Advances in Aerodynamics

RESEARCH Open Access

Adaptive local discontinuous Galerkin methods with semi-implicit time discretizations for the Navier-Stokes equations

Xiangyi Meng and Yan Xu*

*Correspondence: yxu@ustc.edu.cn
School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, Anhui, People's Republic of China

Abstract

In this paper, we present a mesh adaptation algorithm for the unsteady compressible Navier-Stokes equations under the framework of local discontinuous Galerkin methods coupled with implicit-explicit Runge-Kutta or spectral deferred correction time discretization methods. In both of the two high order semi-implicit time integration methods, the convective flux is treated explicitly and the viscous and heat fluxes are treated implicitly. The remarkable benefits of such semi-implicit temporal discretizations are that they can not only overcome the stringent time step restriction compared with time explicit methods, but also avoid the construction of the large Jacobian matrix as is done for fully implicit methods, thus are relatively easy to implement. To save computing time as well as capture the flow structures of interest accurately, a local mesh refinement (h-adaptive) technique, in which we present detailed criteria for selecting candidate elements and complete strategies to refine and coarsen them, is also applied for the Navier-Stokes equations. Numerical experiments are provided to illustrate the high order accuracy, efficiency and capabilities of the semi-implicit schemes in combination with adaptive local discontinuous Galerkin methods for the Navier-Stokes equations.

Keywords: Mesh adaptation, Local discontinuous Galerkin methods, Implicit-explicit Runge-Kutta methods, Spectral deferred correction methods, Navier-Stokes equations

1 Introduction

In this paper, we focus on an h-adaptive local discontinuous Galerkin (LDG) method in combination with implicit-explicit Runge-Kutta (IMEX-RK) or spectral deferred correction (SDC) time discretizations to solve the unsteady compressible Navier-Stokes (NS) equations.

The discontinuous Galerkin (DG) method, which is a class of finite element methods using a completely discontinuous piecewise polynomial space for the numerical solutions and the test functions in spatial variables, has been widely used in computational fluid dynamics, computational acoustics and computational magneto-hydrodynamics, since it was firstly introduced in 1973 by Reed and Hill [1] in the framework of linear neutron transport and then further extended by Cockburn et al. [2–5] for solving nonlinear hyperbolic conservation laws through a combination of explicit, Strong-Stability-Preserving Runge-Kutta (SSP-RK) time discretizations and total variation bounded nonlinear limiters to achieve non-oscillatory properties for strong shocks. The DG methods combine two advantageous features commonly associated to continuous finite element and finite volume methods, such as high order accuracy, flexibility on complex geometry, flexibility for *h-p* adaptivity, and so on, and are indeed a natural consideration when solving hyperbolic conservation laws, such as the compressible Euler equations. However, when it comes to partial differential equations (PDEs) containing high order spatial derivatives, e.g., the compressible NS equations where the viscous and heat fluxes exist, a severe difficulty with the approximations of the numerical fluxes for solution derivatives arises by the direct application of the DG methods, and a naive arithmetic mean of the solution derivatives without taking into account the possible jump of the solutions will yield a weakly unstable scheme.

In order to properly resolve the solution derivatives in the NS equations at the interfaces, plenty of numerical methods have been proposed in the literature. In 1997, Bassi and Rebay [6] firstly attempted to apply the DG methods to the compressible NS equations, later on, they introduced an improved method [7] based on the previous scheme in order to maintain the compactness and stability for the pure diffusion problems. Hartmann and Houston [8] proposed employing the symmetric interior penalty method, which can guarantee optimal order of convergence in terms of L^2 norm of the error, for the discretization of the leading order terms of the compressible NS equations. To deal with moving and deforming boundaries, Klaij and van der Vegt [9] presented a space-time DG method for the compressible NS equations. The key feature of the space-time DG method is that no distinction is made between space and time variables and thus this provides optimal flexibility to deal with time dependent boundaries and deforming elements. Based on a so-called inter-cell reconstruction, Luo et al. [10] developed a reconstruction-based DG method for the NS equations on arbitrary grids. The numerical viscous and heat fluxes at interfaces are obtained by locally reconstructing a smooth solution with a least-square method from the underlying discontinuous solution. Based on a direct weak formulation for the parabolic equations, a so-called direct DG method for solving second-order diffusion problems was originally introduced by Liu and Yan [11], and further extended by Cheng et al. [12] to solve the compressible NS equations.

Inspired and attracted by the high order accuracy, easy extension to high order PDEs and arbitrary grids of the LDG method, we select it as the spatial discretization in our study. Besides, adopting *h*-adaptive technique and semi-implicit time marching methods are another two aspects we focus on in this paper to improve the computational accuracy and efficiency as well as deduce the computational complexity. The LDG method was initially developed to solve nonlinear convection-diffusion equations by Cockburn and Shu [13], enlightened by Bassi and Rebay's successful simulation for the compressible NS equations. The idea behind the LDG method is to properly rewrite the PDEs containing high order derivatives as a first-order system by introducing auxiliary variables, then apply the DG method to the system, in which correctly designing the numerical fluxes is the key ingredient to guarantee stability and local solvability of the auxiliary variables. There has been abundant literature on designing and analyzing the LDG schemes for different types of high order PDEs, for a detailed review, see [14] and the references therein.

By sharing the advantages of the DG method, the LDG scheme requires no imposition of solution continuity between adjacent elements and allows the appearance of hanging nodes, thus is extremely flexible to do local grid refinement. To start the mesh adaptation, a criterion to determine whether an element in the computational mesh needs to be refined or coarsened is demanded. Generally, the criteria can be divided into two types, i.e., error estimators and heuristic indicators [15]. Error estimators are based on a so-called a posteriori error estimation which seeks to bound the error with respect to a given norm. As a comparison, heuristic indicators usually depend on the local gradient of thermodynamical variables such as the density, pressure, entropy and so on, or the local divergence or curl of the velocity field, thus are relatively simple to be implemented for complicated PDEs. In the current, a rigorous mathematical analysis to develop the a posteriori error estimation for the LDG scheme of the compressible NS equations is out of our scope, therefore we consult the criteria in [16, 17] for the compressible Euler equations and choose a combination of different heuristic indicators. In practical implementation, the candidate elements marked for refinement are indeed to be refined, however, to retain the mesh quality, some other elements around these elements may also require to be refined. In contrast, to achieve the same goal, the candidate elements marked for coarsening may not be coarsened depending on their neighbors. What's more, to maintain conservation and accuracy of the numerical solutions during refinement and coarsening, the L^2 projection is implemented for the prolongation and restriction of grid.

The application of any above extension of the DG methods to the compressible NS equations will generate a large coupled system of ordinary differential equations (ODEs), which require accurate and efficient time integration methods to march in time. The explicit, high order SSP-RK time discretizations, which are suitable choices for hyperbolic conservation laws, will sustain extremely small time step restriction for stability, but not for accuracy due to the appearance of stiffness terms of the NS equations, i.e., the viscous and heat fluxes terms. Nearly all the papers referred above applied the implicit temporal discretizations, e.g., the backward Euler method, to overcome this restriction. This kind of remedy is efficient to some extent, since the Courant-Friedrichs-Lewy (CFL) number can be adjusted to be quite large, but the construction of the large Jacobian matrix and the choice of efficient preconditioners for the linear systems arising in the inner loop greatly increase the difficulty in implementation. In addition, for unsteady flow problems with complicated flow structures, a time step of at least the same order of magnitude as the mesh size is required to capture the complex flow structures accurately at any time. By turning to the semi-implicit time discretization methods, specifically, IMEX-RK and SDC methods, and treating the convective, viscous and heat fluxes differently, no Jacobian matrix is required to be assembled and only two linear systems for the momentum and energy equations, respectively, need to be solved. Numerical experiments show that the residual of the GMRES solver even without any preconditioner can approach the machine error within a dozen of, sometimes two or three, outer iterations if the mesh quality is not too bad. L. Pareschi and G. Russo considered IMEX-RK methods in time for hyperbolic systems of conservation laws with stiff relaxation terms in [18]. In [19, 20] Wang et al. analyzed the stability and error estimates of the LDG methods coupled with IMEX-RK time discretizations for the linear convection-diffusion equations in one and two dimensions, and unconditional stability and optimal error estimates are obtained. Xia et al. [21] explored three different time discretization techniques including SDC methods for solving the stiff ODEs resulting from an LDG spatial discretization to PDEs containing high order spatial derivatives, and verified that all the three methods are efficient and a time step $\triangle t = O(\triangle x)$ is allowed for k-th order PDEs. The LDG method is more flexible for treating the nonlinear terms when introducing the auxiliary for high order derivative terms. Due to the local properties of the LDG methods, the resulting implicit scheme is easy to implement and can be solved in an explicit way when it is coupled with iterative methods, which has been demonstrated in the references [22, 23]. For other work on semi-implicit schemes for gasdynamics, we suggest the readers consulting [24–30] and the references therein.

The outline of this paper is organized as follows. Governing equations are given in Section 2. In Section 3, we present the LDG scheme for the compressible NS equations, including the detailed and clear treatment for the numerical fluxes for subsonic inflow/outflow, supersonic inflow/outflow and solid wall boundary conditions. Section 4 is devoted to the discussion about two kinds of semi-implicit time discretization methods we adopt in this paper, i.e., IMEX-RK and SDC methods, and the fully discrete schemes of the NS equations. In Section 5 we provide the implementation details of the h-adaptive technique. Section 6 contains numerical results for the unsteady compressible NS equations, which demonstrate the high order accuracy and capabilities of the presented methods. Finally we give conclusions in Section 7.

2 Governing equations

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with $d \leq 3$, the NS equations governing the dynamics of viscous compressible flows express the conservation of mass, momentum and energy, and can be written in dimensionless and conservative form [31]

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \nabla \cdot \mathbf{G}(\mathbf{U}, \nabla \mathbf{U}), \tag{1}$$

in $\Omega \times (0, T]$, $T > 0$, with $\mathbf{U} \in \mathbb{R}^{d+2}$ the vector of conserved variables and $\mathbf{F}, \mathbf{G} \in \mathbb{R}^{(d+2)\times d}$ the inviscid and viscous fluxes, which are defined as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\mathbf{I} \\ (E+p)\mathbf{u} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \\ \boldsymbol{\tau}\mathbf{u} - \mathbf{q} \end{bmatrix},$$

respectively. Here ρ, $\mathbf{u} \in \mathbb{R}^d$ and p denote the mass density, velocity field and pressure, respectively. $\mathbf{I}$ represents unit tensor. The total energy per unit volume E is the sum of internal and kinetic energy

$$E = \rho\left(e + \frac{1}{2}|\mathbf{u}|^2\right),$$

with e the specific internal energy. In order to close the system, we consider the equation of state for a calorically perfect gas

$$p = (\gamma - 1)\rho e,$$

where $\gamma = \frac{c_p}{c_v}$ is the ratio of c_p and c_v, the specific heats at constant pressure and volume, respectively. By Newtonian approximation and Stokes hypothesis, the viscous stress tensor $\boldsymbol{\tau} \in \mathbb{R}^{d\times d}$ relating to the derivatives of the velocity field $\mathbf{u}$ is defined as

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

where the dynamic viscosity coefficient μ, which is determined through Sutherland's law, reads in dimensionless form

$$\mu = \frac{1}{Re_\infty} \frac{1 + T_s/T_\infty}{T + T_s/T_\infty} T^{\frac{3}{2}},$$

with Re_∞ the Reynolds number, T the temperature, T_s a constant and subscript ∞ denoting the uniform free-stream values. The heat flux vector $\mathbf{q} \in \mathbb{R}^d$ caused by the gradient of temperature T is defined as

$$\mathbf{q} = -\kappa \nabla T,$$

via Fourier's heat conduction law, where $\kappa = \frac{c_p \mu}{Pr}$ is the thermal conductivity coefficient, with Pr the Prandtl number. In addition, the specific internal energy e can relate to the temperature T by

$$e = c_v T.$$

As a matter of fact, the specific heats c_p and c_v in dimensionless form can be defined as

$$c_p = \frac{1}{(\gamma - 1) M_\infty^2}, \quad c_v = \frac{1}{\gamma (\gamma - 1) M_\infty^2},$$

with Mach number M_∞ for uniform flow given by

$$M_\infty = \frac{u_\infty}{a_\infty}, \quad a_\infty = \sqrt{\frac{\gamma p_\infty}{\rho_\infty}}.$$

In this paper, unless otherwise stated, we will use $\gamma = 1.4$, $Pr = 0.72$ for air, and μ as a constant for simplicity, i.e., $\mu = \frac{1}{Re_\infty}$.

3 The LDG method

In this section, we will introduce the LDG method for the spatial discretization of the compressible NS equations. Special attention will be paid to the numerical fluxes on various boundaries.

3.1 Notations and finite element spaces

For the DG spatial discretizations, the domain Ω is approximated with a tessellation $\mathcal{T}_h$ consisting of nonoverlapping shape-regular elements K, which satisfy the condition $\bigcup_{K \in \mathcal{T}_h} \overline{K} := \overline{\Omega}_h \to \overline{\Omega}$ as $h \to 0$, with the mesh size $h := \max_{K \in \mathcal{T}_h} h_K$ and the diameter h_K of element K. Let Γ denote the union of the boundaries of all the elements K and classify them into internal faces Γ_I and boundary faces Γ_B, i.e., $\Gamma = \Gamma_I \cup \Gamma_B$. We also assume that Γ_B may be decomposed as follows

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

where $\Gamma_{\text{sub-in}}$, $\Gamma_{\text{sub-out}}$, $\Gamma_{\text{sup-in}}$, $\Gamma_{\text{sup-out}}$ and Γ_W are distinct subsets of Γ_B representing subsonic-inflow, subsonic-outflow, supersonic-inflow, supersonic-outflow and solid wall boundaries, respectively. For solid wall boundaries, we also distinguish them either according to slip (reflective) and no-slip conditions, i.e.,

$$\Gamma_W = \Gamma_{W,\text{slip}} \cup \Gamma_{W,\text{no-slip}},$$

or according to isothermal and adiabatic conditions, i.e.,

$$\Gamma_W = \Gamma_{W,\text{iso}} \cup \Gamma_{W,\text{adia}}.$$

Let $e \in \Gamma_I$ be an internal face shared by the "left" and "right" elements K_L and K_R, i.e., $e = \partial K_L \cap \partial K_R$, where the so-called "left" and "right" can be uniquely defined for each internal face according to any fixed rule. Suppose ϕ is a function on K_L and K_R, but possibly discontinuous across e, let ϕ^L and ϕ^R denote $(\phi|_{K_L})|_e$ and $(\phi|_{K_R})|_e$, the left and right traces, respectively. Similar definitions can be obtained component-wisely for vector-valued and matrix-valued functions.

