Continuous adjoint-based error estimation and its application to adaptive discontinuous Galerkin method*

Huiqiang YUE,   Tiegang LIU†,   V. SHAYDUROV

Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education,
School of Mathematics and Systems Science, Beihang University, Beijing 100191, China

Abstract   An adaptive mesh refinement algorithm based on a continuous adjoint approach is developed. Both the primal equation and the adjoint equation are approximated with the discontinuous Galerkin (DG) method. The proposed adaptive algorithm is used in compressible Euler equations. Numerical tests are made to show the superiority of the proposed adaptive algorithm.

Key words   adaptivity, discontinuous Galerkin (DG), adjoint

Chinese Library Classification   O351.3
2010 Mathematics Subject Classification   76M25, 76D

1   Introduction

In lots of applications, e.g., engineering analysis and design, engineers are interested in not only the whole solution but also its certain values, i.e., output functionals. For example, in computational aerodynamics, lift and drag are usually more important than the whole solution in the entire space[1]. In such computations, engineers are interested in the accuracy of these output functionals. Considering the finite computational resources, the adaptive approach is an efficient way to improve the calculation precision of these functionals. The adaptive approach usually uses local error indicators to guide the mesh refinement. The indicators derived based on the solution features such as the large gradient in flow variables and flow discontinuities have been widely used in adaptive algorithms. Although the feature-based adaptation has had some successes, it is liable to overrefine the regions that have little effect on the output functionals[2–3].

In order to derive reliable and efficient adaptive algorithms to produce the desired increase in the accuracy of the output functionals, adjoint-based adaptive methods have been widely developed and studied[4–13]. Becker and Rannacher[4–5] proposed the dual weighted residual method based on the property of Galerkin orthogonality of the finite element method. Venditti[6] and Venditti and Darmofal[7] derived the adaptive procedure by the local error indicators constructed from both the primal solution and the adjoint solution in the standard finite volume discretization. Hartmann and Honston[8–9] and Hartmann[10] made extensive studies on the adaptive discontinuous Galerkin (DG) method based on an adjoint approach for aerodynamic

† Corresponding author, E-mail: liutg@buaa.edu.cn

flows. Wang and Mavriplis[11] developed the h-p adaptation approach for compressible Euler equations. Fidkowski[12] developed anisotropic and adjoint-based mesh adaptation for the DG discretization and cut-cell meshing. However, all these studies need output-based adjoint solutions as the weight in order to represent the sensitivity of the output functionals to the local primal residuals.

In shape optimization designs, the adjoint problems can be derived by two ways, i.e., the continuous approach and the discrete approach. Both of these approaches have made some successes. However, up to now, most of the adjoint-based adaptive algorithms have been developed only from the discrete adjoint approach[3–15]. In the discrete adjoint approach, discretization is made for the primal equations firstly, and then the adjoint equations are obtained by linearizing the discrete primal problems. In this procedure, a full Jacobian matrix is needed. Therefore, if we can get it from the primal flow solver, the implementation of the adjoint equation will be much easier. However, if not, the calculation of the full Jacobian matrix will be very complicated. Moreover, in the continuous adjoint approach, the adjoint partial differential equations (PDEs) are derived from the primal PDEs directly, and then these equations are discretized by the numerical schemes which may be different from the primal problem and may be independent of the code structure of the primal problem. More importantly, we can enforce the boundary conditions more rigorously in the continuous adjoint approach. Fidkowski and Roe[16] proposed an entropy adjoint approach, and used the continuous adjoint approach to derive the error indicators. Li et al.[17] developed an adaptive mesh refinement algorithm, and showed the promising results based on the continuous adjoint approach in the spectral difference framework. In this paper, we will provide an approach to construct the adaptive error indicator based on the continuous adjoint approach in the DG method, and show its efficiency for the error estimation.

The paper is organized as follows. In Section 2, we introduce the primal equations and the DG discretization used in this work. In Section 3, we introduce the derivation of the continuous adjoint equations and construct the adjoint-based error indicator. In Section 4, we use the continuous adjoint-based error to estimate the two-dimensional (2D) Euler equations. In Section 5, we describe the numerical implementation of our work. In Section 6, we discuss the numerical experiments to highlight the superiority of the proposed algorithm.

2   Primal equations and discretization

2.1   Governing equations

Let Ω be the computational domain in $\mathbb{R}^2$. Then, the 2D steady-state Euler equations can be written as follows:

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

where $\boldsymbol{u} = (\rho, \rho u, \rho v, \rho E)^{\mathrm{T}}$ is the solution vector consisting of the density ρ, $\boldsymbol{V} = (u, v)^{\mathrm{T}}$ is the velocity, and E is the total energy per unit mass. The flux functional $\mathcal{F}(\boldsymbol{u})$ is defined by

$$\mathcal{F}(\boldsymbol{u}) = \big(\boldsymbol{f}_1(\boldsymbol{u}), \boldsymbol{f}_2(\boldsymbol{u})\big) = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ (\rho E + p)u & (\rho E + p)v \end{pmatrix}, \tag{2}$$

where p is the pressure. The system is closed by

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

where γ is the heat capacity ratio, which is 1.4 for the diatomic gas.

