymbol{n}^+)) \cdot \boldsymbol{w}_h \mathrm{d}s \\ &- \int_{\Gamma} (\boldsymbol{G}_\Gamma(\boldsymbol{u}_h^+) \nabla \boldsymbol{u}_h - \boldsymbol{G}_\Gamma(\widehat{\boldsymbol{u}_h) \nabla} \boldsymbol{u}_h) : \boldsymbol{w}_h \otimes \boldsymbol{n} \mathrm{d}s \\ &- \int_{\Gamma} \boldsymbol{G}_\Gamma^{\mathrm{T}}(\widehat{\boldsymbol{u}_h) \nabla} \boldsymbol{w}_h : (\boldsymbol{u}_\Gamma(\boldsymbol{u}_h^+) - \boldsymbol{u}_h^+) \otimes \boldsymbol{n} \mathrm{d}s,\end{aligned} \tag{54}$$

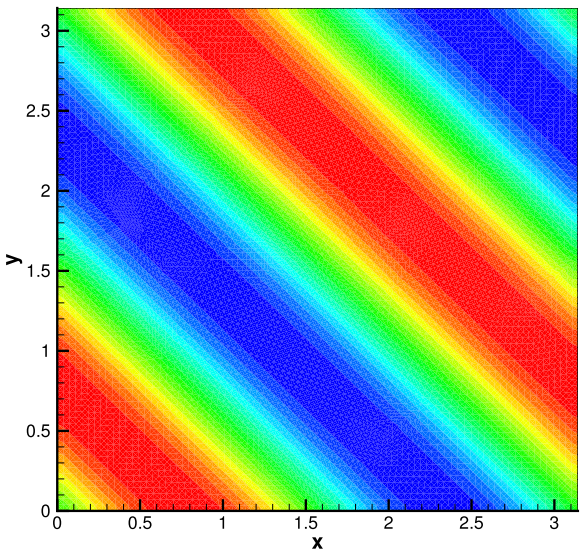
with $\boldsymbol{w}_h = \boldsymbol{z} - \boldsymbol{z}_h$ for all $\boldsymbol{z}_h$ in $\mathcal{V}_{h,p}$.

Proof. The proof of this theorem is similar to the proof of Proposition 4.1 in [13], thus, omit here. □

It should be noted that the above error estimation formula still involve the exact solutions $\boldsymbol{u}$ and $\boldsymbol{z}$ from the primal problem and the adjoint problem, respectively, which needs to be appropriately approximated in practical implementations. Currently in this work, in order to eliminate the appearance of exact primal solution $\boldsymbol{u}$, all the linearization is performed with respect to the numerical solution $\boldsymbol{u}_h$ of primal solver. Moreover, the exact adjoint solution $\boldsymbol{z}$ is replaced by $\hat{\boldsymbol{z}}_h$, which



(a) Unstructured mesh, 121 cells, 142 nodes.



(b) Solution contours of x-momentum, 36 contours from 3.8 to 4.2.

Fig. 3. Mesh and solution contours of model problem with a source term.

is approximately computed by using one order higher polynomial space compared to the solution space used for the primal solution $\boldsymbol{u}_h$. $\boldsymbol{z}_h$ is obtained by projecting the adjoint solution $\hat{\boldsymbol{z}}_h$ from higher-order polynomial space to the original polynomial space as same as $\boldsymbol{u}_h$. Finally, with the above approximation, the formula of error estimate used in our practical implementation can be given as

$$J(\boldsymbol{u}) = \tilde{J}(\boldsymbol{u}_h) + \boldsymbol{\varepsilon}_\Omega(\boldsymbol{u}_h, \boldsymbol{u}_h; \hat{\boldsymbol{z}}_h - \boldsymbol{z}_h), \tag{55}$$

where $\boldsymbol{\varepsilon}_\Omega(\boldsymbol{u}_h, \boldsymbol{u}_h; \hat{\boldsymbol{z}}_h - \boldsymbol{z}_h) := \sum_{k\in\mathcal{T}_h} \eta_k$, the definition of η_k for a given element k is same as Eq. (54).