For the LDG discretizations, we require to introduce the finite element spaces. Actually each element K of the tessellation $\mathcal{T}_h$ is connected to a reference element $\hat{K}$ through some mapping F_K. The mapping $F_K : \hat{K} \to K$ from the reference element $\hat{K}$ to the real physical element K is a function defined in the space of the reference element for each independent variable. For example, for a quadrilateral element K in two dimension, $\hat{K} =[-1, 1]^2$ is the unit square and F_K is expressed in terms of the nodal shape functions N_i, $i = 1, \cdots, 4$ by

$$\mathbf{x} = F_K(\boldsymbol{\xi}) = \sum_{i=1}^{4} \mathbf{x}_K^i N_i(\boldsymbol{\xi}),$$

where $\mathbf{x} = (x_1, x_2) \in K$, $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \hat{K}$, $\mathbf{x}_K^i$, $i = 1, \cdots, 4$ are the four vertex coordinates of K and

$$\begin{aligned} N_1(\boldsymbol{\xi}) &:= \frac{1}{4}(1-\xi_1)(1-\xi_2), \quad N_2(\boldsymbol{\xi}) := \frac{1}{4}(1+\xi_1)(1-\xi_2), \\ N_3(\boldsymbol{\xi}) &:= \frac{1}{4}(1+\xi_1)(1+\xi_2), \quad N_4(\boldsymbol{\xi}) := \frac{1}{4}(1-\xi_1)(1+\xi_2), \end{aligned}$$

see schematic map in Fig. 1. Then the finite element spaces associated with the tessellation $\mathcal{T}_h$ are given by

$$\begin{aligned} V_h &= \left\{ \mathbf{V} = (v_1, \cdots, v_{d+2})^T \in (L^2(\Omega))^{d+2} : v_i|_K \circ F_K \in P^k(\hat{K}),\ i = 1, \cdots, d+2,\ \forall K \in \mathcal{T}_h \right\}, \\ W_h &= \left\{ \mathbf{W} = (w_1, \cdots, w_d)^T \in (L^2(\Omega))^{d} : w_i|_K \circ F_K \in P^k(\hat{K}),\ i = 1, \cdots, d,\ \forall K \in \mathcal{T}_h \right\}, \\ \Sigma_h &= \left\{ \boldsymbol{\sigma} = (\sigma_{ij})_{1\le i,j\le d} \in (L^2(\Omega))^{d\times d} : \sigma_{ij}|_K \circ F_K \in P^k(\hat{K}),\ i,j = 1, \cdots, d,\ \forall K \in \mathcal{T}_h \right\}, \end{aligned}$$

with $P^k(\hat{K})$ the space of polynomials of degree up to k with respect to d variables in the reference element $\hat{K}$. Note that the functions in V_h, W_h and Σ_h are allowed to be completely discontinuous across element interfaces.

Further, we also define the inner product notations in element and on face as

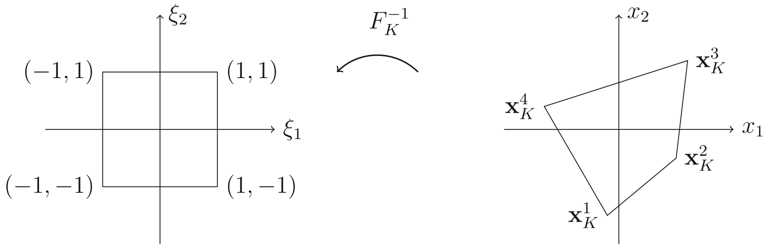


Fig. 1 Schematic map for the transformation F_K from a unit square in the reference space to the real element in the physical space

$$(\phi,\varphi)_K := \int_K \phi\varphi dK, \quad (\phi,\varphi)_{\partial K} := \int_{\partial K} \phi\varphi ds,$$
$$(\mathbf{u},\mathbf{v})_K := \int_K \mathbf{u}\cdot\mathbf{v} dK, \quad (\mathbf{u},\mathbf{v})_{\partial K} := \int_{\partial K} \mathbf{u}\cdot\mathbf{v} ds,$$
$$(\boldsymbol{\sigma},\boldsymbol{\eta})_K := \int_K \boldsymbol{\sigma} : \boldsymbol{\eta} dK, \quad (\boldsymbol{\sigma},\boldsymbol{\eta})_{\partial K} := \int_{\partial K} \boldsymbol{\sigma} : \boldsymbol{\eta} ds,$$

for the scalar-valued functions ϕ, φ, vector-valued functions $\mathbf{u}, \mathbf{v}$ and matrix-valued functions $\boldsymbol{\sigma}, \boldsymbol{\eta}$, respectively, where

$$\boldsymbol{\sigma} : \boldsymbol{\eta} := \sigma_{ij}\eta_{i,j}.$$

3.2 The LDG discretization

In order to propose the LDG discretization for the NS equations, we firstly rewrite (1) as a first order system, which is composed of the primary equations

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{U}) - \mathbf{G}(\mathbf{U},\boldsymbol{\tau},\mathbf{q})) = 0, \tag{2}$$

and the auxiliary equations

$$\mathbf{z} = \nabla\mathbf{u}, \tag{3a}$$
$$\mathbf{q} = -\kappa\nabla T, \tag{3b}$$

where $\mathbf{U} = (\rho, \mathbf{m}, E)^T$ and

$$\mathbf{u} = \frac{\mathbf{m}}{\rho}, \tag{4a}$$
$$\boldsymbol{\tau} = \frac{1}{Re_\infty}\left(\mathbf{z} + \mathbf{z}^T - \frac{2}{3}tr(\mathbf{z})\mathbf{I}\right), \tag{4b}$$
$$\kappa T = \frac{\gamma}{Re_\infty Pr}\left(\frac{E}{\rho} - \frac{1}{2}|\mathbf{u}|^2\right), \tag{4c}$$

with $tr(\mathbf{z}) := z_{ii}$ the trace of the matrix $\mathbf{z}$. Based on the weak formulation of (2)− (3), which is obtained by multiplying (2)−(3) by test functions, integrating over some domain, and then performing an integration by parts, we can obtain the following semi-discrete LDG scheme : find $\mathbf{U}_h \in V_h$, $\mathbf{z}_h \in \Sigma_h$ and $\mathbf{q}_h \in W_h$, such that for all test functions $\mathbf{V} \in V_h$, $\boldsymbol{\sigma} \in \Sigma_h$ and $\mathbf{p} \in W_h$, the following equations are satisfied

$$\frac{d}{dt}(\mathbf{U}_h, \mathbf{V})_K - (\mathbf{F}_h - \mathbf{G}_h, \nabla\mathbf{V})_K + (\widehat{\mathbf{F}} - \widehat{\mathbf{G}}\mathbf{n}, \mathbf{V})_{\partial K} = 0, \tag{5}$$

and

$$(\mathbf{z}_h, \boldsymbol{\sigma})_K + (\mathbf{u}_h, \nabla\cdot\boldsymbol{\sigma})_K - (\widehat{\mathbf{u}}, \boldsymbol{\sigma}\mathbf{n})_{\partial K} = 0, \tag{6a}$$
$$(\mathbf{q}_h, \mathbf{p})_K - \kappa(T_h, \nabla\cdot\mathbf{p})_K + \kappa(\widehat{T}, \mathbf{p}\cdot\mathbf{n})_{\partial K} = 0, \tag{6b}$$

where $\mathbf{U}_h = (\rho_h, \mathbf{m}_h, E_h)^T$, $\mathbf{F}_h = \mathbf{F}(\mathbf{U}_h)$, $\mathbf{G}_h = \mathbf{G}(\mathbf{U}_h, \boldsymbol{\tau}_h, \mathbf{q}_h)$ and

$$\mathbf{u}_h = \frac{\mathbf{m}_h}{\rho_h},$$
$$\boldsymbol{\tau}_h = \frac{1}{Re_\infty}\left(\mathbf{z}_h + \mathbf{z}_h^T - \frac{2}{3}tr(\mathbf{z}_h)\mathbf{I}\right),$$
$$\kappa T_h = \frac{\gamma}{Re_\infty Pr}\left(\frac{E_h}{\rho_h} - \frac{1}{2}|\mathbf{u}_h|^2\right),$$

$\mathbf{n}$ is the unit outward normal vector to the boundary ∂K. The terms denoted with a hat in (5) and (6) in the cell boundary terms from integration by parts are the so-called numerical fluxes, which are functions defined on the faces and should be designed based on different guiding principles for different PDEs to ensure stability and local solvability of the intermediate variables. Consider a face $e \in \partial K$, and we denote by superscripts L and R the internal interface state and neighboring element interface state, respectively. Note, the neighboring element could be a ghost element which lies exterior to the computational domain.

- If $e \in \Gamma_I$ is an internal face, we choose the local Lax-Friedrichs flux for the convective part and central flux for the other flux terms. In detail,

  $$\widehat{\mathbf{F}} = \frac{1}{2}\left(\mathbf{F}_h^L\mathbf{n} + \mathbf{F}_h^R\mathbf{n} - \alpha_e\left(\mathbf{U}_h^R - \mathbf{U}_h^L\right)\right), \quad \widehat{\mathbf{G}} = \frac{1}{2}\left(\mathbf{G}_h^L + \mathbf{G}_h^R\right),$$
  $$\widehat{\mathbf{u}} = \frac{1}{2}\left(\mathbf{u}_h^L + \mathbf{u}_h^R\right), \quad \widehat{T} = \frac{1}{2}\left(T_h^L + T_h^R\right),$$

  where α_e is the biggest eigenvalue of the Jacobian matrix $\frac{\partial(\mathbf{F}_h\mathbf{n})}{\partial \mathbf{U}}$ on e.
- If $e \in \Gamma_{\text{sub}-\text{in}} \cup \Gamma_{\text{sub}-\text{out}} \cup \Gamma_{\text{sup}-\text{in}} \cup \Gamma_{\text{sup}-\text{out}}$ is a farfield boundary face, we define the right external states as

  $$\mathbf{U}_h^R = \begin{cases} \left(\rho_\infty, \rho_\infty\mathbf{u}_\infty, \frac{p_h^L}{\gamma-1} + \frac{1}{2}\rho_\infty|\mathbf{u}_\infty|^2\right)^T, & \text{if } e \in \Gamma_{\text{sub}-\text{in}}, \\ \mathbf{U}_\infty, & \text{if } e \in \Gamma_{\text{sup}-\text{in}}, \\ \left(\rho_h^L, \mathbf{m}_h^L, \frac{p_\infty}{\gamma-1} + \frac{1}{2}\rho_h^L|\mathbf{u}_h^L|^2\right)^T, & \text{if } e \in \Gamma_{\text{sub}-\text{out}}, \\ \mathbf{U}_h^L, & \text{if } e \in \Gamma_{\text{sup}-\text{out}}, \end{cases}$$

  then the numerical flux $\widehat{\mathbf{F}}$ is similarly defined as in the internal face, and

  $$\widehat{\mathbf{G}} = \mathbf{G}\left(\mathbf{U}_h^R, \boldsymbol{\tau}_h^L, \mathbf{q}_h^L\right), \quad \widehat{\mathbf{u}} = \mathbf{u}_h^R, \quad \widehat{T} = T_h^R.$$
- If $e \in \Gamma_W$ is a solid wall boundary face, in order to weakly prescribe the slip and no-slip boundary conditions, i.e., $\mathbf{u} \cdot \mathbf{n} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$, respectively, for the convective flux $\widehat{\mathbf{F}}$, we again define the right external states as

  $$\mathbf{U}_h^R = \begin{cases} \left(\rho_h^L, \rho_h^L\mathbf{u}_h^L - 2\rho_h^L\left(\mathbf{u}_h^L \cdot \mathbf{n}\right)\mathbf{n}, E_h^L\right)^T, & \text{if } e \in \Gamma_{W,\text{slip}}, \\ \left(\rho_h^L, -\rho_h^L\mathbf{u}_h^L, E_h^L\right)^T, & \text{if } e \in \Gamma_{W,\text{no}-\text{slip}}, \end{cases}$$

  then $\widehat{\mathbf{F}}$ can be formulated the same as in the internal face. The velocity at the boundary is defined as

  $$\widehat{\mathbf{u}} = \mathbf{u}_h^L - \left(\mathbf{u}_h^L \cdot \mathbf{n}\right)\mathbf{n}, \quad \text{if } e \in \Gamma_{W,\text{slip}}, \quad \text{or} \quad \widehat{\mathbf{u}} = \mathbf{0}, \quad \text{if } e \in \Gamma_{W,\text{no}-\text{slip}}.$$

  In addition, for the isothermal case $e \in \Gamma_{W,\text{iso}}$, $\widehat{T}$ and $\widehat{\mathbf{G}}$ are given by

  $$\widehat{T} = T_w \quad \text{and} \quad \widehat{\mathbf{G}} = \left(\mathbf{0}, \boldsymbol{\tau}_h^L, \boldsymbol{\tau}_h^L\widehat{\mathbf{u}} - \mathbf{q}_h^L\right)^T,$$

  with T_w the given temperature on the wall. For the adiabatic case $e \in \Gamma_{W,\text{adia}}$, $\widehat{T}$ and $\widehat{\mathbf{G}}$ are given by

  $$\widehat{T} = T_h^L \quad \text{and} \quad \widehat{\mathbf{G}} = \left(\mathbf{0}, \boldsymbol{\tau}_h^L, \boldsymbol{\tau}_h^L\widehat{\mathbf{u}}\right)^T.$$

Now we have completed the definition of the LDG scheme for the NS equations.

4 Time discretizations

The LDG spatial discretization for the NS equations typically results in a system of ODEs

$$\begin{aligned} &\mathbf{u}_t = \mathbf{F}_N(t, \mathbf{u}) + \mathbf{F}_S(t, \mathbf{u}), \quad \text{in } (0, \mathrm{T}], \\ &\mathbf{u}(0) = \mathbf{u}_0, \end{aligned} \tag{7}$$

which contains a non-stiff term $\mathbf{F}_N(t,\mathbf{u})$ as well as a stiff term $\mathbf{F}_S(t,\mathbf{u})$. One would then need to use a suitable ODE solver to discretize the temporal variable. Generally, in many cases, the non-stiff component $\mathbf{F}_N(t,\mathbf{u})$ is nonlinear and the stiff component $\mathbf{F}_S(t,\mathbf{u})$ is linear, such as Burgers equation with viscous term, KdV type equations, etc. It would be desirable to treat the different components separately, more specifically, to treat the non-stiff term explicitly and stiff term implicitly. This kind of treatment can not only relax the severe time step restriction due to implicit integration for the stiff term, but also be easy to implement since we apply explicit discretization for potentially nonlinear non-stiff terms. And what we would like to emphasize is that, even though the stiff term of the NS equations is nonlinear with respect to all the conserved variables, we can still update the variables at the new time level by solving a series of linear systems due to the special structure of the NS equations. In the following work, we will consider two kinds of time discretizations: IMEX-RK methods and semi-implicit SDC methods.