2.2   Standard DG discretization

Denote $\mathcal{T}_h = \{k\}$ the subdivision of the domain Ω. Then, the standard DG discretization of Eq. (1) can be given as follows: find $\boldsymbol{u}_h \in \mathcal{V}_{h,p}$ such that

$$\mathcal{R}(\boldsymbol{u}_h, \boldsymbol{\varphi}) = \sum_{k \in \mathcal{T}_h} \Big(-\int_k \nabla \boldsymbol{\varphi}^{\mathrm{T}} \cdot \mathcal{F}(\boldsymbol{u}_h) \mathrm{d}\boldsymbol{x} + \int_{\partial \kappa \backslash \partial \Omega} \boldsymbol{\varphi}^{\mathrm{T}} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}) \mathrm{d}s$$

$$+ \int_{\partial k \cap \partial \Omega} \boldsymbol{\varphi}^{\mathrm{T}} \mathcal{H}^b(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-(\boldsymbol{u}_h^+), \boldsymbol{n}) \mathrm{d}s \Big) = 0, \quad \forall \boldsymbol{\varphi} \in \mathcal{V}_{h,p}, \tag{4}$$

where

$$\mathcal{V}_{h,p} = \{\boldsymbol{u} \in [L_2(\Omega)]^d : \boldsymbol{u}|_K \in [\mathcal{P}^p]^d, \quad \forall k \in \mathcal{T}_h\}.$$

In the above equations, d is the dimension of $\boldsymbol{u}$, $\mathcal{P}^p$ is the polynomial with the degree p, and $(\cdot)^+$ and $(\cdot)^-$ are the interior trace and the outer trace on ∂k, respectively. $\mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n})$ and $\mathcal{H}^b(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-(\boldsymbol{u}_h^+), \boldsymbol{n})$ are the numerical flux functionals depending on the interior and $\partial \Omega$ meeting the physical boundary conditions, respectively.

3   Adjoint-based error estimation

3.1   Continuous adjoint equations

Consider the following primal problem arising from Eq. (1):

$$\nabla \cdot \mathcal{F}(\boldsymbol{u}) = 0 \quad \text{in} \quad \Omega, \tag{5}$$

$$\mathcal{B}\boldsymbol{u} = 0 \quad \text{on} \quad \Gamma, \tag{6}$$

where $\mathcal{B}$ is a boundary operator, $\Gamma = \partial\Omega$, $\boldsymbol{u} \in \mathcal{V}$, and $\mathcal{V}$ is the functional space of the analytical solution. Then, the output functional which we are interested can be defined by

$$I(\boldsymbol{u}) = \int_\Omega i_\Omega(\boldsymbol{u}) \mathrm{d}x + \int_\Gamma i_\Gamma(\boldsymbol{u}) \mathrm{d}s, \tag{7}$$

where $i_\Omega(\cdot)$ and $i_\Gamma(\cdot)$ are the operators which may be nonlinear. Let $\boldsymbol{u}_h$ be the numerical approximation of $\boldsymbol{u}$ in the space $\mathcal{V}_{h,p} \subset \mathcal{V}$. Then, we denote the error between $\boldsymbol{u}$ and $\boldsymbol{u}_h$ by $\widetilde{\boldsymbol{u}}$, i.e.,

$$\boldsymbol{u} = \boldsymbol{u}_h + \widetilde{\boldsymbol{u}}.$$

For the analytical solution $\boldsymbol{u} \in \mathcal{V}$ and $\boldsymbol{\psi} \in \mathcal{V}$, the residual

$$\mathcal{R}(\boldsymbol{u}, \boldsymbol{\psi}) = 0.$$

Therefore, we can add the residual to the output functional $I(\boldsymbol{u})$ without changing its value, i.e.,

$$\overline{I}(\boldsymbol{u}) = I(\boldsymbol{u}) = \int_\Omega i_\Omega(\boldsymbol{u}) \mathrm{d}x + \int_\Gamma i_\Gamma(\boldsymbol{u}) \mathrm{d}s - \mathcal{R}(\boldsymbol{u}, \boldsymbol{\psi}). \tag{8}$$

In this way,

$$\overline{I}(\boldsymbol{u}_h) = \int_\Omega i_\Omega(\boldsymbol{u}_h) \mathrm{d}x + \int_\Gamma i_\Gamma(\boldsymbol{u}_h) \mathrm{d}s - \mathcal{R}(\boldsymbol{u}_h, \boldsymbol{\psi}). \tag{9}$$

Therefore, the error of $\overline{I}(\boldsymbol{u})$ can be given by

$$\overline{I}(\boldsymbol{u}) - \overline{I}(\boldsymbol{u}_h) = \int_\Omega \big(i_\Omega(\boldsymbol{u}) - i_\Omega(\boldsymbol{u}_h)\big) \mathrm{d}x + \int_\Gamma \big(i_\Gamma(\boldsymbol{u}) - i_\Gamma(\boldsymbol{u}_h)\big) \mathrm{d}s$$

$$- (\mathcal{R}(\boldsymbol{u}, \boldsymbol{\psi}) - \mathcal{R}(\boldsymbol{u}_h, \boldsymbol{\psi})). \tag{10}$$

Then, we linearize the right-hand side of Eq. (10) to get the first-order approximations as follows:

$$\int_\Omega \left(i_\Omega(\boldsymbol{u}) - i_\Omega(\boldsymbol{u}_h)\right)\mathrm{d}x = \int_\Omega i_\Omega'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}x,$$

(handwritten: →应是是 u, 而非 u_h !)

$$\int_\Gamma \left(i_\Gamma(\boldsymbol{u}) - i_\Gamma(\boldsymbol{u}_h)\right)\mathrm{d}s = \int_\Omega i_\Gamma'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}s,$$

$$\mathcal{R}(\boldsymbol{u}, \boldsymbol{\psi}) - \mathcal{R}(\boldsymbol{u}_h, \boldsymbol{\psi}) = \mathcal{R}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}, \boldsymbol{\psi}),$$

where the prime on $(\cdot)'$ means the Fréchet derivative linearized at the term in the square brackets. Thus, we have