5. Numerical examples

In this section, we present several typical numerical examples to examine and verify the previous theoretical analysis of adjoint consistency for the DDG method, the DDG(IC) method and the SDDG method, meanwhile we will investigate the performance of the adjoint-based h-adaptive DDG(IC) method for solving the steady state compressible Navier–Stokes equations.

5.1. Model problem with a source term

In this first example, we consider a simple model problem in order to examine and verify the adjoint consistency of the original DDG method, the DDG(IC) method and the SDDG method. In specific, we let $\Omega = (0, \pi)^2$, and supplement the steady state compressible Navier–Stokes equations with an inhomogeneous forcing function $\mathcal{S}$, which is chosen so that the analytical solution to Eq. (10) with source term $\mathcal{S}$ is given by

$$\boldsymbol{u} = (sin(2(x+y)) + 4, 0.2sin(2(x+y)) + 4, 0.2sin(2(x+y)) + 4, (sin(2(x+y)) + 4)^2),$$

the dynamic viscosity coefficient has been set to be constant $\mu = 0.1$. In this section, we shall be interested in measuring the discretization error of a given output functional $J(\cdot)$, to be specific, we consider the weighted mean-value of the density, i.e.,

$$J(\boldsymbol{u}) = J_\Omega(\boldsymbol{u}) = \int_\Omega \rho\psi d\Omega,$$

where $\psi = sin(\pi x)sin(\pi y)$; thereby, the true value of the functional is given by $J(u) = 1.168587648689877$ according to [14].

Initially, an unstructured quadrilateral grid is adopted for this test case as shown in Fig. 3(a), which contains 121 cells, 142 nodes, and the solution of the x-momentum on a global refined mesh is shown in Fig. 3(b). We present a comparison of the error in the $L_2(\Omega)$-norm with the number of elements for both DG(p_1) method and DG(p_2) method. From Table 1

Table 1
Comparison of order of convergence in terms of error in computed output functional.

Number of cells	Total DoFs	DDG(p_1)	Order	DDG(IC)(p_1)	Order	SDDG(p_1)	Order
121	1452	1.04e−01		1.39e−01		2.28e−01	
484	5808	2.38e−02	2.13	3.31e−02	2.07	6.27e−02	1.86
1936	23232	5.81e−03	2.03	8.47e−03	1.97	1.77e−02	1.82
7744	92928	1.43e−04	2.02	2.19e−03	1.95	4.75e−03	1.90

Number of cells	Total DoFs	DDG(p_2)	Order	DDG(IC)(p_2)	Order	SDDG(p_2)	Order
121	2904	2.25e−03		1.51e−03		1.93e−03	
484	11616	2.70e−04	3.05	1.11e−04	3.77	9.53e−05	4.33
1936	46464	2.95e−05	3.19	9.31e−06	3.58	7.93e−06	3.58
7744	185856	3.53e−06	3.06	4.65e−07	4.32	4.25e−07	4.22