To numerically solve (7), we divide the time interval $(0,T]$ into M elements by the partition $0 = t_0 < t_1 < \cdots < t_n < \cdots < t_M = T$ and denote by $\triangle t_n = t_{n+1} - t_n$ the time step at level n and $\mathbf{u}_n$ the numerical approximation of $\mathbf{u}(t_n)$.

4.1 IMEX-RK discretizations

By applying the IMEX-RK time marching methods, the numerical solution of (7) advanced from time t_n to t_{n+1} is given by

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}_n,\\
\mathbf{u}^{(i)} &= \mathbf{u}_n + \triangle t_n \sum_{j=1}^{i-1} \hat{a}_{i,j}\mathbf{F}_N\left(t_n^j,\mathbf{u}^{(j)}\right) + \triangle t_n \sum_{j=2}^{i} a_{i,j}\mathbf{F}_S\left(t_n^j,\mathbf{u}^{(j)}\right), \quad 2 \le i \le s+1,\\
\mathbf{u}_{n+1} &= \mathbf{u}_n + \triangle t_n \sum_{i=1}^{s+1} \hat{b}_i\mathbf{F}_N\left(t_n^i,\mathbf{u}^{(i)}\right) + \triangle t_n \sum_{i=2}^{s+1} b_i\mathbf{F}_S\left(t_n^i,\mathbf{u}^{(i)}\right),
\end{aligned}$$

where $t_n^i = t_n + c_i\triangle t_n$ with $c_i = \sum_{j=1}^{i-1}\hat{a}_{ij} = \sum_{j=2}^{i} a_{ij}$. If we define $\hat{\mathbf{A}} = (\hat{a}_{ij})$, $\mathbf{A} = (a_{ij}) \in R^{(s+1)\times(s+1)}$, $\hat{\mathbf{b}} = (\hat{b}_1,\cdots,\hat{b}_{s+1})^T$, $\mathbf{b} = (0, b_2,\cdots,b_{s+1})^T$ and $\mathbf{c} = (0, c_2,\cdots,c_{s+1})^T$, we can interpret the IMEX-RK methods simply and clearly using a Butcher tableau

$$\begin{array}{c|c|c}
\mathbf{c} & \hat{\mathbf{A}} & \mathbf{A} \\
\hline
 & \hat{\mathbf{b}}^T & \mathbf{b}^T
\end{array}.$$

For a detailed introduction to IMEX-RK schemes, we refer the readers to [18, 32, 33]. The followings are examples of the first, second and third order IMEX-RK methods, respectively:

- First order (one stage):

$$\begin{array}{c|cc|cc}
0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 \\
\hline
 & 1 & 0 & 0 & 1
\end{array}$$

It is obvious that the first order IMEX-RK method is just taking the forward Euler discretization for the non-stiff term and the backward Euler discretization for the stiff term.

- Second order (two stages):

$$\begin{array}{c|ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma & \gamma & 0 & 0 & 0 & \gamma & 0 \\
1 & \delta & 1-\delta & 0 & 0 & 1-\gamma & \gamma \\
\hline
 & \delta & 1-\delta & 0 & 0 & 1-\gamma & \gamma
\end{array}$$

with $\gamma = 1 - \frac{\sqrt{2}}{2}$ and $\delta = 1 - \frac{1}{2\gamma}$.

- Third order (three stages):

$$\begin{array}{c|cccc|cccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma & \gamma & 0 & 0 & 0 & 0 & \gamma & 0 & 0 \\
\frac{1+\gamma}{2} & \frac{1+\gamma}{2} - \alpha_1 & \alpha_1 & 0 & 0 & 0 & \frac{1-\gamma}{2} & \gamma & 0 \\
1 & 0 & 1-\alpha_2 & \alpha_2 & 0 & 0 & \beta_1 & \beta_2 & \gamma \\
\hline
 & 0 & \beta_1 & \beta_2 & \gamma & 0 & \beta_1 & \beta_2 & \gamma
\end{array}$$

with γ the middle root of $6x^3 - 18x^2 + 9x - 1 = 0$, $\gamma \approx 0.435866521508459$, $\beta_1 = -\frac{3}{2}\gamma^2 + 4\gamma - \frac{1}{4}$, $\beta_2 = \frac{3}{2}\gamma^2 - 5\gamma + \frac{5}{4}$, $\alpha_1 = -0.35$ and $\alpha_2 = \frac{\frac{1}{3} - 2\gamma^2 - 2\beta_2\alpha_1\gamma}{\gamma(1-\gamma)}$.

4.2 Semi-discrete SDC discretizations

Algorithm 1 Semi-discrete SDC method

```
Compute the initial first order approximation
  $\mathbf{u}_{n,0}^1 = \mathbf{u}_n$
  for $m = 0, \cdots, P-1$ do
    $\mathbf{u}_{n,m+1}^1 = \mathbf{u}_{n,m}^1 + \triangle t_{n,m} \left( \mathbf{F}_N \left( t_{n,m}, \mathbf{u}_{n,m}^1 \right) + \mathbf{F}_S \left( t_{n,m+1}, \mathbf{u}_{n,m+1}^1 \right) \right)$
  end for
Compute successive corrections
  for $k = 1, \cdots, K$ do
    $\mathbf{u}_{n,0}^{k+1} = \mathbf{u}_n$
  end for
  for $m = 0, \cdots, P-1$ do
    $\mathbf{u}_{n,m+1}^{k+1} = \mathbf{u}_{n,m}^{k+1} + \triangle t_{n,m} \left( \mathbf{F}_N \left( t_{n,m}, \mathbf{u}_{n,m}^{k+1} \right) - \mathbf{F}_N \left( t_{n,m}, \mathbf{u}_{n,m}^{k} \right) \right)$
        $+ \triangle t_{n,m} \left( \mathbf{F}_S \left( t_{n,m+1}, \mathbf{u}_{n,m+1}^{k+1} \right) - \mathbf{F}_S \left( t_{n,m+1}, \mathbf{u}_{n,m+1}^{k} \right) \right)$
        $+ I_m^{m+1} \left( \mathbf{F}_N \left( t, \mathbf{u}^k \right) + \mathbf{F}_S \left( t, \mathbf{u}^k \right) \right)$
  end for
  Finally we have $\mathbf{u}_{n+1} = \mathbf{u}_{n,P}^{K+1}$.
```

The SDC time discretization is based on the Picard integral equation and low order time integration methods, which are corrected iteratively, with the order of accuracy increased by one for each additional iteration. The key advantage of SDC method is that it can systematically and simply develop time integration methods of arbitrary high order of accuracy.

To define the semi-discrete SDC scheme, we further divide the time interval $[t_n, t_{n+1}]$ into P subintervals by points $t_{n,m}$ for $m = 0, 1, \cdots, P$ such that $t_n = t_{n,0} < t_{n,1} < \cdots < t_{n,m} < \cdots < t_{n,P} = t_{n+1}$. Let $\triangle t_{n,m} = t_{n,m+1} - t_{n,m}$ and $\mathbf{u}^k_{n,m}$ denote the k-th order approximation to $\mathbf{u}(t_{n,m})$. We choose the points $\{t_{n,m}\}^P_{m=0}$ as Gauss-Lobatto nodes on $[t_n, t_{n+1}]$. Then the algorithm to calculate $(k+1)$-th order accuracy numerical solution of (7) advanced from time t_n to t_{n+1} is given in Algorithm 1, where $I^{m+1}_m \left(\mathbf{F}_N\left(t, \mathbf{u}^k\right) + \mathbf{F}_S\left(t, \mathbf{u}^k\right)\right)$ is the integral of the P-th degree interpolating polynomial on the $P+1$ points $\left(t_{n,m}, \mathbf{F}_N\left(t_{n,m}, \mathbf{u}^k_{n,m}\right) + \mathbf{F}_S\left(t_{n,m}, \mathbf{u}^k_{n,m}\right)\right)^P_{m=0}$ over the subinterval $[t_{n,m}, t_{n,m+1}]$.

4.3 Fully discrete schemes

Finally, as the implementation of IMEX-RK and SDC time discretizations to the semi-discrete LDG scheme (5) and (6), we will present the fully discrete LDG schemes in the following for the NS equations. We only present the third-order fully discrete scheme here and the lower order fully discrete schemes can be deduced similarly.

- Third order IMEX-RK-LDG scheme

  The LDG scheme with the third order IMEX-RK time marching method is given as :

$$
\begin{aligned}
\left(\mathbf{U}_h^{n,1}, \mathbf{V}\right)_K = {} & \left(\mathbf{U}_h^n, \mathbf{V}\right)_K + \gamma \triangle t_n \left(\left(\mathbf{F}_h^n, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^n, \mathbf{V}\right)_{\partial K}\right) \\
& - \gamma \triangle t_n \left(\left(\mathbf{G}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,1}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right), \qquad (8a) \\
\left(\mathbf{U}_h^{n,2}, \mathbf{V}\right)_K = {} & \left(\mathbf{U}_h^n, \mathbf{V}\right)_K + \left(\frac{1+\gamma}{2} - \alpha_1\right) \triangle t_n \left(\left(\mathbf{F}_h^n, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^n, \mathbf{V}\right)_{\partial K}\right) \\
& + \alpha_1 \triangle t_n \left(\left(\mathbf{F}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,1}, \mathbf{V}\right)_{\partial K}\right) \\
& - \frac{1-\gamma}{2} \triangle t_n \left(\left(\mathbf{G}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,1}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& - \gamma \triangle t_n \left(\left(\mathbf{G}_h^{n,2}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,2}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right), \qquad (8b) \\
\left(\mathbf{U}_h^{n,3}, \mathbf{V}\right)_K = {} & \left(\mathbf{U}_h^n, \mathbf{V}\right)_K + (1-\alpha_2) \triangle t_n \left(\left(\mathbf{F}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,1}, \mathbf{V}\right)_{\partial K}\right) \\
& + \alpha_2 \triangle t_n \left(\left(\mathbf{F}_h^{n,2}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,2}, \mathbf{V}\right)_{\partial K}\right) \\
& - \beta_1 \triangle t_n \left(\left(\mathbf{G}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,1}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& - \beta_2 \triangle t_n \left(\left(\mathbf{G}_h^{n,2}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,2}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& - \gamma \triangle t_n \left(\left(\mathbf{G}_h^{n,3}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,3}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right), \qquad (8c) \\
\left(\mathbf{U}_h^{n+1}, \mathbf{V}\right)_K = {} & \left(\mathbf{U}_h^n, \mathbf{V}\right)_K + \beta_1 \triangle t_n \left(\left(\mathbf{F}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,1}, \mathbf{V}\right)_{\partial K}\right) \\
& + \beta_2 \triangle t_n \left(\left(\mathbf{F}_h^{n,2}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,2}, \mathbf{V}\right)_{\partial K}\right) \\
& + \gamma \triangle t_n \left(\left(\mathbf{F}_h^{n,3}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^{n,3}, \mathbf{V}\right)_{\partial K}\right) \\
& - \beta_1 \triangle t_n \left(\left(\mathbf{G}_h^{n,1}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,1}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& - \beta_2 \triangle t_n \left(\left(\mathbf{G}_h^{n,2}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,2}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& - \gamma \triangle t_n \left(\left(\mathbf{G}_h^{n,3}, \nabla \mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^{n,3}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right), \qquad (8d)
\end{aligned}
$$

$$\left(\mathbf{z}_h^{n,l}, \boldsymbol{\sigma}\right)_K = -\left(\mathbf{u}_h^{n,l}, \nabla \cdot \boldsymbol{\sigma}\right)_K + \left(\widehat{\mathbf{u}}^{n,l}, \boldsymbol{\sigma}\mathbf{n}\right)_{\partial K}, \quad l = 1, 2, 3, \tag{9a}$$

$$\left(\mathbf{q}_h^{n,l}, \mathbf{p}\right)_K = \kappa\left(T_h^{n,l}, \nabla \cdot \mathbf{p}\right)_K - \kappa\left(\widehat{T}^{n,l}, \mathbf{p} \cdot \mathbf{n}\right)_{\partial K}, \quad l = 1, 2, 3, \tag{9b}$$

for any test function $\mathbf{V} \in V_h$, $\boldsymbol{\sigma} \in \Sigma_h$, $\mathbf{p} \in W_h$, with $\mathbf{U}_h^{n,l} = \left(\rho_h^{n,l}, \mathbf{m}_h^{n,l}, E_h^{n,l}\right)^T$, $\mathbf{F}_h^{n,l} = \mathbf{F}\left(\mathbf{U}_h^{n,l}\right)$, $\mathbf{G}_h^{n,l} = \mathbf{G}\left(\mathbf{U}_h^{n,l}, \boldsymbol{\tau}_h^{n,l}, \mathbf{q}_h^{n,l}\right)$, $l = 0, 1, 2, 3$, and

$$\mathbf{u}_h^{n,l} = \frac{\mathbf{m}_h^{n,l}}{\rho_h^{n,l}}, \quad l = 0, 1, 2, 3, \tag{10a}$$

$$\boldsymbol{\tau}_h^{n,l} = \frac{1}{Re_\infty}\left(\mathbf{z}_h^{n,l} + \left(\mathbf{z}_h^{n,l}\right)^T - \frac{2}{3}tr\left(\mathbf{z}_h^{n,l}\right)\mathbf{I}\right), \quad l = 1, 2, 3, \tag{10b}$$

$$\kappa T_h^{n,l} = \frac{\gamma}{Re_\infty Pr}\left(\frac{E_h^{n,l}}{\rho_h^{n,l}} - \frac{1}{2}\left|\mathbf{u}_h^{n,l}\right|^2\right), \quad l = 1, 2, 3, \tag{10c}$$

and notations $*^{n,0} = *^n$.