(handwritten: $[I(u) - R(u,v)] - [I(u_h) - R(u_h,v)] = I(u) - I(u_h) + R(u_h,v) - R(u,v)$)

$$\overline{I}(\boldsymbol{u}) - \overline{I}(\boldsymbol{u}_h) = \int_\Omega i_\Omega'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}x + \int_\Omega i_\Gamma'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}s - \mathcal{R}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}, \boldsymbol{\psi}). \tag{11}$$

In order to make $\overline{I}(\boldsymbol{u}) - \overline{I}(\boldsymbol{u}_h)$ be independent of $\widetilde{\boldsymbol{u}}$, in other words, to find a suitable $\boldsymbol{\psi}$ such that the error $\widetilde{\boldsymbol{u}}$ has no influence on $\overline{I}(\boldsymbol{u}) - \overline{I}(\boldsymbol{u}_h)$, we consider the following problem: $\forall \widetilde{\boldsymbol{u}} \in \mathcal{V}$, find $\boldsymbol{\psi}$ such that

$$\overline{I}(\boldsymbol{u}) - \overline{I}(\boldsymbol{u}_h) = 0. \tag{12}$$

Therefore, $\boldsymbol{\psi}$ should satisfy

$$\int_\Omega i_\Omega'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}x + \int_\Omega i_\Gamma'[\boldsymbol{u}](\widetilde{\boldsymbol{u}})\mathrm{d}s = \mathcal{R}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}, \boldsymbol{\psi}). \tag{13}$$

Substituting

$$\mathcal{R}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}, \boldsymbol{\psi}) = (\nabla \cdot \mathcal{F}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}), \boldsymbol{\psi})_\Omega$$

into Eq. (13), we get

(handwritten: 又剩在没 ∇· 应是为 $R(u,\psi) = (\nabla F(u), \psi)$. 而非§散之后的格式!)

$$(\nabla \cdot \mathcal{F}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}), \boldsymbol{\psi})_\Omega = (i_\Omega'[\boldsymbol{u}], \widetilde{\boldsymbol{u}})_\Omega + (i_\Gamma'[\boldsymbol{u}], \widetilde{\boldsymbol{u}})_\Gamma. \tag{14}$$

Integrating the left-hand side of the above equation by parts, we get

$$(\nabla \cdot \mathcal{F}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}), \boldsymbol{\psi})_\Omega = (\boldsymbol{n} \cdot \mathcal{F}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}), \boldsymbol{\psi})_\Gamma - (\mathcal{F}'[\boldsymbol{u}](\widetilde{\boldsymbol{u}}), \nabla\boldsymbol{\psi})_\Omega$$

$$= (\widetilde{\boldsymbol{u}}, (\boldsymbol{n} \cdot \mathcal{F}'[\boldsymbol{u}])^* \boldsymbol{\psi})_\Gamma - (\widetilde{\boldsymbol{u}}, (\mathcal{F}'[\boldsymbol{u}])^* \nabla\boldsymbol{\psi})_\Omega, \tag{15}$$

where the superscript * denotes the adjoint operation. Considering the right-hand side of Eq. (14), we can rewrite the continuous adjoint equations of Eq. (5) as follows:

$$-(\mathcal{F}'[\boldsymbol{u}])^* \nabla\boldsymbol{\psi} = i_\Omega'[\boldsymbol{u}] \quad \text{in} \quad \Omega, \tag{16}$$

$$(\boldsymbol{n} \cdot \mathcal{F}'[\boldsymbol{u}])^* \boldsymbol{\psi} = i_\Gamma'[\boldsymbol{u}] \quad \text{on} \quad \Gamma. \tag{17}$$

3.2 Error estimation

In engineering calculation, we hope to get the best possible approximation of the output functionals with the smallest computing resources. An error estimate based on the computed values is required to evaluate how accurate these output functionals are approximated. From prior experience, it is very hard to estimate the error. Using the adjoint problem, we can obtain the general result.

Theorem 3.1 *Let $\boldsymbol{u}$ be the analytical solution and $\boldsymbol{u}_h$ be the numerical solution of the primal equation* (5). *Let $\boldsymbol{\psi}$ denote the analytical solution of the adjoint equation* (16). *Then, the error between $I(\boldsymbol{u})$ and $I(\boldsymbol{u}_h)$ can be estimated by*

$$I(\boldsymbol{u}) - I(\boldsymbol{u}_h) = \sum_{k \in \mathcal{T}_h} \eta_k, \tag{18}$$

where

$$\eta_k = -\int_k \boldsymbol{\psi}^{\mathrm{T}}(\nabla \cdot \mathcal{F}(\boldsymbol{u}_h))\mathrm{d}\boldsymbol{x} + \int_{\partial k} \boldsymbol{\psi}^{\mathrm{T}}(\boldsymbol{n} \cdot \mathcal{F}(\boldsymbol{u}_h^+) - \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}))\mathrm{d}s. \tag{19}$$

Proof $I(\boldsymbol{u}) - I(\boldsymbol{u}_h)$

$$\overset{(8)}{=} \overline{I}(\boldsymbol{u}) - I(\boldsymbol{u}_h)$$

$$\overset{(12)}{=} \overline{I}(\boldsymbol{u}_h) - I(\boldsymbol{u}_h)$$

$$= -\mathcal{R}(\boldsymbol{u}_h, \boldsymbol{\psi})$$

$$\overset{(4)}{=} \sum_{k \in \mathcal{T}_h} \left(\int_k \nabla \boldsymbol{\psi}^{\mathrm{T}} \cdot \mathcal{F}(\boldsymbol{u}_h)\mathrm{d}\boldsymbol{x} - \int_{\partial k} \boldsymbol{\psi}^{\mathrm{T}} \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n})\mathrm{d}s \right)$$