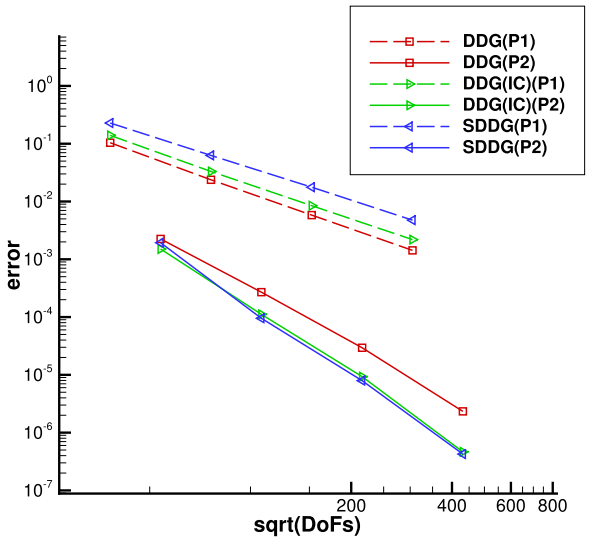


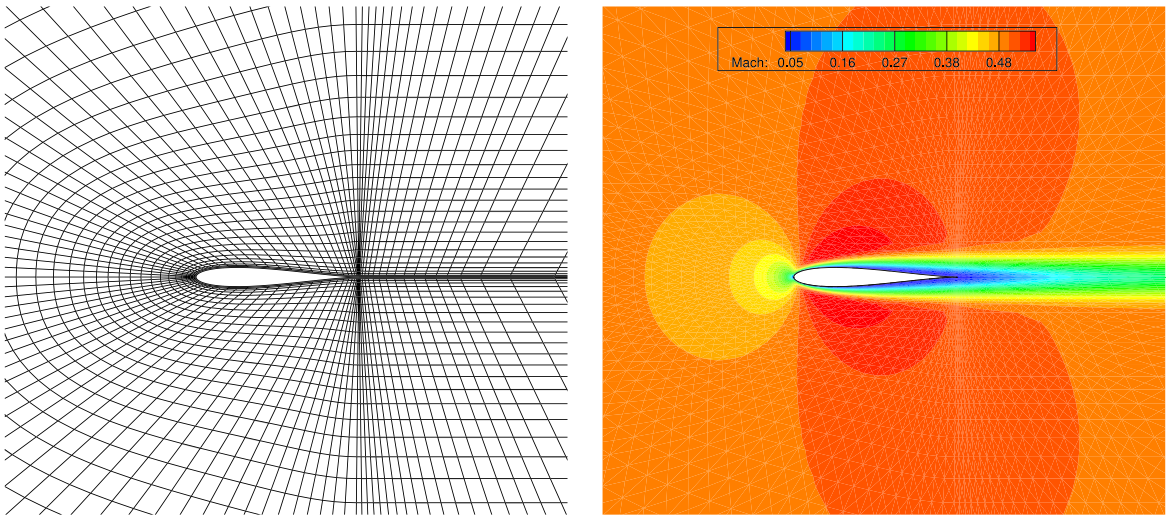
Fig. 4. Grid convergence study in computed output functional for model problem with a source term.

and Fig. 4, we observe that the error in the computed output functional $J(\cdot)$ behaves approximately $O(h^{2p})$ for the DDG(IC) and SDDG methods and only $O(h^{p+1})$ for the original DDG method, for each fixed p as the mesh is uniformly refined. These rates of convergence for the error in the computed output functional $J(\cdot)$ is in complete agreement with the corresponding analysis of the convergence behavior as our analysis presented in the Lemma 1 of Section 4. As only adjoint consistent discretization can deliver the optimal order of convergence $O(h^{2p})$ for the compressible Navier–Stokes equations with respective to the error in the computed output functional. The test case verifies our proof that the original DDG method is not adjoint consistent, while the DDG(IC) method and the SDDG method are adjoint consistent for this test case, thus, can deliver the optimal order of convergence with respect to the computed error of the output functional.

5.2. Laminar flow over a Joukowski airfoil

This test case is designed as a verification case of the modification towards the output functional in solving the compressible Navier–Stokes equations for the model problem described in Section 4.1. The Joukowski airfoil is used for this test as the cusped tailing edge removes the inviscid singularity at the tailing edge. The freestream Mach number is 0.5, the Reynolds number is 1000 based on chord, the angle of attack is 0 degree. The dynamic viscosity is constant with the Prandtl number is fixed to $Pr = 0.72$. An adiabatic wall boundary is assumed along the airfoil in this test case.