- High order SDC-LDG scheme
  The LDG scheme with high order SDC time marching method reads : for any test function $\mathbf{V} \in V_h$, $\boldsymbol{\sigma} \in \Sigma_h$, $\mathbf{p} \in W_h$, the following first order approximation as well as successive correction equations are satisfied

$$\begin{aligned}
\left((\mathbf{U}_h)_{n,m+1}^1, \mathbf{V}\right)_K = {} & \left((\mathbf{U}_h)_{n,m}^1, \mathbf{V}\right)_K + \triangle t_{n,m}\left(\left((\mathbf{F}_h)_{n,m}^1, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}_{n,m}^1, \mathbf{V}\right)_{\partial K}\right) \\
& - \triangle t_{n,m}\left(\left((\mathbf{G}_h)_{n,m+1}^1, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}_{n,m+1}^1\mathbf{n}, \mathbf{V}\right)_{\partial K}\right),
\end{aligned} \tag{11a}$$

$$\begin{aligned}
\left((\mathbf{U}_h)_{n,m+1}^{k+1}, \mathbf{V}\right)_K = {} & \left((\mathbf{U}_h)_{n,m}^{k+1}, \mathbf{V}\right)_K + \triangle t_{n,m}\left(\left((\mathbf{F}_h)_{n,m}^{k+1}, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}_{n,m}^{k+1}, \mathbf{V}\right)_{\partial K}\right) \\
& - \triangle t_{n,m}\left(\left((\mathbf{F}_h)_{n,m}^{k}, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}_{n,m}^{k}, \mathbf{V}\right)_{\partial K}\right) \\
& - \triangle t_{n,m}\left(\left((\mathbf{G}_h)_{n,m+1}^{k+1}, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}_{n,m+1}^{k+1}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& + \triangle t_{n,m}\left(\left((\mathbf{G}_h)_{n,m+1}^{k}, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}_{n,m+1}^{k}\mathbf{n}, \mathbf{V}\right)_{\partial K}\right) \\
& + I_m^{m+1}\left(\left((\mathbf{F}_h)^k, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{F}}^k, \mathbf{V}\right)_{\partial K}\right) \\
& - I_m^{m+1}\left(\left((\mathbf{G}_h)^k, \nabla\mathbf{V}\right)_K - \left(\widehat{\mathbf{G}}^k\mathbf{n}, \mathbf{V}\right)_{\partial K}\right),
\end{aligned} \tag{11b}$$

for $m = 0, \cdots, P-1$, $k = 1, \cdots, K$, and

$$\left((\mathbf{z}_h)_{n,m}^k, \boldsymbol{\sigma}\right)_K = -\left((\mathbf{u}_h)_{n,m}^k, \nabla \cdot \boldsymbol{\sigma}\right)_K + \left(\widehat{\mathbf{u}}_{n,m}^k, \boldsymbol{\sigma}\mathbf{n}\right)_{\partial K}, \tag{12a}$$

$$\left((\mathbf{q}_h)_{n,m}^k, \mathbf{p}\right)_K = \kappa\left(\left((T_h)_{n,m}^k, \nabla \cdot \mathbf{p}\right)_K - \left(\widehat{T}_{n,m}^k, \mathbf{p} \cdot \mathbf{n}\right)_{\partial K}\right), \tag{12b}$$

for $m = 0, \cdots, P$, $k = 1, \cdots, K+1$, where $(\mathbf{U}_h)_{n,m}^k = \left((\rho_h)_{n,m}^k, (\mathbf{m}_h)_{n,m}^k, (E_h)_{n,m}^k\right)^T$, $(\mathbf{F}_h)_{n,m}^k = \mathbf{F}\left((\mathbf{U}_h)_{n,m}^k\right)$, $(\mathbf{G}_h)_{n,m}^k = \mathbf{G}\left((\mathbf{U}_h)_{n,m}^k, (\boldsymbol{\tau}_h)_{n,m}^k, (\mathbf{q}_h)_{n,m}^k\right)$ and

$$(\mathbf{u}_h)_{n,m}^k = \frac{(\mathbf{m}_h)_{n,m}^k}{(\rho_h)_{n,m}^k}, \tag{13a}$$

$$(\boldsymbol{\tau}_h)_{n,m}^k = \frac{1}{Re_\infty}\left((\mathbf{z}_h)_{n,m}^k + \left((\mathbf{z}_h)_{n,m}^k\right)^T - \frac{2}{3}tr\left((\mathbf{z}_h)_{n,m}^k\right)\mathbf{I}\right), \tag{13b}$$

$$\kappa(T_h)_{n,m}^k = \frac{\gamma}{Re_\infty Pr}\left(\frac{(E_h)_{n,m}^k}{(\rho_h)_{n,m}^k} - \frac{1}{2}\left|(\mathbf{u}_h)_{n,m}^k\right|^2\right), \tag{13c}$$

for $m = 0, \cdots, P$, $k = 1, \cdots, K+1$.

Finally we can obtain the $(K+1)$-th approximation $\mathbf{U}_h^{n+1} = (\mathbf{U}_h)_{n,P}^{K+1}$ at time t_{n+1}.

5 Mesh adaptation

In this section, we mainly present the detailed implementation of the h-adaptation algorithm for the structured mesh consisting of quadrilateral elements in 2D or hexahedral elements in 3D, including the method to refine and coarsen elements, the criteria to select candidate elements for refinement and coarsening, and the specific procedure of mesh adaptation.

5.1 Refinement and coarsening of elements

For the data structure of adaptive mesh, we adopt a generalized binary tree, i.e., quadtree in 2D and octree in 3D, to store all the elements, that is, a parent element is always divided into 4 in 2D or 8 in 3D child elements. The DG method is extremely local in data communication and allows the appearance of hanging nodes. This means that an element can be refined an unlimited number of times, regardless of its immediate neighbors. However, if a huge difference between the levels of adjacent elements exists, the stability of the resulting numerical scheme will be damaged. In order to enhance the mesh quality and the stability of the scheme, we impose the quadtree or octree balanced, that is, the levels between two adjacent elements differ at most by 1.

The numerical solution $\mathbf{U}_h$ restricted in element K can be expressed as

$$\mathbf{U}_h(\mathbf{x},t)|_K = \sum_{l=1}^{N_b} \mathbf{U}_l^K(t) v_l^K(\mathbf{x}), \tag{14}$$

where $\mathbf{U}_l^K(t)$ and $v_l^K(\mathbf{x})$, $l=1,\cdots,N_b$ denote the degrees of freedom and basis functions, respectively, in element K. To maintain the accuracy and local conservation of the solutions, we adopt L^2 projection to obtain the degrees of freedom in the new generated elements during refinement and coarsening. In detail, provided the numerical solution $\mathbf{U}_h$ is already known on the mesh $\mathcal{T}_h(t_n)$, we require to determine the degrees of freedom $\mathbf{U}_l^{K'}(t_n)$, $l=1,\cdots,N_b$ in the new element $K' \in \mathcal{T}_h(t_{n+1})$. Let $\mathbf{U}_h'$ be the L^2 projection of $\mathbf{U}_h$, which is computed through the following equations:

$$\int_{K'} \mathbf{U}_h'|_{K'} v_l^{K'}(\mathbf{x}) d\mathbf{x} = \int_{K'} \mathbf{U}_h v_l^{K'}(\mathbf{x}) d\mathbf{x}, \quad l=1,\cdots,N_b. \tag{15}$$

Then the degrees of freedom $\mathbf{U}_l^{K'}(t_n)$, $l=1,\cdots,N_b$ can be obtained by substituting (14) into (15).

5.2 Indicators

A criterion will be presented to initially determine the candidate elements for refinement and coarsening in a given mesh. According to [16, 17], the gradient of density finds shock and contact discontinuity well, the divergence of velocity is direction independent and very effective in locating shock including strong shock and weak shock and the curl of velocity is also direction independent and very effective in finding shear and vortex. For these three different indicators, we compute the following quantities respectively:

$$\eta_{gi} = |\nabla \rho| d_i^{\frac{3}{2}}, \quad \eta_{di} = |\nabla \cdot \mathbf{u}| d_i^{\frac{3}{2}}, \quad \eta_{ci} = |\nabla \times \mathbf{u}| d_i^{\frac{3}{2}}, \quad i = 1, \cdots, N_c,$$

where N_c is the total number of elements, $d_i = \sqrt{|K|}$ or $d_i = \sqrt[3]{|K|}$ depending on the dimension. The standard deviation of the gradient of density, the divergence and the curl of velocity are

$$\eta_g = \sqrt{\frac{\sum_{i=1}^{N_c} \eta_{gi}^2}{N_c}}, \quad \eta_d = \sqrt{\frac{\sum_{i=1}^{N_c} \eta_{di}^2}{N_c}}, \quad \eta_c = \sqrt{\frac{\sum_{i=1}^{N_c} \eta_{ci}^2}{N_c}},$$

respectively. In our work, a single indicator or a combination of two of the above indicators will be taken into account for different problems. Suppose the level of all the elements in the initial mesh equals zero, and each element in the initial mesh can be refined at most LEV times. Based on the values of indicators, if only a single indicator, e.g., η_{gi} is considered, then

- If $\eta_{gi} > \omega_1 \eta_g$ and the level of $K < LEV$, then K is marked as a candidate element for refinement,
- If $\eta_{gi} < \omega_2 \eta_g$ and the level of $K > 0$, then K is marked as a candidate element for coarsening,

with problem dependent parameters ω_l, $l = 1, 2$. If a combination of two indicators, e.g., η_{gi} and η_{di} are employed, then

- If $\eta_{gi} > \omega_1 \eta_g$ or $\eta_{di} > \omega_3 \eta_d$, and the level of $K < LEV$, then K is marked as a candidate element for refinement,
- If $\eta_{gi} < \omega_2 \eta_g$ and $\eta_{di} < \omega_4 \eta_d$, and the level of $K > 0$, then K is marked as a candidate element for coarsening,

with problem dependent parameters ω_l, $l = 1, 2, 3, 4$. In general, ω_1 and ω_3 are chosen between 1.1 and 1.5, ω_2 and ω_4 are chosen between 0.1 and 0.5.

5.3 Strategy for refinement and coarsening

In the above subsection, several candidate elements for refinement and coarsening are selected through given criteria. And whether these elements can be ultimately refined or coarsened, they also have to adhere to the principle that the level difference between two adjacent elements is at most 1 to improve the stability and accuracy of the LDG methods. To satisfy this principle, we adopt the strategy "refinement must, coarsening can" in [15], which means that a candidate element flagged for refinement is certainly to be refined, and in comparison, a candidate element marked for coarsening may not be coarsened.

After the execution of Algorithm 2 and Algorithm 3 in sequence, we can obtain two sets C_r and C_c which contain the final elements for refinement and coarsening, respectively. In Algorithm 2, all the candidate elements for refinement will be refined, and some neighbors of these elements are also added to the set C_r to ensure the mesh quality. In Algorithm 3, a candidate element for coarsening may be deleted from the set C_c due to either the absence of marked for coarsening for its brother elements, or the level difference with its neighbors.

5.4 Flow chart for mesh adaptation

As an end of this section, we present the general procedure for the adaptive LDG methods. Given the initial coarse mesh $\mathcal{T}_h(t_0)$ and final computational time T,

Algorithm 2 Refinement of candidate elements

```
Suppose the set of all the candidate elements for refinement is $C_r$
for each element $K$ in $C_r$ do
    for each neighbor $\tilde{K}$ of $K$ do
        if $\tilde{K} \notin C_r$ and the level of $\tilde{K}$ < the level of $K$ then
            add $\tilde{K}$ to $C_r$
        end if
    end for
end for
```

Algorithm 3 Coarsening of candidate elements

```
Suppose the set of all the candidate elements for coarsening is $C_c$
for each element $K$ in $C_c$ do
    Set $K_p$ as the parent element of $K$
    if all the child elements of $K_p$ belong to $C_c$ then
        if the level of all the neighbors of $K_p \leq$ the level of $K$ then
            keep $K$ in $C_c$
        else
            delete $K$ from $C_c$
        end if
    else
        delete $K$ from $C_c$
    end if
end for
```

Step 1 Initialize the level of all the elements in $\mathcal{T}_h(t_0)$ to be 0 and obtain the degrees of freedom in each element $\left\{\mathbf{U}_l^K(t_0),\ l = 1, \cdots, N_b,\ K \in \mathcal{T}_h(t_0)\right\}$ through L^2 projection for the initial exact solution.

Step 2 Given the mesh $\mathcal{T}_h(t_n)$ and the degrees of freedom in each element $\left\{\mathbf{U}_l^K(t_n),\ l = 1, \cdots, N_b,\ K \in \mathcal{T}_h(t_n)\right\}$, an updated mesh $\mathcal{T}_h(t_{n+1})$ is gained as follows:

- Identify the candidate elements for refinement and coarsening in $\mathcal{T}_h(t_n)$ through criteria in subsection 5.2.
- Determine the final element sets C_r and C_c for refinement and coarsening through Algorithm 2 and Algorithm 3, respectively and subsequently.
- For each element in C_r, divide it into 4 in 2D or 8 in 3D child elements, get the degrees of freedom of child elements by use of L^2 projection, and increase the level of child elements by one.
- For child elements with the same parent element in C_c, delete them and get the degrees of freedom of the parent element through L^2 projection.

Step 3 Obtain $\left\{\mathbf{U}_l^K(t_n+1),\ l = 1, \cdots, N_b,\ K \in \mathcal{T}_h(t_n+1)\right\}$ by aid of the LDG method coupled with an IMEX-RK or SDC method.

Step 4 If $t_{n+1} < T$, go to Step 2.