$$= \sum_{k \in \mathcal{T}_h} \left(-\int_k \boldsymbol{\psi}^{\mathrm{T}}(\nabla \cdot \mathcal{F}(\boldsymbol{u}_h))\mathrm{d}\boldsymbol{x} + \int_{\partial k} \boldsymbol{\psi}^{\mathrm{T}}(\boldsymbol{n} \cdot \mathcal{F}(\boldsymbol{u}_h^+) - \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}))\mathrm{d}s \right)$$

$$= \sum_{k \in \mathcal{T}_h} \eta_k.$$

From this theorem, we can conclude that the primal residual can be related to the output functional error via the adjoint variable as a weight. Then, we take the absolute value of η_k to construct the local error indicator for adaptive algorithms, namely,

$$\eta_k^{\mathrm{ind}} = |\eta_k|. \tag{20}$$

4   Application on Euler equations

4.1   Some basic notations and output functionals

In this section, firstly, we define the flux Jacobian matrix as follows:

$$A_i = \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{u}}, \quad i = 1, 2.$$

Secondly, we define the boundary flux as follows:

$$\boldsymbol{f}_{\mathrm{b}} = \boldsymbol{n} \cdot \mathcal{F}(\boldsymbol{u}) = \boldsymbol{f}_1 n_1 + \boldsymbol{f}_2 n_2 = u_n(\rho \ \rho u \ \rho v \ \rho E)^{\mathrm{T}} + p(0 \ n_1 \ n_2 \ u_n)^{\mathrm{T}},$$

where $\boldsymbol{n} = (n_1, n_2)$ is the normal vector of the boundary, and u_n is the normal velocity, i.e., $u_n = un_1 + vn_2$. The pressure induced force coefficients, such as the drag coefficient, are the most important output functionals in inviscid compressible flows. Here, the drag coefficient is given by

$$I(\boldsymbol{u}) = \int_{\Gamma} i_{\Gamma}(\boldsymbol{u})\mathrm{d}s = \frac{1}{C_{\infty}} \int_{\Gamma_{\mathrm{W}}} p\boldsymbol{n} \cdot \boldsymbol{\varphi}\mathrm{d}s, \tag{21}$$

where $\Gamma = \partial\Omega$, and Γ_{W} is the solid wall of Γ. $C_{\infty} = \frac{1}{2}\rho_{\infty}|\boldsymbol{V}_{\infty}|^2\overline{l}$, where $\overline{l}$ is the reference length. $\boldsymbol{\varphi} = (\cos(\theta), \sin(\theta))^{\mathrm{T}}$, where θ denotes the angle of attack. The subscript ∞ means the far-field quantity.

4.2　Adjoint equation and boundary conditions

In this section, based on the drag coefficient, we introduce how to derive the continuous adjoint Euler equations and its boundary conditions. The approaches for other output functionals are similar. From Eq. (21), we note that the drag coefficient does not depend on the domain terms, i.e., $i_\Omega(\boldsymbol{u}) = 0$ in Eq. (7). Therefore, the adjoint equation (16) can be given by

$$-(A_1^{\mathrm{T}}, A_2^{\mathrm{T}}) \cdot \nabla \boldsymbol{\psi} = 0 \quad \text{in} \quad \Omega, \tag{22}$$

where $\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3, \psi_4)$ is the adjoint solution vector.

In inviscid compressible flows, on the solid wall Γ_{W}, the boundary condition imposes $u_n = 0$. Therefore, $\boldsymbol{n} \cdot \mathcal{F}(\boldsymbol{u}) = p(0, n_1, n_2, 0)^{\mathrm{T}}$ on Γ_{W}. Recalling the general functional of the boundary condition (17), the adjoint boundary condition on Γ_{W} under this output functional can be given by

$$p'[\boldsymbol{u}](0, n_1, n_2, 0)\boldsymbol{\psi} = \frac{1}{C_\infty} p'[\boldsymbol{u}]\boldsymbol{n} \cdot \boldsymbol{\varphi}, \tag{23}$$

i.e.,

$$\psi_2 n_1 + \psi_3 n_2 = \frac{1}{C_\infty}\boldsymbol{n} \cdot \boldsymbol{\varphi} \quad \text{on} \quad \Gamma_{\mathrm{W}}. \tag{24}$$

On a far-field boundary $\partial\Omega_\infty$, recall the definition of the drag coefficient (21). We notice that the value of this output functional does not depend on the far-field. Therefore, simply, we set

$$\boldsymbol{\psi} = \boldsymbol{0} \quad \text{on} \quad \partial\Omega_\infty. \tag{25}$$

Finally, we obtain the adjoint problem as follows:

$$\begin{cases} -(A_1^{\mathrm{T}}, A_2^{\mathrm{T}}) \cdot \nabla \boldsymbol{\psi} = 0 \quad \text{in} \quad \Omega, \\[2mm] \psi_2 n_1 + \psi_3 n_2 = \dfrac{1}{C_\infty}\boldsymbol{n} \cdot \boldsymbol{\varphi} \quad \text{on} \quad \Gamma_{\mathrm{W}}, \\[2mm] \boldsymbol{\psi} = \boldsymbol{0} \quad \text{on} \quad \partial\Omega_\infty. \end{cases} \tag{26}$$

5　Numerical implementation

5.1　Solving primal equations

Firstly, we add a pseudo time term to Eq. (1), i.e., we rewrite the steady-state Euler equations as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = 0. \tag{27}$$

The spatial discretization by the DG method leads to a system of ordinary differential equations as follows:

$$\mathcal{M}\frac{\partial \boldsymbol{u}_h}{\partial t} = -R(\boldsymbol{u}_h), \tag{28}$$

where $\mathcal{M}$ is the mass matrix, and R is the right-hand residual. In this work, we are interested in stationary problems. Therefore, the backward Euler scheme is used in the temporal discretization. After temporal discretization and linearizing Eq. (28) in time, we can get a system of linear equations as follows:

$$\Big(\frac{\mathcal{M}}{\Delta t} + \frac{\partial R}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{u}_h^n}\Big)\Delta \boldsymbol{u}_h^n = -R(\boldsymbol{u}_h^n), \tag{29}$$

where Δt is the time increment, and

$$\Delta \boldsymbol{u}_h^n = \boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n.$$

Then, the linear system can be solved by the generalized minimal residual (GMRES) method with ILU0 preconditioning.