In order to perform the grid convergence study, a set of fourth-order curved grids which contains 512, 2,048, 8,192, 32,768 cells, respectively, is used in this test case as shown in Fig. 5(a). The grids can be obtained freely from the website of the fourth workshop of high-order CFD methods [41]. The farfield boundary is located in a distance of over 100 chords away from the airfoil. It should be noted that this set of grids is specially designed and optimized for this test case and can be regarded as challenging for the simulation of laminar flows. The boundary layer mesh is not well orthogonal to the surface of the airfoil and has a very high growth rate, nearly 5 in the normal direction. Moreover, in order to obtain the optimal order of convergence, the height of the first layer grid decreases rapidly, e.g. at leading edge from approximately 1.0e−02 on the coarsest mesh to only 4.0e−05 on the finest mesh.



(a) Structured mesh, 8,192 cells, 131,712 nodes, Q4 grid.

(b) Solution contours of Mach number, 24 contours from 0.05 to 0.55.

Fig. 5. Mesh and solution contours of laminar flow over a Joukowski airfoil. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2
Grid convergence study in computed output functional for laminar flow over a Joukowski airfoil.

Cell number	DDG(IC)(p_1) method		SDDG(p_1) method	
	C_d error	Order	C_d error	Order
512	2.818e−02		5.880e−02	
2048	1.079e−02	1.39	1.964e−02	1.58
8192	3.058e−03	1.82	5.313e−03	1.89
32768	7.093e−04	2.11	1.293e−03	2.04
Cell number	DDG(IC)(p_2) method		SDDG(p_2) method	
	C_d error	Order	C_d error	Order
512	7.102e−03		9.156e−03	
2048	8.492e−04	3.064	3.607e−04	4.665
8192	4.604e−05	4.205	4.461e−05	3.010
32768	2.507e−06	4.199	3.432e−06	3.700

In this test case, we consider the estimation of the drag coefficient C_d as the output functional. We remark that the adjoint consistency of the DDG(IC) method and the SDDG method is based on the consistent reformulation of modified output functional $\tilde{J}(\cdot)$, which is given in Theorem 1. Steady state is assumed if the initial residual is dropped by at least 10 orders of magnitude in this work. The reference C_d value for DDG(IC) method is about 1.21897823e−01 based on a computation using DDG(IC)(p_3) method on the finest grids.

Solution contours of Mach number obtained by the DDG(IC) method is shown in Fig. 5(b). Theoretically, the adjoint consistent DDG(IC) method and SDDG method with the modified output functional should be able to deliver the optimal $2p$ order of convergence with respect to the computed error towards output functional. In Table 2 and Fig. 6, we present our numerical results of grid convergence study for both the DDG(IC) method and the SDDG method. From the numerical results, we can see clearly that the error of the computed output functional $\tilde{J}(\cdot)$ behaves approximately $O(h^{2p})$ for both the DDG(IC) and SDDG methods along with the mesh refinement. These rates of convergence for the error of the computed output functional $\tilde{J}(\cdot)$ agrees well with the corresponding analysis as presented in the Theorem 1 of Section 4.

5.3. Laminar flow over a NACA0012 airfoil

In this test case, we investigate the performance of the DDG(IC) method applied for the implementation of adjoint-based h-adaptation. This test case involves a subsonic laminar flow past a NACA0012 airfoil at a Mach number of 0.5, an angle of attack $\alpha = 0^o$, and a Reynolds number of 5,000 based on the freestream velocity and the chord length of the airfoil. An adiabatic wall boundary is assumed along the airfoil in this test case. The computational domain is subdivided into