6 Numerical experiments

In this section, we will carry out several numerical experiments including accuracy tests and other benchmark problems for the NS equations to evaluate the performance of the h-adaptive LDG methods. In addition, IMEX-RK and/or SDC time integration methods are adopted to solve the ODEs resulting from the spatial discretizations. Note, at every stage for the evolution in time, GMRES solver without any preconditioner is adopted to solve the two successive linear systems with respect to momentum and energy equations. In all experiments, the time step $\triangle t_n$ at time t_n is chosen to satisfy the following CFL condition

$$\frac{d\triangle t_n}{\min_K h_K}\max_K(|\mathbf{u}_K|+a_K)<cfl,$$

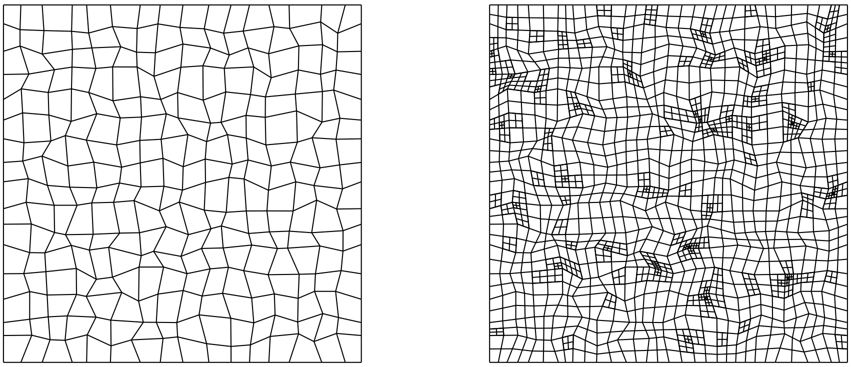
where $d\leq 3$ denotes the dimension, h_K is the diameter of element K, $\mathbf{u}_K$ and a_K are the velocity vector and speed of sound at element K, respectively, cfl is the CFL number taken as 0.98, 0.3, 0.18 for the first, second and third order time discretizations, respectively. In the following simulations, unless otherwise stated, an element in a mesh will be refined at most $LEV=3$ times, that is, the depth of the generalized binary tree is 3.

Example 1 ***Accuracy test in two dimension***

In order to demonstrate the uniform high order accuracy both in space and time of our proposed h-adaptive LDG scheme in conjunction with IMEX-RK or SDC methods, for the first test case we consider a smooth exact solution

$$\begin{aligned}
\rho(\mathbf{x},t)&=0.6+0.1\sin(5\pi t)\cos(2\pi x)\cos(2\pi y),\\
u(\mathbf{x},t)&=\sin(3\pi t)\sin(2\pi x)\sin(2\pi y),\\
v(\mathbf{x},t)&=\sin(3\pi t)\sin(4\pi x)\sin(4\pi y),\\
p(\mathbf{x},t)&=0.8+0.1\sin(\pi t)\sin(2\pi x)\cos(2\pi y),
\end{aligned}\tag{16}$$

for the compressible NS Eqs. (1) with $d=2$ and an additional source term, which is acquired by inserting (16) into (1). Additionally, we take the computational domain as $[0,1]^2$ with periodic boundary conditions and Reynolds number $Re_\infty=200, 1000, 5000$, respectively. All the simulations are performed until time $T=0.1$ on a set of successively globally refined background meshes containing $N\times N$, $N=16, 32, 64, 128$ quadrilateral elements, respectively. For a fixed $N\times N$ background mesh, we randomly choose $2N$ candidate elements marked for refinement and the remaining elements marked for coarsening at each time step, and gain the final element sets C_r and C_c through Algorithms 2 and 3. The initial 16×16 background mesh and the adaptive mesh at some time based on the 32×32 background mesh are shown in Fig. 2. We adopt $(k+1)$-th order IMEX-RK or SDC time marching methods when k-th degree polynomials are employed. The L^2 errors and orders of accuracy of density, velocity and pressure for both two schemes are presented in Tables 1 and 2, respectively, which illustrate that our two schemes can both obtain the expected optimal order of accuracy for these physical quantities with various Reynolds numbers. Moreover, by comparing the results in Tables 1 and 2, we can conclude that the IMEX-RK and SDC methods with the same order share nearly identical accuracy based on the background meshes with the same number of elements. For further efficiency comparison between IMEX-RK and SDC methods, we run the simulations on the uniform 64×64 mesh to eliminate the influence of frequent mesh refinement and



(a) Initial 16×16 background mesh

(b) Adaptive mesh based on 32×32 background mesh

Fig. 2 Sample mesh for accuracy test in two dimension

coarsening. It can be seen from Table 3 that, compared with the IMEX-RK scheme, the SDC time discretization takes nearly the same time for the second order case and twice the time for the third order case, respectively. This can evidently be explained by their respective number of stages (there are 2 stages for both the second order IMEX-RK and SDC schemes, 3 stages for the third order IMEX-RK scheme and 6 stages for the third

Table 1 Accuracy test for adaptive IMEX-RK-LDG scheme of 2D NS equations

Re_∞	k	N	$\|\rho - \rho_h\|_{L^2}$	Order	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Order	$\|p - p_h\|_{L^2}$	Order
200	1	16	4.93E-03	–	2.72E-02	–	8.78E-03	–
		32	1.04E-03	2.24	6.15E-03	2.14	1.81E-03	2.27
		64	2.34E-04	2.15	1.44E-03	2.09	4.08E-04	2.14
		128	5.61E-05	2.02	3.51E-04	2.03	9.86E-05	2.05
	2	16	6.57E-04	–	3.17E-03	–	1.13E-03	–
		32	8.52E-05	2.94	3.82E-04	3.05	1.40E-04	3.01
		64	1.07E-05	2.99	4.76E-05	3.00	1.76E-05	2.99
		128	1.28E-06	3.06	5.96E-06	3.00	2.16E-06	3.02
1000	1	16	4.51E-03	–	2.39E-02	–	8.05E-03	–
		32	1.03E-03	2.11	5.84E-03	2.03	1.81E-03	2.14
		64	2.50E-04	2.05	1.40E-03	2.05	4.38E-04	2.05
		128	6.20E-05	2.01	3.46E-04	2.02	1.08E-04	2.01
	2	16	6.75E-04	–	3.32E-03	–	1.23E-03	–
		32	8.47E-05	2.99	3.94E-04	3.07	1.51E-04	3.02
		64	1.08E-05	2.96	4.86E-05	3.02	1.90E-05	2.99
		128	1.36E-06	3.00	6.02E-06	3.01	2.37E-06	3.00
5000	1	16	4.51E-03	–	2.39E-02	–	8.10E-03	–
		32	1.03E-03	2.12	5.83E-03	2.03	1.82E-03	2.15
		64	2.50E-04	2.05	1.40E-03	2.05	4.39E-04	2.05
		128	6.19E-05	2.01	3.46E-04	2.01	1.08E-04	2.01
	2	16	6.93E-04	–	3.65E-03	–	1.27E-03	–
		32	8.67E-05	3.00	4.44E-04	3.03	1.56E-04	3.02
		64	1.06E-05	3.02	5.18E-05	3.09	1.94E-05	3.00
		128	1.35E-06	2.98	6.24E-06	3.05	2.43E-06	3.00

Table 2 Accuracy test for adaptive SDC-LDG scheme of 2D NS equations

Re_∞	k	N	$\|\rho - \rho_h\|_{L^2}$	Order	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Order	$\|p - p_h\|_{L^2}$	Order
200	1	16	4.96E-03	–	2.68E-02	–	8.81E-03	–
		32	1.04E-03	2.25	6.16E-03	2.12	1.81E-03	2.27
		64	2.34E-04	2.15	1.44E-03	2.09	4.10E-04	2.14
		128	5.61E-05	2.06	3.52E-04	2.03	9.87E-05	2.05
	2	16	6.56E-04	–	3.15E-03	–	1.12E-03	–
		32	8.51E-05	2.94	3.81E-04	3.04	1.39E-04	3.01
		64	1.07E-05	2.99	4.76E-05	3.00	1.75E-05	2.99
		128	1.28E-06	3.06	5.95E-06	3.00	2.16E-06	3.02
1000	1	16	4.51E-03	–	2.39E-02	–	8.05E-03	–
		32	1.03E-03	2.11	5.84E-03	2.03	1.81E-03	2.14
		64	2.50E-04	2.05	1.40E-03	2.05	4.38E-04	2.05
		128	6.20E-05	2.01	3.46E-04	2.02	1.08E-04	2.01
	2	16	6.68E-04	–	3.30E-03	–	1.22E-03	–
		32	8.47E-05	2.97	3.97E-04	3.05	1.51E-04	3.01
		64	1.08E-05	2.96	4.86E-05	3.02	1.90E-05	2.99
		128	1.36E-06	3.00	6.02E-06	3.01	2.37E-06	3.00
5000	1	16	4.51E-03	–	2.39E-02	–	8.10E-03	–
		32	1.03E-03	2.12	5.83E-03	2.03	1.82E-03	2.15
		64	2.50E-04	2.05	1.40E-03	2.05	4.39E-04	2.05
		128	6.19E-05	2.01	3.46E-04	2.01	1.08E-04	2.01
	2	16	6.93E-04	–	3.65E-03	–	1.27E-03	–
		32	8.67E-05	3.00	4.44E-04	3.03	1.56E-04	3.02
		64	1.06E-05	3.02	5.18E-05	3.09	1.94E-05	3.00
		128	1.35E-06	2.98	6.24E-06	3.05	2.43E-06	3.00

order SDC scheme). However, unlike the IMEX-RK scheme, the SDC method of arbitrary high order of accuracy can be systematically and simply developed.

Example 2 ***Accuracy test in three dimension***

In this example, we extend our accuracy test to three dimension, which is much more challenging to implement the procedure of refinement and coarsening for a general hexahedral mesh. To this end, similar to the two dimensional counterpart, an artificial exact smooth solution

$$\begin{aligned}
\rho(\mathbf{x},t) &= 2 + 0.5e^{-t}\sin(2\pi x)\sin(2\pi y)\sin(2\pi z),\\
u(\mathbf{x},t) &= e^{-2t}\sin(2\pi x)\cos(2\pi y)\sin(2\pi z),\\
v(\mathbf{x},t) &= e^{-2t}\cos(2\pi x)\sin(2\pi y)\cos(2\pi z),\\
w(\mathbf{x},t) &= e^{-2t}\sin(2\pi x)\sin(2\pi y)\cos(2\pi z),\\
p(\mathbf{x},t) &= 1.5 + 0.2e^{-t}\cos(2\pi x)\cos(2\pi y)\cos(2\pi z),
\end{aligned} \tag{17}$$

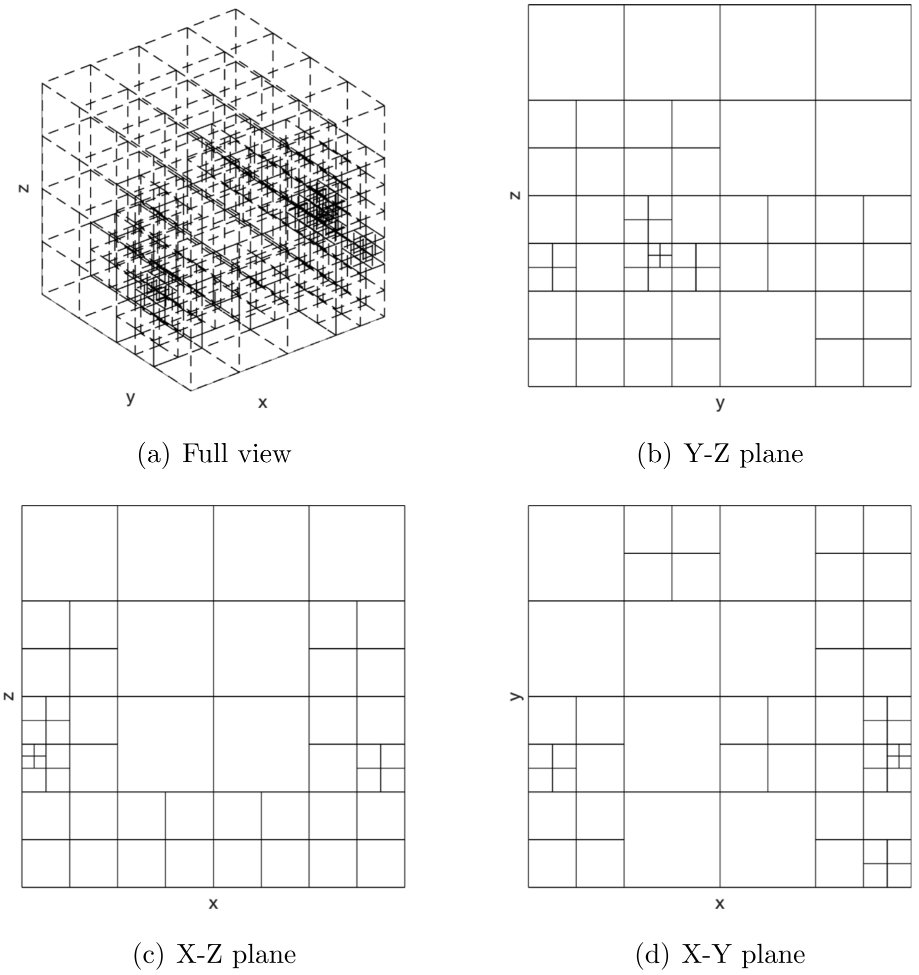
Table 3 Simulation CPU time(s) of IMEX-RK and SDC methods based on the uniform 64 × 64 mesh

k	IMEX-RK	SDC
1	12.14	14.67
2	96.76	193.80

is constructed to satisfy (1) with $d = 3$ and an additional source term. We compute the numerical solution on $[0,1]^3$ with periodic boundary conditions and Reynolds number $Re_\infty = 200$ till final time $T = 0.1$. The series of background meshes are composed of $N \times N \times N$ hexahedral elements with $N = 4, 8, 16, 32$, see Fig. 3 for the 3D adaptive sample mesh. We also compute the L^2 errors and orders of density, velocity and pressure for IMEX-RK-LDG and SDC-LDG schemes. As in the previous case, Tables 4 and 5 show that our adaptive methods can also achieve optimal order of accuracy for 3D NS equations and nearly the same accuracy is obtained for the two schemes with identical setup.