5.2   Solving adjoint equations

In Eq. (19), the error indicator involves the analytical adjoint solution $\boldsymbol{\psi}$ which is unknown. Therefore, we need to make an approximation to it. Let $\boldsymbol{\psi}_h$ denote the approximation. We note that $\boldsymbol{\psi}_h$ must not be approximated by the same finite element space $\boldsymbol{V}_{h,p}$. Otherwise, the error indicator will be identically equal to zero due to the Galerkin orthogonality. There are several approaches to compute the numerical approximation $\boldsymbol{\psi}_h{}^{[14-15]}$. In this work, the adjoint equations are approximated on the same mesh employed for the primal equations, but the order of the finite element space is increased from p to $p+1$.

The numerical implementation of the adjoint equations is a little different from the primal equations. From Eq. (26), we notice that $(A_1^{\mathrm{T}}, A_2^{\mathrm{T}})$ is independent of the adjoint variable $\boldsymbol{\psi}$. Therefore, the adjoint equations are linear. Therefore, we can solve this problem by iteration or the direct linear system solver directly after spatial discretization.

We should remark that there are two main differences between the DG discretization of the primal problem and the DG discretization of the adjoint problem. The first one is the difference between the numerical fluxes. Here, we write Eq. (1) in the following quasi-linear form:

$$\frac{\partial \boldsymbol{f_1}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial x} + \frac{\partial \boldsymbol{f_2}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial y} = 0 \quad \text{in} \quad \Omega,$$

$$\rightarrow \; -(A_1^{\mathsf{T}}, A_2^{\mathsf{T}})\cdot\nabla\psi = 0$$

i.e., $(A_1, A_2)\cdot\nabla\boldsymbol{u} = 0$ in Ω. We note that Eq. (22) and the above equation have the same eigenvalues. Therefore, the local Lax-Friedrichs numerical flux for the adjoint equation can be given by

$$\mathcal{H}_{\mathrm{adj}}(\boldsymbol{\psi}_h^+, \boldsymbol{\psi}_h^-, \boldsymbol{n}) = \frac{1}{2}\big(-(A_1^{\mathrm{T}}, A_2^{\mathrm{T}})\cdot\boldsymbol{n}\boldsymbol{\psi}_h^+ - (A_1^{\mathrm{T}}, A_2^{\mathrm{T}})\cdot\boldsymbol{n}\boldsymbol{\psi}_h^- + \alpha(\boldsymbol{\psi}_h^+ - \boldsymbol{\psi}_h^-)\big), \qquad (30)$$

where α is the largest eigenvalue which can be calculated from the primal problem directly. The second difference lies in the boundary conditions. As we have discussed in Section 4 on the far-field boundary $\partial\Omega_\infty$, we simply set $\boldsymbol{\psi}_h^- = 0$. On the solid wall boundary, there are a set of boundary values of $\boldsymbol{\psi}_h$ satisfying the boundary conditions in Eq. (26). Here, we enforce the solid wall boundary condition as follows:

$$\psi_1^- = \psi_1^+, \tag{31}$$

$$\psi_2^- = \psi_2^+ + 2n_1\Big(\frac{1}{C_\infty}\boldsymbol{n}\cdot\boldsymbol{\varphi} - n_1\psi_2^+ - n_2\psi_3^+\Big), \tag{32}$$

$$\psi_3^- = \psi_3^+ + 2n_2\Big(\frac{1}{C_\infty}\boldsymbol{n}\cdot\boldsymbol{\varphi} - n_1\psi_2^+ - n_2\psi_3^+\Big), \tag{33}$$

$$\psi_4^- = \psi_4^+, \tag{34}$$

which guarantees the average value

$$\overline{\boldsymbol{\psi}}_h = \frac{1}{2}(\boldsymbol{\psi}_h^+ + \boldsymbol{\psi}_h^-)$$

satisfying the adjoint solid wall boundary condition, i.e., $\overline{\psi}_2 n_1 + \overline{\psi}_3 n_2 = \frac{1}{C_\infty}\boldsymbol{n}\cdot\boldsymbol{\varphi}$.

5.3   Adjoint-based adaptive mesh refinement algorithm

Utilize the local error indicator (20) by an adaptive mesh refinement algorithm shown in Fig. 1. In each refinement cycle, the cells with a large local error indicator η_k^{ind} will be refined, and the cells with small η_k^{ind} will be coarsened. Following the process, we will get a mesh on which the local error indicator η_k^{ind} is equidistribution. For more details of the mesh adaptation strategies, we refer readers to Ref. [8].
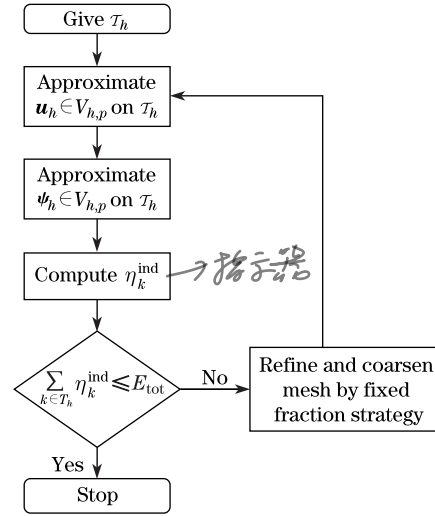
Fig. 1 Adjoint-based adaptive mesh refinement algorithm