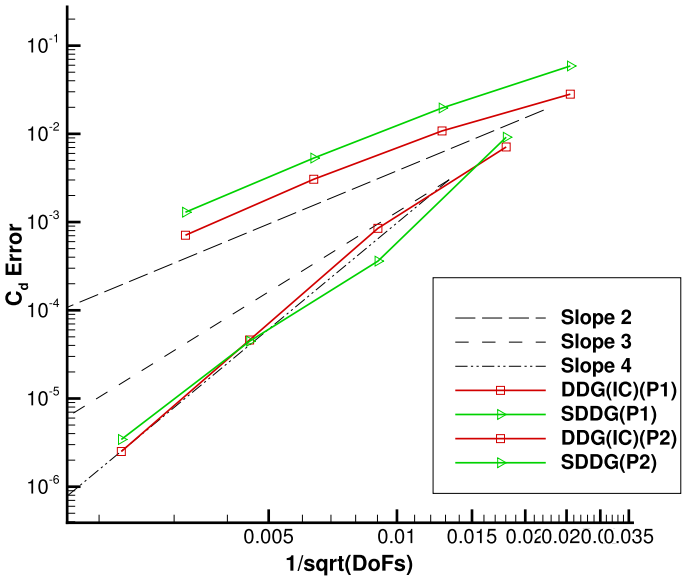


Fig. 6. Grid convergence study in computed output functional for laminar flow over a Joukowski airfoil.

quadrilateral elements, in order to handle the curved boundary geometry, a quadratic quadrilateral grid is adopted in this test as shown in Fig. 7(a). This is a standard laminar test case which has been investigated by many other authors, cf. [12, 14,34,37]. The solution to this problem consists of a strictly subsonic flow which is symmetric about the x-axis.

In this test case, we consider the estimation of the drag coefficient C_d as the target functional. We remark that the adjoint consistency of the proposed DDG(IC) method is based on the consistent reformulation of modified output functional $\tilde{J}(\cdot)$ given in Theorem 1. We note that all the h-adaption is performed with the fixed-number mesh adaptation strategy, with refinement and coarsening fractions as 20% and 10%, respectively. Steady state is assumed if the initial residual is dropped by at least 8 orders of magnitude in this work. The reference C_d value is obtained based on the computation from a relatively fine mesh.