Example 3 ***Efficiency test of h-adaptive methods***

For the purpose of illustrating the superiority of the presented mesh adaptation technique compared with the uniform mesh method in terms of storage and computational time, we consider the evolution of an isentropic vortex in inviscid flows. This procedure is governed by the 2D Euler equations, which are obtained by neglecting the viscous effects and heat conduction, that is, omitting the right-hand side of Eqs. (1). The initial solution of this problem is obtained by adding some perturbations to the uniform mean flow



(a) Full view

(b) Y-Z plane

(c) X-Z plane

(d) X-Y plane

Fig. 3 Sample mesh for accuracy test in three dimension

Table 4 Accuracy test for adaptive IMEX-RK-LDG scheme of 3D NS equations

k	N	$\|\rho - \rho_h\|_{L^2}$	Order	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Order	$\|p - p_h\|_{L^2}$	Order
1	4	1.76E-01	–	2.37E-01	–	1.50E-01	–
	8	4.61E-02	1.93	6.37E-02	1.90	3.49E-02	2.10
	16	9.07E-03	2.34	1.34E-02	2.25	6.77E-03	2.37
	32	1.81E-03	2.31	3.00E-03	2.15	1.44E-03	2.23
2	4	5.00E-02	–	7.46E-02	–	4.70E-02	–
	8	7.73E-03	2.69	1.02E-02	2.86	6.35E-03	2.88
	16	8.06E-04	3.26	1.16E-03	3.13	7.35E-04	3.11
	32	9.26E-05	3.12	1.40E-04	3.05	8.98E-05	3.03

$\rho_\infty = u_\infty = v_\infty = p_\infty = 1$. The vortex perturbations of the temperature T, velocity field (u, v) and entropy S can be formulated as follows:

$$\delta T = -\frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad (\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} (y_0 - y, x - x_0), \quad \delta S = 0,$$

where $r^2 = (x - x_0)^2 + (y - y_0)^2$, (x_0, y_0) is the coordinate of the vortex center and ϵ is the vortex strength. Then the initial solution is determined through isentropic relation

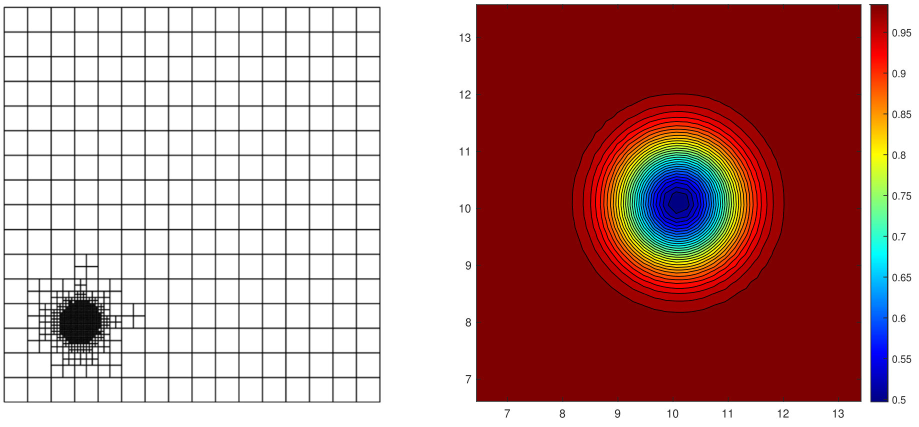
$$\rho = (1 + \delta T)^{\frac{1}{\gamma-1}}, \quad u = 1 + \delta u, \quad v = 1 + \delta v, \quad p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}}.$$

The exact solution to this problem is simply the passive convection of the initial solution. We take $(x_0, y_0) = (10, 10)$, $\epsilon = 5$ and computational domain as $[0, 50]^2$. The numerical solutions are computed using piecewise quadratic polynomials and third order SSP-RK method which is more suitable for hyperbolic conservation laws. For more details about SSP-RK methods, see [2–5]. To evaluate the efficiency improvement of the h-adaptive methods, we run the simulations both in uniform and adaptive meshes with Dirichlet boundary conditions till time $T = 0.1$. For the adaptive case, a combination of indicators η_{gi} and η_{ci} with parameters $\omega_1 = 1.2$, $\omega_2 = 0.3$, $\omega_3 = 1.3$ and $\omega_4 = 0.4$ are employed, and the depth of the tree is $LEV = 4$. The final adaptive mesh based on a 16×16 background mesh and the density distribution around the vortex are presented in Fig. 4. The computed L^2 errors of density as functions of the final number of elements in the mesh and CPU time are shown in Fig. 5, illustrating that the h-adaptive methods can use less degrees of freedom and CPU time when achieving the same L^2 error. Table 6 quantitatively shows that our h-adaptive methods can save 89.9% in storage and 70.4% in CPU time compared with the uniform mesh method when they achieve the same L^2 error 4.84E-6 of density.

Example 4 ***Shock-tube problems***

Table 5 Accuracy test for adaptive SDC-LDG scheme of 3D NS equations

k	N	$\|\rho - \rho_h\|_{L^2}$	Order	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Order	$\|p - p_h\|_{L^2}$	Order
1	4	1.76E-01	–	2.37E-01	–	1.50E-01	–
	8	4.61E-02	1.93	6.37E-02	1.89	3.50E-02	2.10
	16	9.07E-03	2.34	1.34E-02	2.24	6.77E-03	2.37
	32	1.81E-03	2.31	3.00E-03	2.15	1.44E-03	2.23
2	4	4.98E-02	–	7.39E-02	–	4.67E-02	–
	8	7.74E-03	2.68	1.02E-02	2.84	6.35E-03	2.87
	16	8.06E-04	3.26	1.16E-03	3.13	7.35E-04	3.11
	32	9.26E-05	3.12	1.40E-04	3.05	8.98E-05	3.03



(a) (b)

Fig. 4 (**a**) The final adaptive mesh based on a 16 × 16 background mesh. (**b**) Density distribution around the vortex

To explicitly show the shock-capturing property of our proposed methods, we simulate the Riemann problems which are usually taken as Euler benchmark problems directly using the NS equations. Specifically, the NS Eqs. (1) with $d = 1$ and initial conditions

$$\mathbf{U}(x,0) = \begin{cases} U_L, & x < 0, \\ U_R, & x > 0, \end{cases}$$

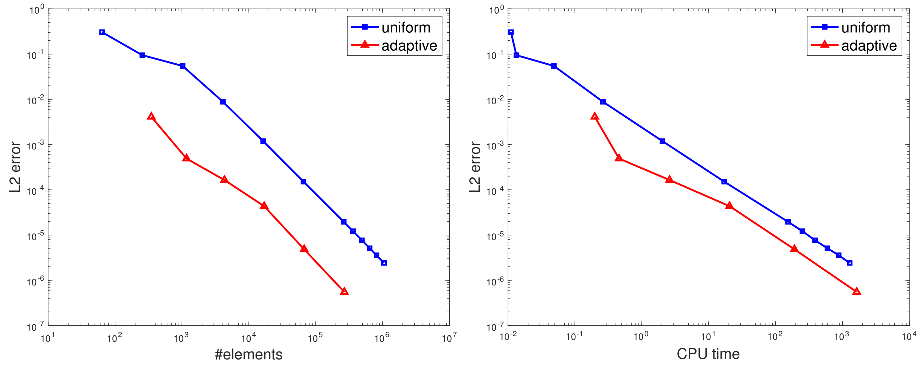
are considered in this test case. We adopt the classical Sod and Lax initial settings:

$$(\rho_L, u_L, p_L) = (1, 0, 1), \quad (\rho_R, u_R, p_R) = (0.125, 0, 0.1), \tag{18}$$

and

$$(\rho_L, u_L, p_L) = (0.445, 0.698, 3.528), \quad (\rho_R, u_R, p_R) = (0.125, 0, 0.1). \tag{19}$$

We run the two simulations with Reynolds number $Re_\infty = 1800$, P^2 element, third order IMEX-RK methods as well as a combination of indicators η_{gi} and η_{di} with parameters $\omega_1 = 1.2$, $\omega_2 = 0.3$, $\omega_3 = 1.2$, and $\omega_4 = 0.3$ on a background mesh of 200 cells till



(a) (b)

Fig. 5 L^2 errors of density as functions of the final number of elements and CPU time(s) on uniform and adaptive meshes for the evolution of an isentropic vortex. (**a**) L^2 error versus #elements; (**b**) L^2 error versus CPU time(s)

Table 6 Comparison of uniform and adaptive meshes with respect to number of elements and CPU time(s) when achieving 4.84E-6 L^2 error of density for the evolution of an isentropic vortex

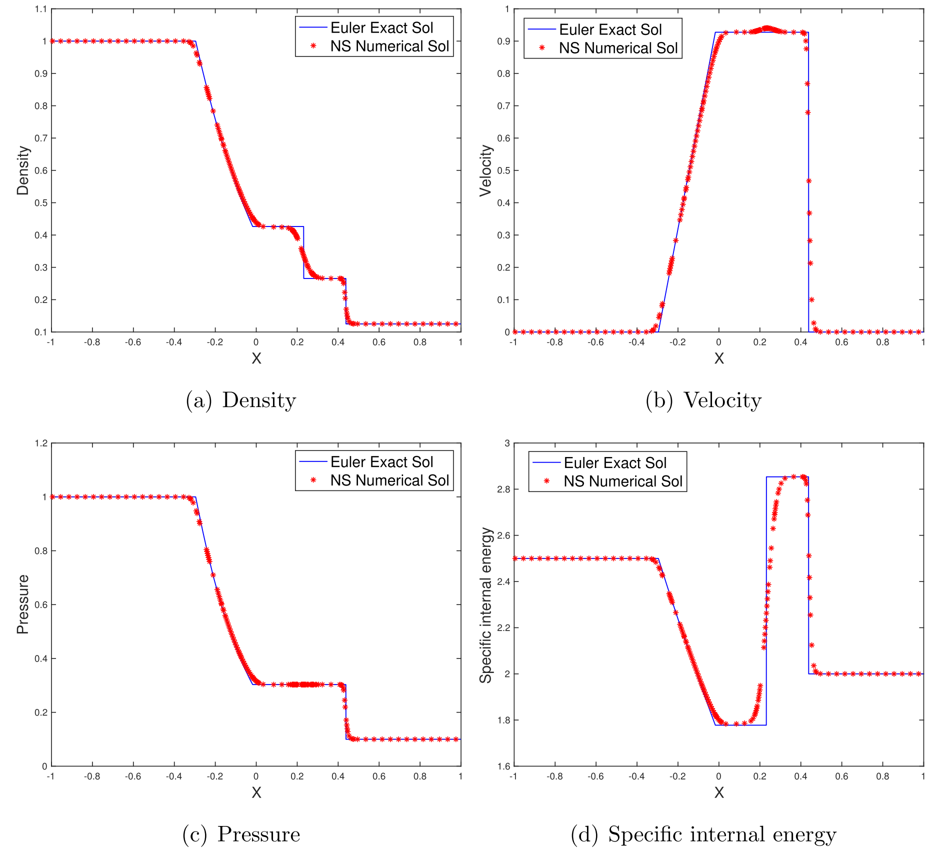
	Uniform mesh	**Adaptive mesh**	**Percentage of savings**
#elements	669176	67195	89.9%
CPU time(s)	645	191	70.4%

time $T = 0.25$ and $T = 0.13$, respectively. Figures 6 and 7 show the comparisons of density, velocity, pressure, specific internal energy between the Euler exact solutions and the NS numerical solutions. Note, the points for the NS numerical solutions are plotted every four cells and more points are clustered near the shock waves, contact discontinuities and rarefaction waves due to mesh adaptation to get better resolution.

Example 5 ***Taylor-Green vortex***

We consider the Taylor-Green vortex, which is a low Mach number problem, in this test case. The initial solution of this problem is

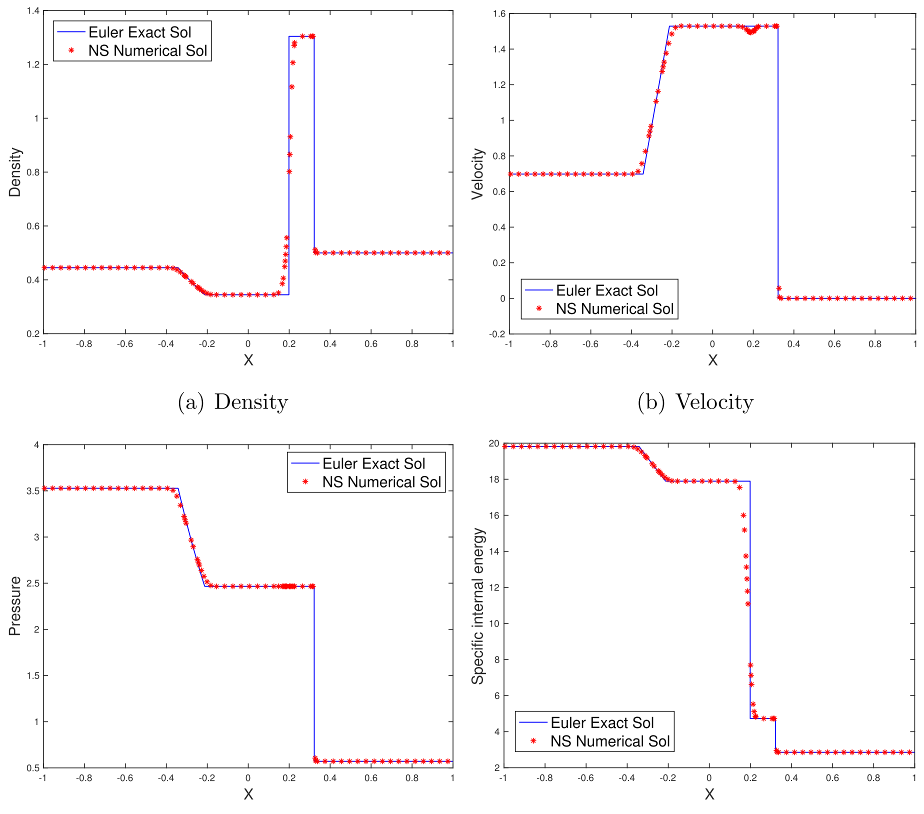
$$\begin{aligned}
&\rho(\mathbf{x},0) = 1,\\
&u(\mathbf{x},0) = \sin(x)\cos(y),\\
&v(\mathbf{x},0) = -\cos(x)\sin(y),\\
&p(\mathbf{x},0) = \frac{p_0}{\gamma-1} + \frac{1}{4}(\cos(2x)+\cos(2y)),
\end{aligned}$$