6 Numerical results

We show some numerical tests to discuss the advantages of our adaptive mesh refinement algorithm, comparing with the traditional algorithms based on the feature-based or residual-based error indicators. Here, firstly, let us introduce the feature-based error indicator $\eta_k^{\text{feature}} = \lg(1 + |\nabla\rho_k|)$ and the residual-based error indicator[18]

$$\eta_k^{\text{residual}} = \|hR(\boldsymbol{u}_h)\|_{L_2(k)} + \|h^{\frac{1}{2}}r(\boldsymbol{u}_h)\|_{L_2(\partial k)},$$

where $\nabla\rho_k$ is the gradient of the density, and

$$R(\boldsymbol{u}_h)\big|_k = -\nabla\cdot F(\boldsymbol{u}_h), \quad r(u_h)\big|_k = F(\boldsymbol{u}_h)\cdot\boldsymbol{n} - \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}), \quad k \in \mathcal{T}_h.$$

In the following numerical tests, the primal problem will be approximated in $\mathcal{V}_{h,1}$, and the adjoint problem will be approximated in $\mathcal{V}_{h,2}$.

6.1 NACA-0012 airfoil

In the first numerical experiment, we consider an NACA-0012 airfoil in an inviscid compressible flow with $Ma_\infty = 0.4$ and $\theta = 0$. The output functional considered here is the drag coefficient. Therefore, the theoretical value of the output functional is zero.

Figure 2 shows the initial mesh of this test. The meshes obtained by use of both the adjoint error indicator and the feature-based error indicator are shown in Fig. 3. From these figures, we can see that most of the refinement is enforced near the leading and trailing edges of the airfoil. Furthermore, we find that the mesh obtained by our adjoint-based error indicator is also refined around the surfaces of the airfoil.

Figure 4 shows the output functional error convergence under different refinement approaches. From these results, we can see that the error of the output functional estimated on the mesh obtained by use of the adjoint-based error indicator is the smallest for a given number of cells. In addition, we can observe that, at a given error, the number of the mesh cells needed by the adjoint-based adaptive approach is an order of magnitude less than the uniform refinement approach.

Figure 5 shows the distribution of the adjoint-based error indicator during the refinement cycle. We find that the errors are larger around the surfaces of the airfoil on the initial mesh.
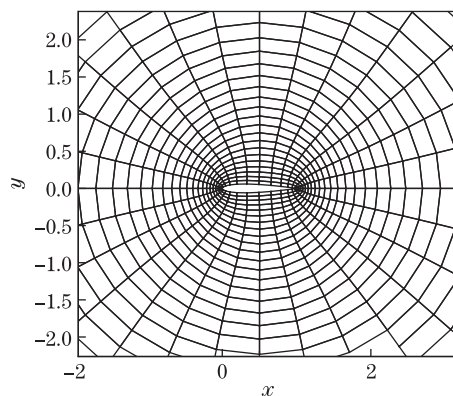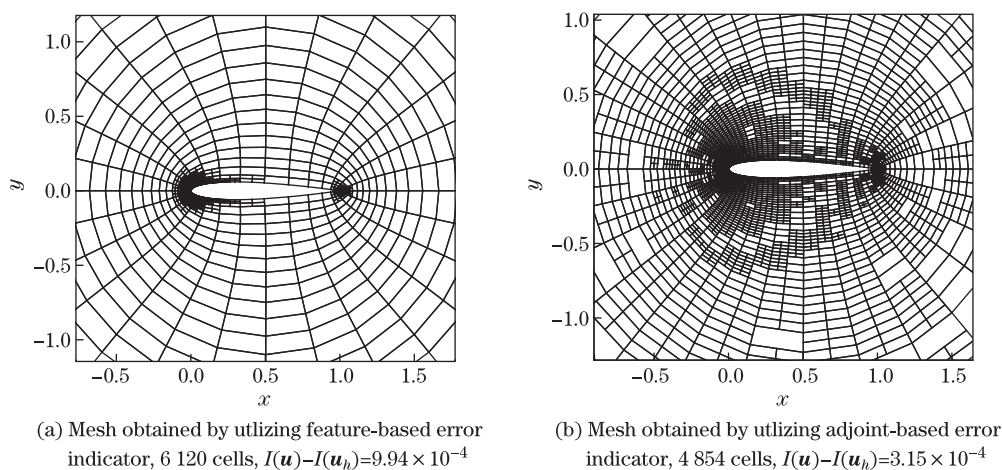
Fig. 2   Initial mesh, 924 cells

(a) Mesh obtained by utlizing feature-based error
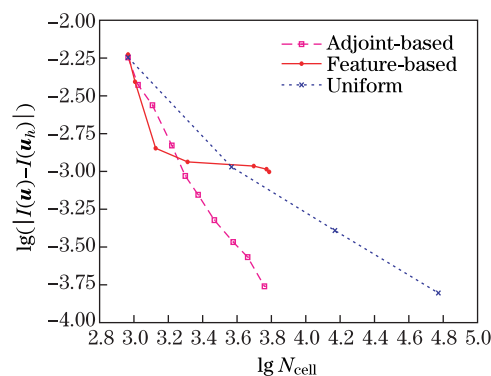indicator, 6 120 cells, $I(\boldsymbol{u})-I(\boldsymbol{u}_h)=9.94\times10^{-4}$

(b) Mesh obtained by utlizing adjoint-based error
indicator, 4 854 cells, $I(\boldsymbol{u})-I(\boldsymbol{u}_h)=3.15\times10^{-4}$

Fig. 3   Meshes obtained by utilizing feature-based and adjoint-based error indicators

Fig. 4   Convergence of drag coefficient utilizing different error indicators

The cells with larger errors are refined, which explains why the cells near the airfoil surfaces are refined. Also, we notice that with the repetition of the mesh refinement, the errors in all cells are decreasing. Consequently, using our adaptive algorithm, we can get a mesh, on which the error is equally distributed.