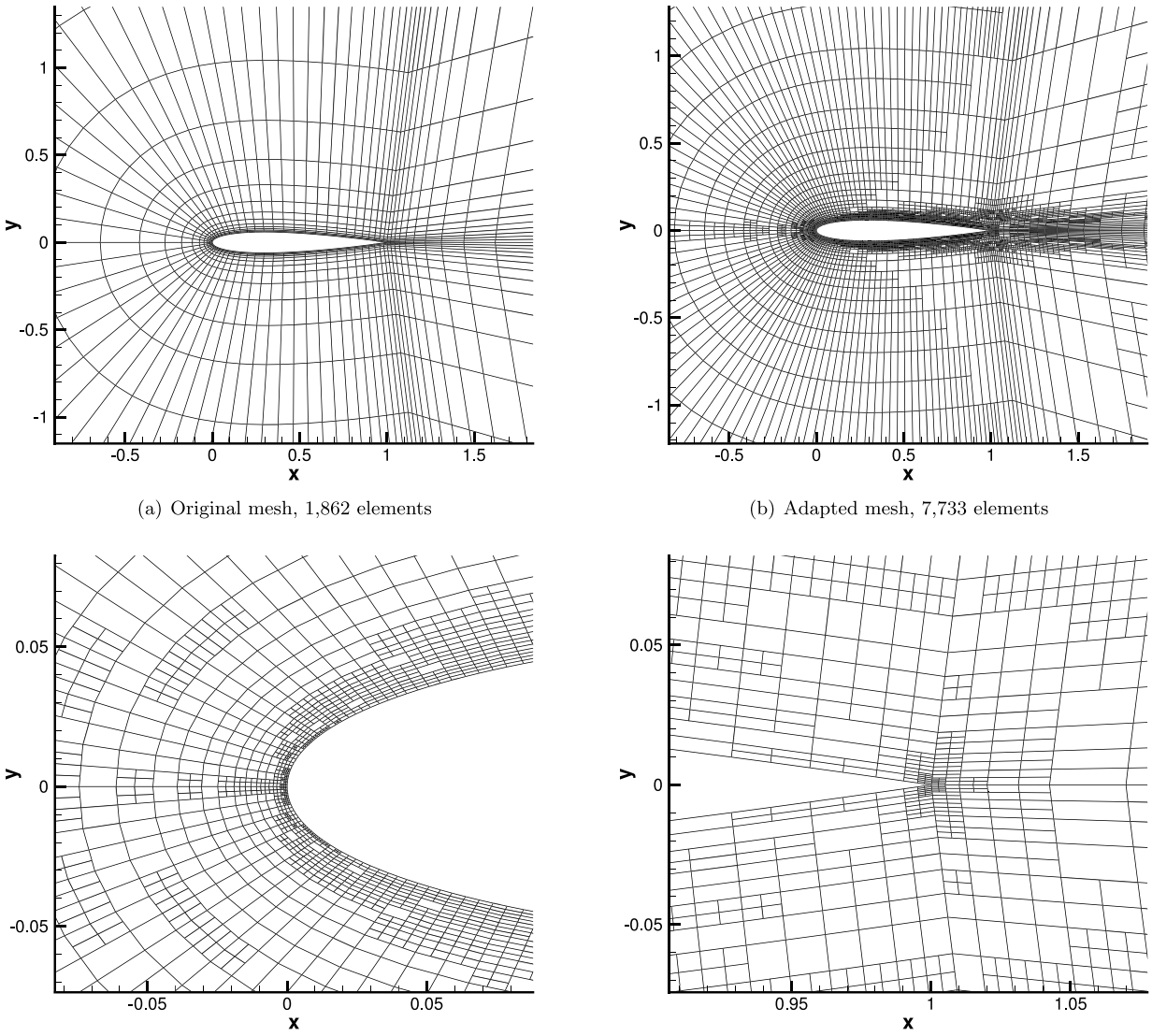
In order to verify the efficiency index of the adjoint-based adaptation, we show the number of elements and degrees of freedom, the true error in the target function $J(\boldsymbol{u}) - \tilde{J}(\boldsymbol{u}_h)$, the computed error representation formula $\sum_{k\in\mathcal{T}_h}\eta_k$, and the corresponding effectivity index $\theta = \sum_{k\in\mathcal{T}_h}\eta_k/(J(\boldsymbol{u}) - \tilde{J}(\boldsymbol{u}_h))$ in Table 3. We see that from the numerical experiment, the DDG(IC)(p_1) method deliver a relatively good estimation of the error of the target functional in the sense that the effectivity index θ is close to one, even on the relatively coarse mesh, while the performance of DDG(IC)(p_2) method also delivers expected results as the effectivity index is also close to one. In Fig. 7, we show the adapted meshes for the approximation of drag coefficients after three times of grid adaptation, which contains 7,733 elements. As one can see that the refinement is mainly concentrated within the vicinity of the airfoil, especially around the leading edge and tailing edge, and the area along the streamline ahead of leading edge and behind the tailing edge as expected.

6. Conclusions

In this paper, we have analyzed and developed an adjoint-based high-order h-adaptive direct discontinuous Galerkin method for the two dimensional steady state compressible Navier–Stokes equations. Analysis of the adjoint consistency has been presented for three DDG type methods, including the original direct discontinuous Galerkin (DDG) method, the direct discontinuous Galerkin method with interface correction (DDG(IC)) and the symmetric direct discontinuous Galerkin method (SDDG). Based on the theoretical analysis, for the model problem considered in this work, the original DDG method is not adjoint consistent, while the DDG(IC) and SDDG methods can be adjoint consistent with appropriate treatment of boundary conditions and correct modification towards the underlying output functionals. Numerical experiments also verified that the DDG(IC) method and SDDG method can achieve the optimal order of accuracy with respect to the error in the computed output functional. Based on the theoretical analysis, an adjoint-based h-adaptive DDG(IC) method is further developed and investigated. The adjoint-based error estimator has been derived based on the DDG(IC) discretization and its effectivity is carefully tested and verified. Future work will be focused on the development of the adjoint-based h-adaptive DDG(IC) method for more practical applications, such as the simulations of turbulent flows.