(a) Density (b) Velocity

(c) Pressure (d) Specific internal energy

Fig. 6 Sod shock-tube problem, $Re_\infty = 1800$, P^2



(a) Density (b) Velocity

(c) Pressure (d) Specific internal energy

Fig. 7 Lax shock-tube problem, $Re_\infty = 1800$, P^2

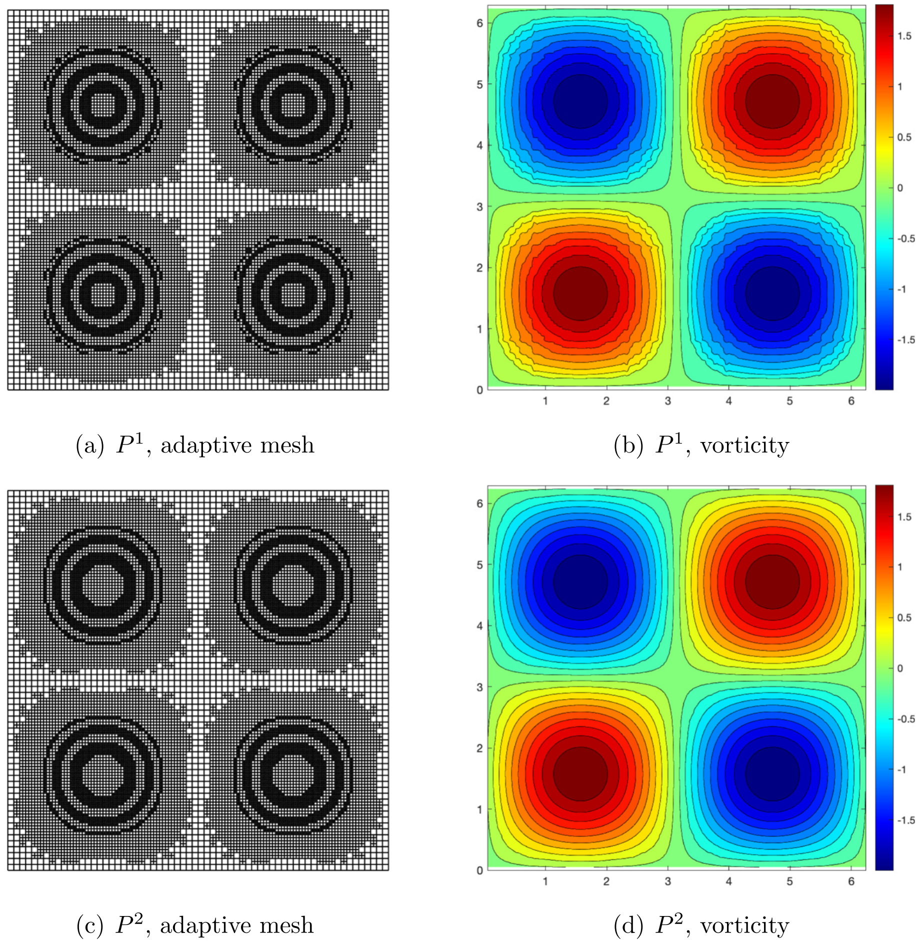
with $p_0 = 10^5$. This configuration leads to a characteristic Mach number $M \approx 1.7 \cdot 10^{-3}$, thus corresponds to the low Mach number regime [27]. The computational domain is $[0, 2\pi]^2$. We simulate this procedure on a 64×64 background mesh with periodic boundary conditions till final time $T = 0.1$. The $(k+1)$-th order IMEX-RK time discretization will be employed when P^k element is used. A single adaptive indicator η_{ci} with parameters $\omega_1 = 1.5$, $\omega_2 = 0.5$ is adopted. As is shown in Fig. 8, the vortex can be accurately captured by the adaptive mesh, and the contours become smoother when higher order elements are employed.

Example 6 *Subsonic flow around a NACA0012 airfoil*

In this test case we consider a subsonic flow around a NACA0012 airfoil which is symmetric about the x-axis with upper and lower surfaces described by functions [34]

$$y^{\pm}(x) = \pm 5 \times 0.12 \times (0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4),$$
$$0 \le x \le 1.008930411365,$$

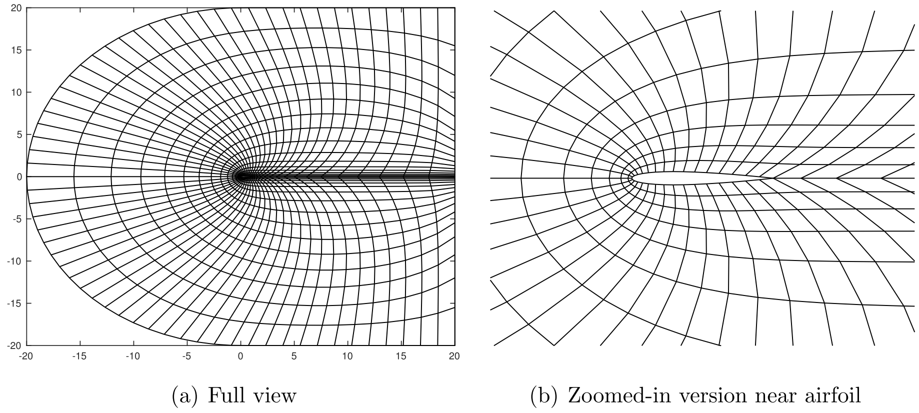
and we require to rescale $y^{\pm}$ so as to gain an airfoil of unit chord length. A rather coarse 64×16 C-type background mesh which extends about 20 chord length away from the airfoil is adopted for computation, see Fig. 9 for an impression of the mesh. The subsonic flow has configuration of Mach number $M_\infty = 0.5$, angle of attack $\alpha = 0°$ and Reynolds



(a) P^1, adaptive mesh (b) P^1, vorticity

(c) P^2, adaptive mesh (d) P^2, vorticity

Fig. 8 Taylor-Green vortex. Adaptive mesh and vorticity contours with 20 equally spaced contour lines from -1.998 to 1.998



(a) Full view (b) Zoomed-in version near airfoil

Fig. 9 Background mesh for NACA0012 airfoil

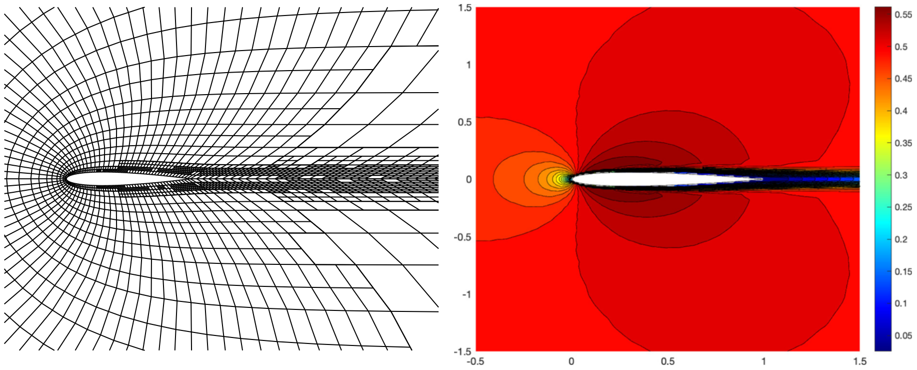
number $Re_\infty = 5000$. We apply an adiabatic no-slip wall boundary condition along the airfoil, subsonic inflow and outflow boundary conditions at respective farfield boundaries. We employ also a combination of indicators η_{gi} and η_{di} with parameters $\omega_1 = 1.2$, $\omega_2 = 0.3$, $\omega_3 = 1.2$, and $\omega_4 = 0.3$. P^2 element as well as the first order IMEX-RK time discretization is used since this is a steady state simulation. The adaptive mesh and the computed Mach number contours are presented in Fig. 10, clearly showing that the elements along and behind the wake of the airfoil can be locally refined to capture the more complex flow field. The distribution of pressure coefficient along the airfoil wall is further displayed in Fig. 11 and is consistent with the results presented in [6, 10].

Example 7 ***Supersonic flow past a NACA0012 airfoil***

In this test case, the profile of the airfoil and the initial background mesh remain the same as in the previous subsonic case. The supersonic flow with Mach number $M_\infty = 2$ passes through the computational domain at an angle of attack $\alpha = 10°$ and Reynolds number $Re_\infty = 106$. An adiabatic no-slip wall boundary condition is imposed on the airfoil and at the farfield part of the computational domain, supersonic/subsonic inflow and supersonic/subsonic outflow boundary conditions are assumed, respectively, based on the uniform freestream and the unit normal vectors to the outer boundary. A salient feature of this case is the formation of a bow shock in front of the profile. We will also apply a combination of indicators η_{gi} and η_{di} with parameters $\omega_1 = 1.2$, $\omega_2 = 0.3$, $\omega_3 = 1.2$ and $\omega_4 = 0.3$. Again owing to steady state simulation, P^2 element as well as the first order IMEX-RK time discretization is employed. Figure 12 shows the adaptive mesh, Mach number and pressure contours of the numerical solution. Evidently, the bow shock can be captured with a nice resolution by virtue of local grid refinement. The pressure distribution is also given in Fig. 13.

Example 8 ***Rayleigh-Taylor instability***

In order to show the capability of our methods to capture the small-scale features of complicated flow structures, we consider the Rayleigh-Taylor instability phenomenon,



(a) Adaptive mesh (b) Mach contours

Fig. 10 (**a**) Adaptive mesh and (**b**) the contour plots corresponding to P^2 solution for subsonic flow around a NACA0012 airfoil, 30 equally spaced contour lines are used to plot Mach number from 0.025 to 0.580

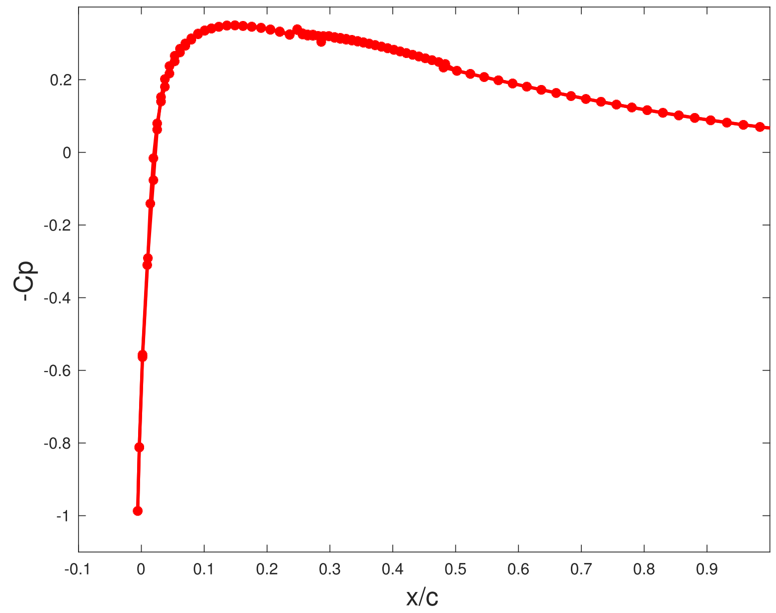
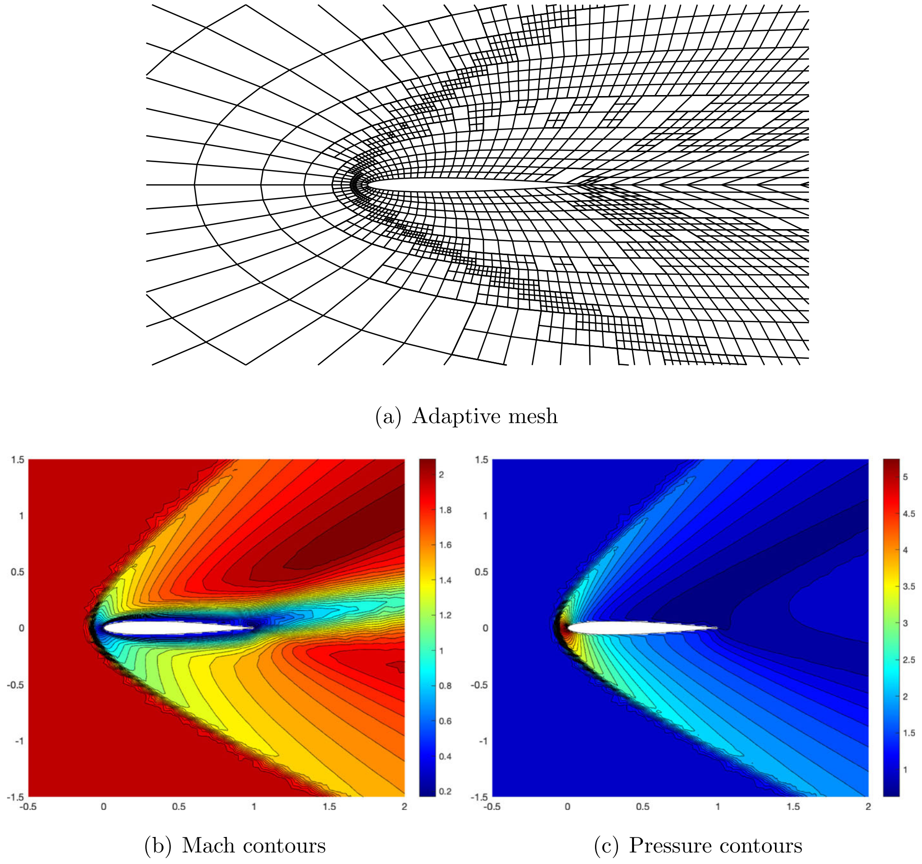


Fig. 11 Pressure coefficient for subsonic flow around a NACA0012 airfoil



(a) Adaptive mesh

(b) Mach contours

(c) Pressure contours

Fig. 12 (**a**) Adaptive mesh and the contour plots corresponding to P^2 solution for supersonic flow past a NACA0012 airfoil, 30 equally spaced contour lines are used to plot (**b**) Mach number from 0.15 to 2.15, and (**c**) pressure from 0.61 to 5.41