(a) Error distribution before 1st refinement

(b) Error distribution before 4th refinement

(c) Error distribution before 6th refinement

(d) Error distribution before 9th refinement
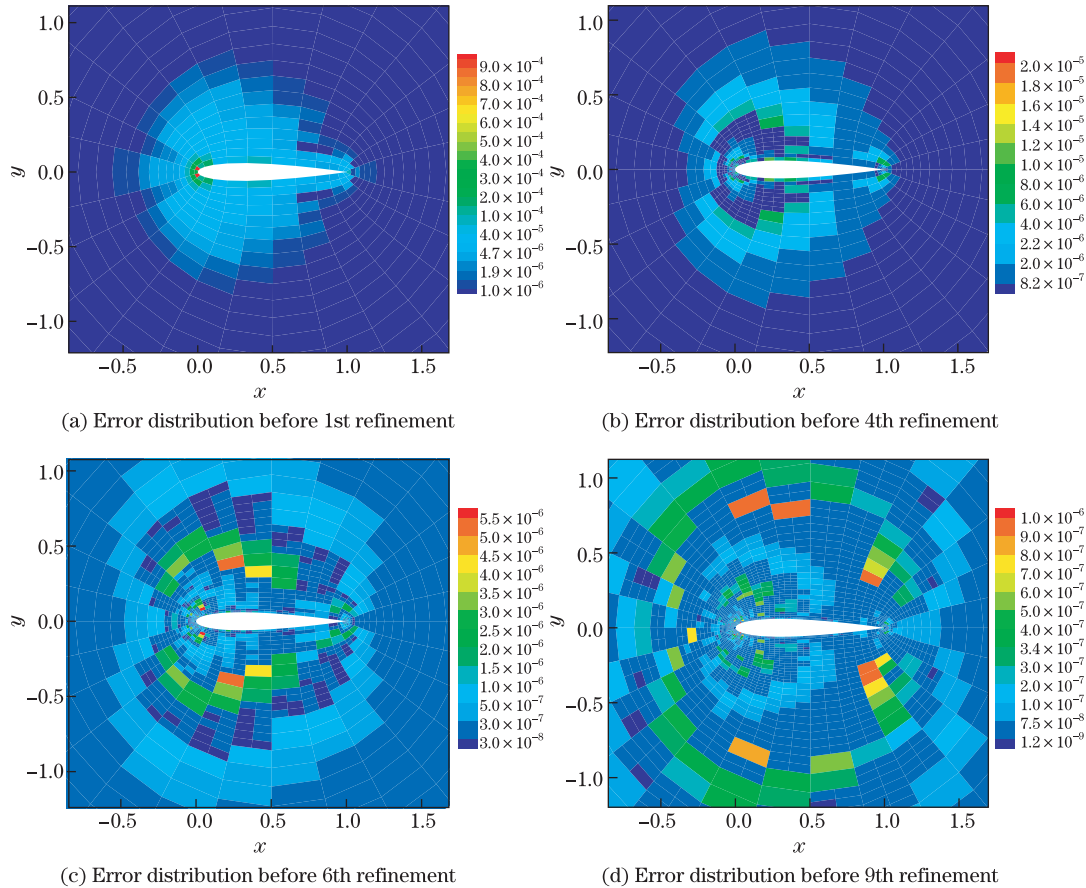
Fig. 5 Distributions of η_k^{ind}

6.2 Smooth bump

In the second numerical experiment, we consider the inviscid flow over a smooth bump. The computational domain is given by $\Omega = \{(x,y) \in [-1.5, 1.5] \times [0, 0.8], y > 0.062\,5\,\mathrm{e}^{-25x^2}\}$. At the inlet, $Ma_\infty = 0.5$, and $\theta = 0$. There should be no entropy production because the flow is smooth enough. Therefore, we construct the following output functional:

$$I(\boldsymbol{u}) = \int_\Omega \Big(\frac{p/\rho^\gamma - p_\infty/\rho_\infty^\gamma}{p_\infty/\rho_\infty^\gamma} \Big)^2 \mathrm{d}x,$$

where p is the pressure, ρ is the density, and the subscript ∞ means the freestream value. At the left- and right-sides, we apply the subsonic inflow boundary condition and the subsonic outflow boundary condition, respectively. At the top and the bottom, the slip wall conditions are applied. The theoretical value of this output functional is zero.

Figure 6 shows the initial mesh for this problem, and Fig. 7 shows the pressure contour of the primal Euler equations. We can see that the solution is axisymmetric. From Fig. 8, we can see that most of the refinements are near the bump. Both of the two adaptive approaches construct almost the same meshes, except that the refinement is denser around the bump in the adjoint-based approach.