Acknowledgements

The authors would like to acknowledge the partial support provided by National Natural Science Foundation of China No. 11601024 and No. 91530325, Fundamental Research of Civil Aircraft No. MJ-F-2012-04. Computations in this work have



(a) Original mesh, 1,862 elements

(b) Adapted mesh, 7,733 elements

(c) Zoom in at leading edge

(d) Zoom in at tailing edge

Fig. 7. Original and adapted mesh for subsonic flow over a NACA0012 airfoil.

Table 3
Study of effectivity index of adjoint-based h-adaptive DDG(IC) method for laminar flow over a NACA0012 airfoil.

Number of cells	DDG(IC)(p_1) method			
	Total DoFs	$J(\boldsymbol{u}) - \tilde{J}(\boldsymbol{u}_h)$	$\sum_{k\in\mathcal{T}_h}\eta_k$	θ
1862	22344	1.176e−03	1.111e−03	0.945
2981	35772	2.996e−04	2.822e−04	0.941
4814	57624	1.485e−04	1.425e−04	0.959
7715	92580	7.941e−05	7.776e−05	0.979
Number of cells	DDG(IC)(p_2) method			
	Total DoFs	$J(\boldsymbol{u}) - \tilde{J}(\boldsymbol{u}_h)$	$\sum_{k\in\mathcal{T}_h}\eta_k$	θ
1862	44688	1.586e−05	1.496e−05	0.943
2981	71544	4.355e−06	3.973e−06	0.912
4814	115536	1.228e−06	1.107e−06	0.901
7733	185592	4.671e−07	4.283e−07	0.917

been performed based on deal.II library [40]. The first author would like to thank Prof. M. Galbraith from M.I.T. for providing the reference data for the test case 5.2.

References

[1] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Los Alamos Scientific Laboratory Report, LA-UR-73-479, 1973.
[2] B. Cockburn, S. Hou, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1988) 411–435.
[3] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional system, J. Comput. Phys. 141 (1998) 199–224.
[4] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
[5] H. Luo, J.D. Baum, R. Löhner, A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids, J. Comput. Phys. 227 (2008) 8875–8893.
[6] H. Luo, J.D. Baum, R. Löhner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, J. Comput. Phys. 225 (2007) 686–713.
[7] H. Luo, J.D. Baum, R. Löhner, A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, J. Comput. Phys. 225 (2006) 767–783.
[8] H. Luo, L.Q. Luo, R. Nourgaliev, V.A. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, J. Comput. Phys. 229 (2010) 6961–6978.
[9] B. van Leer, M. Lo, A discontinuous Galerkin method for diffusion based on recovery, AIAA 2007-4083.
[10] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[11] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations, J. Comput. Phys. 228 (2009) 3232–3254.
[12] R. Hartmann, P. Houston, Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: method formulation, Int. J. Numer. Anal. Model. (3) (2006) 1–20.
[13] R. Hartmann, P. Houston, Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: goal-oriented a posteriori error estimation, Int. J. Numer. Anal. Model. (3) (2006) 141–162.
[14] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations, J. Comput. Phys. 227 (2008) 9670–9685.
[15] F. Bassi, S. Rebay, Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and k-w turbulence model equations, Comput. Fluids 34 (2005) 507–540.
[16] G. Gassner, F. Lörcher, C.-D. Munz, A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes, J. Comput. Phys. 224 (2007) 1049–1063.
[17] H.L. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, SIAM J. Numer. Anal. 47 (2009) 675–698.
[18] H.L. Liu, J. Yan, The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections, Commun. Comput. Phys. 8 (2010) 541–564.
[19] H.L. Liu, Optimal error estimates of the direct discontinuous Galerkin method for convection–diffusion equations, Math. Comput. 84 (2015) 2263–2295.
[20] W.-X. Cao, H.L. Liu, Z.-M. Zhang, Superconvergence of the direct discontinuous Galerkin method for convection–diffusion equations, Numer. Methods Partial Differ. Equ. (2016), https://doi.org/10.1002/num.22087.
[21] C. Vidden, J. Yan, A new direct discontinuous Galerkin method with symmetric structure for nonlinear diffusion equations, J. Comput. Math. 31 (2013) 638–662.
[22] M.P. Zhang, J. Yan, Fourier type error analysis of the direct discontinuous Galerkin method and its variations for diffusion equations, J. Sci. Comput. 52 (2012) 638–655.
[23] H.Y. Huang, Z. Chen, J. Li, J. Yan, Direct discontinuous Galerkin method and its variations for second order elliptic equations, J. Sci. Comput. 70 (2017) 744–765.
[24] J. Cheng, X.Q. Yang, X.D. Liu, T.G. Liu, H. Luo, A direct discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, J. Comput. Phys. 327 (2016) 484–502.
[25] J. Cheng, X.D. Liu, T.G. Liu, H. Luo, A parallel, high-order direct discontinuous Galerkin method for the Navier–Stokes equations on 3D hybrid grids, Commun. Comput. Phys. 21 (2017) 1231–1257.
[26] H.Q. Yue, J. Cheng, T.G. Liu, A symmetric direct discontinuous Galerkin method for the compressible Navier–Stokes equations, Commun. Comput. Phys. 22 (2017) 375–392.
[27] J. Cheng, H.Q. Yue, S.J. Yu, T.G. Liu, A direct discontinuous Galerkin method with interface correction for the compressible Navier–Stokes equations on unstructured grids, Adv. Appl. Math. Mech. 10 (1) (2018) 1–21.
[28] Z. Chen, H.Y. Huang, J. Yan, Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes, J. Comput. Phys. 308 (2016) 198–217.
[29] M.P. Zhang, J. Yan, Fourier type super convergence study on DDGIC and symmetric DDG methods, J. Sci. Comput. (2017), https://doi.org/10.1007/s10915-017-0438-3.
[30] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservations laws, SIAM J. Sci. Comput. 24 (2002) 979–1004.
[31] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, J. Comput. Phys. 183 (2002) 508–532.
[32] L. Wang, D.J. Mavriplis, Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations, J. Comput. Phys. 228 (2009) 7643–7661.
[33] D.A. Venditti, D.L. Darmofal, Grid adaptation for functional outputs: application to two-dimensional inviscid flows, J. Comput. Phys. 176 (2002) 40–69.
[34] L. Shi, Z.J. Wang, Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method, J. Comput. Phys. 295 (2015) 261–284.
[35] H.Q. Yue, T.G. Liu, V. Shaydurov, Continuous adjoint-based error estimation and its application to adaptive discontinuous Galerkin method, Appl. Math. Mech. (37) (2016) 1–12.
[36] K.J. Fidkowski, D.L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA J. 49 (2011) 673–694.
[37] R. Hartmann, Adjoint consistency analysis of discontinuous Galerkin discretizations, SIAM J. Numer. Anal. 45 (2007) 2671–2698.
[38] R. Hartmann, T. Leicht, Generalized adjoint consistent treatment of wall boundary conditions for compressible flows, J. Comput. Phys. 300 (2015) 754–778.

[39] J. Lu, An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method, PhD thesis, M.I.T., 2005.
[40] W. Bangerth, R. Hartmann, G. Kanschat, deal.II – a general-purpose object-oriented finite element library, ACM Trans. Math. Softw. 33 (2007), https://doi.org/10.1145/1268776.1268779.
[41] The 4th International Workshop on High-Order CFD Methods, https://how4.cenaero.be/.