In Fig. 9, we show the convergence of the output functional error. Some observation can be made. Firstly, both the adjoint-based and residual-based adaptive approaches show better convergence than the uniform refinement. Indeed, after several refinements, the error computed
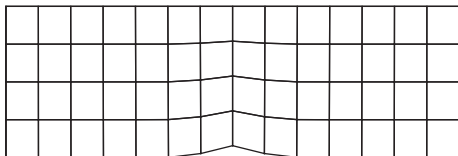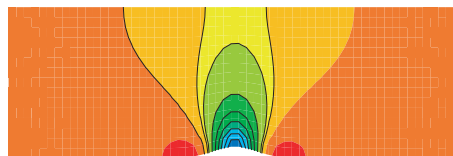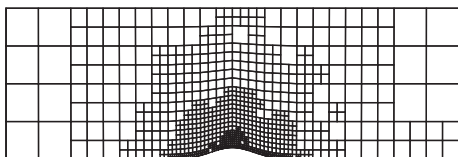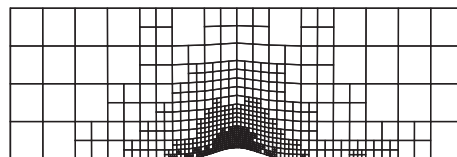
Fig. 6   Initial mesh, 56 cells

Fig. 7   Pressure contour of primal Euler equations

(a) Mesh obtained by utilizing residual-based error indicator, 872 cells, $I(\boldsymbol{u})-I(\boldsymbol{u}_h)$=7.9 × 10^{-8}

(b) Mesh obtained by utilizing adjoint-based error indicator, 773 cells, $I(\boldsymbol{u})-I(\boldsymbol{u}_h)$=8.6 × 10^{-8}

Fig. 8   Meshes obtained by utilizing residual-based and adjoint-based error indicators

on the subsequent adaptive mesh is at least two orders of magnitude smaller comparing with the error computed on the uniform refinement mesh. Secondly, the adjoint-based refinement gives a better accuracy than the residual-based refinement for a given number of cells.
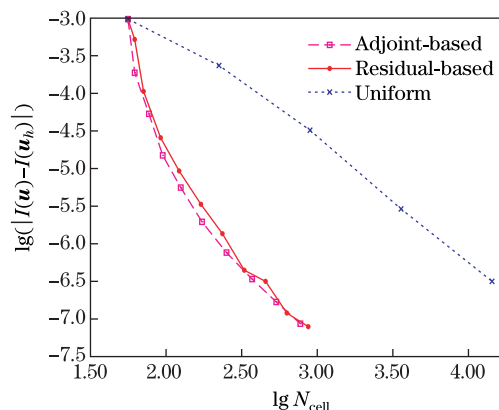

Fig. 9   Convergence of output error utilizing different error indicators

7   Conclusions

We have introduced an adjoint-based adaptive mesh refinement algorithm for inviscid compressible flows. Firstly, we show the derivation of a continuous adjoint problem for the Euler equations. Secondly, we introduce the error estimation for the given output functionals. Thirdly, we discuss the numerical implementation of the DG method for an adjoint problem. Then, we develop an adaptive mesh refinement algorithm based on the continuous adjoint approach. The numerical tests show that the meshes constructed by this adaptive algorithm are much more economical for computing the values of the output functionals than the meshes constructed by the traditional error indicators. We conclude that the adaptive approach presented in this paper is very promising in improving the accuracy of the output functionals within given computational resources.

References

[1] Shaydurov, V., Liu, T., and Zheng, Z. Four-stage computational technology with adaptive numerical methods for computational aerodynamics. *AIP Conference Proceedings*, **1487**, 42–48 (2012)

[2] Xu, Z., Marie-Gabrielle, V., Julien, D., Paul, L., Dominique, P., Jean-Yves, T., Ricardo, C., Jason, L., Luis, M., and David, Z. Mesh adaptation using different error indicators for the Euler equations. 15*th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Anaheim (2001) DOI 10.2514/6.2001-2549

[3] Venditti, D. A. and Darmofal, D. L. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, **187**, 22–46 (2003)

[4] Becker, R. and Rannacher, R. Feed-back approach to error control in finite element methods: basic analysis and examples. *East-West Journal of Numerical Mathematics*, **4**, 237–264 (1996)

[5] Becker, R. and Rannacher, R. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, **10**, 1–102 (2001)

[6] Venditti, D. A. *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph. D. dissertation, Massachusetts Institute of Technology, Massachusetts (2002)

[7] Venditti, D. A. and Darmofal, D. L. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *Journal of Computational Physics*, **176**, 40–69 (2002)

[8] Hartmann, R. and Houston, P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, **183**, 508–532 (2002)

[9] Hartmann, R. and Houston, P. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, **24**, 979–1004 (2003)

[10] Hartmann, R. Error estimation and adjoint based refinement for an adjoint consistent DG discretization of the compressible Euler equations. *International Journal of Computing Science and Mathematics*, **1**, 207–220 (2007)

[11] Wang, L. and Mavriplis, D. J. Adjoint-based *h-p* adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics*, **228**, 7643–7661 (2009)

[12] Fidkowski, K. J. *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations*, Ph. D. dissertation, Massachusettes Institute of Technology, Massachusettes (2007)

[13] Fidkowski, K. J. and Darmofal, D. L. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, **225**, 1653–1672 (2007)

[14] Fidkowski, K. J. and Darmofal, D. L. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, **49**, 673–694 (2011)

[15] Hartmann, R. *Adaptive Finite Element Methods for the Compressible Euler Equations*, Ph. D. dissertation, University of Heidelberg, Heidelberg (2002)

[16] Fidkowski, K. J. and Roe, P. L. An entropy adjoint approach to mesh refinement. *SIAM Journal on Scientific Computing*, **32**, 1261–1287 (2010)

[17] Li, L. Y., Allaneau, Y., and Jamesony, A. Continuous adjoint approach for adaptive mesh refinement. 20*th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Honolulu, Hawaii (2011)

[18] Houston, P. and Süli, E. *hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, **23**, 1226–1252 (2001)

[19] Bangerth, W. *Adaptive Finite Element Methods for Differential Equations Lectures in Mathematics*, Birkhäuser, Basel (2003)