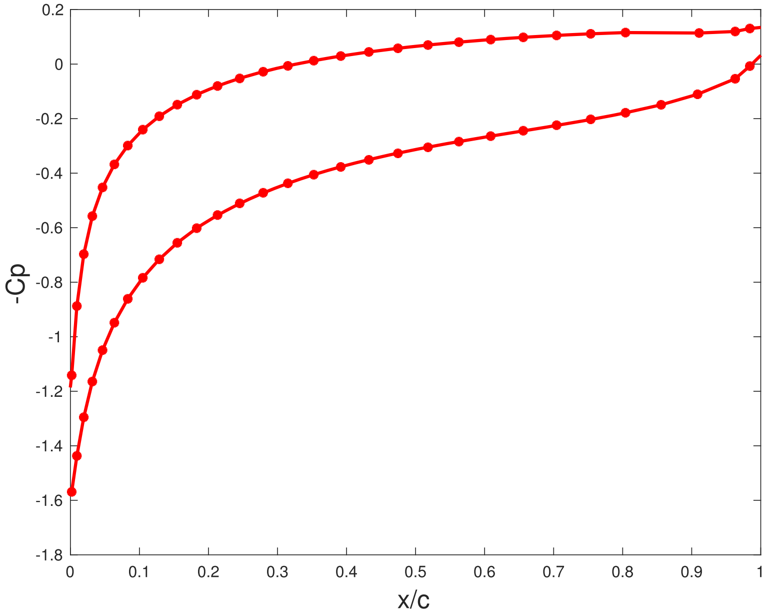


Fig. 13 Pressure coefficient for supersonic flow past a NACA0012 airfoil

which usually occurs at an interface between two fluids with different densities when the acceleration is directed from the heavy fluid to the light fluid, see [35, 36] for a detailed description of this phenomenon. We take the computational domain as $[0, 0.25] \times [0, 1]$. The initial solution is

$$(\rho, u, v, p) = \begin{cases} (2, 0, -0.025a\cos(8\pi x), 1 + 2y), & y \in (0, 0.5), \\ (1, 0, -0.025a\cos(8\pi x), y + 1.5), & y \in [0.5, 1), \end{cases}$$

where $a = \sqrt{\frac{\gamma p}{\rho}}$ is the sound speed with ratio of specific heats $\gamma = \frac{5}{3}$. The Prandtl number is taken as $Pr = 0.7$ for this test case. An adiabatic, reflective boundary condition is imposed for the left and right boundaries. At the bottom and top boundaries, the flow values are set as

$$(\rho, u, v, p) = \begin{cases} (2, 0, 0, 1), & y = 0, \\ (1, 0, 0, 2.5), & y = 1, \end{cases}$$

respectively. The source term ρ is added to the right hand side of the second momentum equation and ρv is added to the right hand side of the energy equation in (1). All the simulations are based on a 30×120 background mesh till final time $T = 1.95$, and the $(k + 1)$-th order IMEX-RK time discretization will be employed when P^k element is used. For the NS simulations, we adopt a single indicator η_{gi} with parameters $\omega_1 = 1.5$, $\omega_2 = 0.5$. Computational results with Reynolds number 50000 and 150000 are shown in Figs. 14 and 15, respectively. As can be seen, the interface between two fluids with different densities becomes thinner as the Reynolds number increases. Our adaptive methods can capture the interface and small-scale features precisely, and the resolution of P^2 element can even be comparable to that obtained with the ninth-order WENO scheme on a 600×2400 uniform mesh in [36]. The adaptive meshes and density contours of the Euler simulations with TVB limiter [3] and without TVB limiter are also presented in Figs. 16 and 17 by using adaptive parameters $\omega_1 = 1.2$, $\omega_2 = 0.3$. Due to the absence of physical

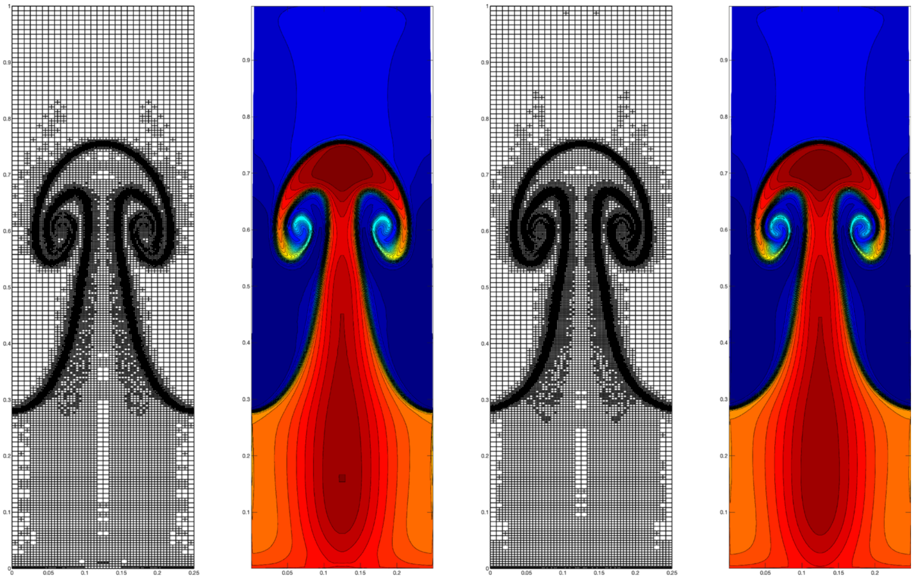


Fig. 14 NS simulations for Rayleigh-Taylor instability with Reynolds number $Re_\infty = 50000$. From left to right: adaptive mesh for P^1 element; 30 equally spaced density contours from 0.8728 to 2.2249 for P^1 element; adaptive mesh for P^2 element; 30 equally spaced density contours from 0.8733 to 2.2262 for P^2 element

viscosities, there exist more small-scale flow structures in these simulations, and the resolution of P^2 element can be comparable to that obtained with the ninth-order WENO scheme on a 480 × 1920 uniform mesh in [35].

7 Conclusions

In this paper, we have presented an *h*-adaptive LDG method for the solution of the unsteady compressible NS equations. To achieve the uniform high order accuracy both in space and time, avoid the extremely small time step restriction of explicit methods as

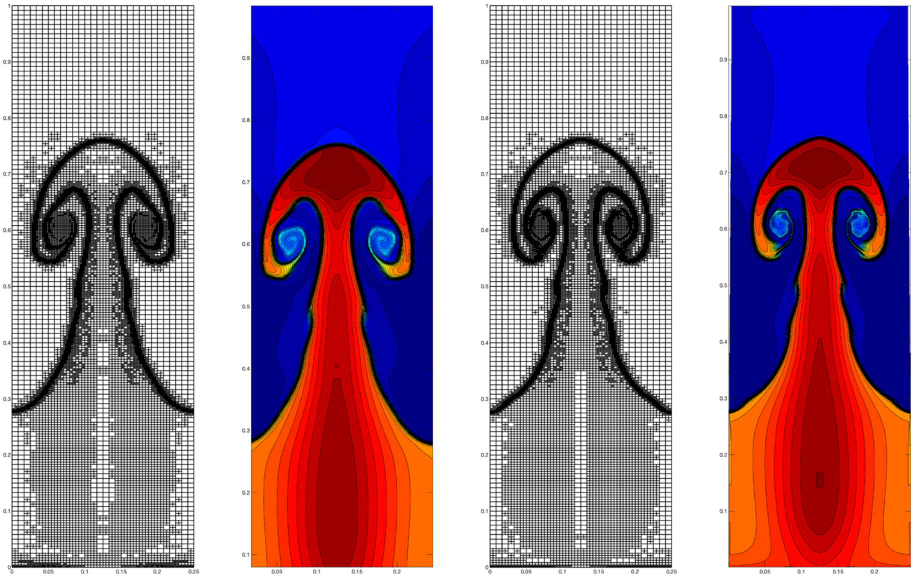


Fig. 15 NS simulations for Rayleigh-Taylor instability with Reynolds number $Re_\infty = 150000$. From left to right: adaptive mesh for P^1 element; 30 equally spaced density contours from 0.8608 to 2.2316 for P^1 element; adaptive mesh for P^2 element; 30 equally spaced density contours from 0.8712 to 2.2301 for P^2 element

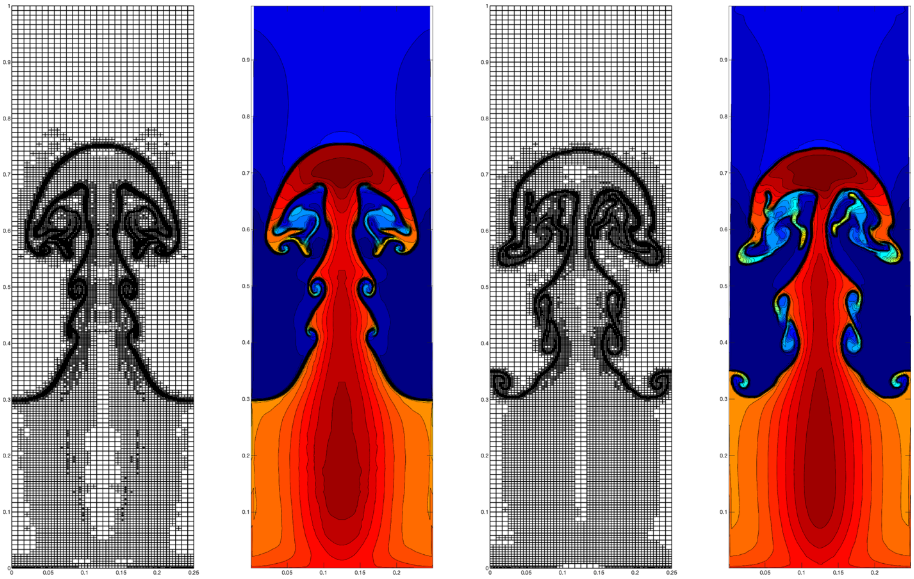


Fig. 16 Euler simulations for Rayleigh-Taylor instability with TVB limiter. From left to right: adaptive mesh for P^1 element; 30 equally spaced density contours from 0.8620 to 2.2407 for P^1 element; adaptive mesh for P^2 element; 30 equally spaced density contours from 0.8705 to 2.2362 for P^2 element

well as the construction of large Jacobian matrix in implicit methods, two semi-implicit time marching methods including the classical IMEX-RK methods and the SDC methods were also employed for temporal discretizations. In the process of mesh adaptation, indicators are computed in each element of the mesh based on the gradient of density, the divergence and curl of velocity field to pick candidate elements for refinement and coarsening. We also adopt the strategy of "refinement must, coarsening can" to determine the ultimate element lists for refinement and coarsening. These proposed methods have been

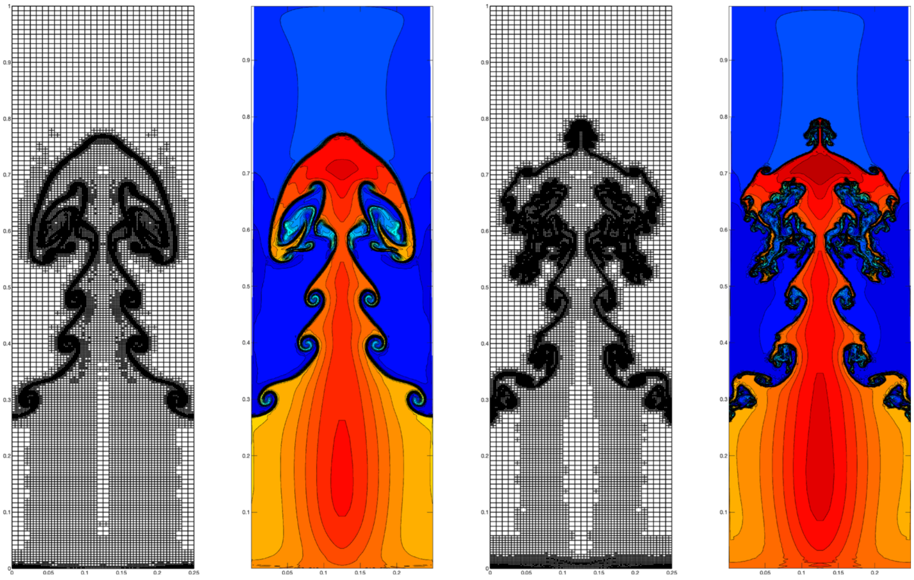


Fig. 17 Euler simulations for Rayleigh-Taylor instability without TVB limiter. From left to right: adaptive mesh for P^1 element; 30 equally spaced density contours from 0.6548 to 2.4582 for P^1 element; adaptive mesh for P^2 element; 30 equally spaced density contours from 0.6852 to 2.3728 for P^2 element

successfully implemented in a sequence of numerical examples, illustrating their expected optimal order of convergence, efficiency and capabilities in flow problems. Due to the high order accuracy and easy implementation, the h-adaptive LDG methods coupled with IMEX-RK or SDC methods provide an appealing alternative for unsteady compressible NS equations which require to accurately capture the detailed flow structures at any time.

Acknowledgements
Not applicable.

Authors' contributions
Xiangyi Meng: Formal analysis, Investigation, Methodology, Software, Writing-original draft, Writing-review & editing. Yan Xu: Conceptualization, Methodology, Funding acquisition, Resources, Supervision, Writing-review & editing. The authors read and approved the final manuscript.

Funding
Research of Yan Xu is supported by the National Numerical Windtunnel Project NNW2019ZT4-B08, NSFC grant No. 12071455.

Availability of data and materials
The datasets generated during the current study are available from the corresponding author on reasonable request. They support our published claims and comply with field standards.

Declarations

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 21 December 2021 Accepted: 1 March 2022
Published online: 01 June 2022

References

1. Reed WH, Hill TR (1973) Triangular mesh methods for the neutron transport equation. Tech Rep LA-UR-73-479. Los Alamos Scientific Lab, New Mexico
2. Cockburn B, Hou S, Shu C-W (1990) The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. Math Comput 54(190):545–581
3. Cockburn B, Lin S-Y, Shu C-W (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. J Comput Phys 84(1):90–113
4. Cockburn B, Shu C-W (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II. General framework. Math Comput 52(186):411–435
5. Cockburn B, Shu C-W (1998) The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J Comput Phys 141(2):199–224
6. Bassi F, Rebay S (1997) A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. J Comput Phys 131(2):267–279
7. Bassi F, Crivellini A, Rebay S, Savini M (2005) Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and k–ω turbulence model equations. Comput Fluids 34(4-5):507–540
8. Hartmann R, Houston P (2006) Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation. Int J Numer Anal Model 3(1):141–162
9. Klaij CM, van der Vegt JJ, van der Ven H (2006) Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations. J Comput Phys 217(2):589–611
10. Luo H, Luo L, Nourgaliev R, Mousseau VA, Dinh N (2010) A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids. J Comput Phys 229(19):6961–6978
11. Liu H, Yan J (2009) The direct discontinuous Galerkin (DDG) methods for diffusion problems. SIAM J Numer Anal 47(1):675–698
12. Cheng J, Yang X, Liu X, Liu T, Luo H (2016) A direct discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids. J Comput Phys 327:484–502
13. Cockburn B, Shu C-W (1998) The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J Numer Anal 35(6):2440–2463
14. Xu Y, Shu C-W (2010) Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. Commun Comput Phys 7(1):1–46
15. Tian L, Xu Y, Kuerten JG, van der Vegt JJ (2016) An h-adaptive local discontinuous Galerkin method for the Navier–Stokes–Korteweg equations. J Comput Phys 319:242–265
16. Liu J, Qiu J, Goman M, Li X, Liu M (2016) Positivity-preserving Runge-Kutta discontinuous Galerkin method on adaptive Cartesian grid for strong moving shock. Numer Math Theory Methods Appl 9(1):87–110
17. Liu J, Qiu J, Hu O, Zhao N, Goman M, Li X (2013) Adaptive Runge–Kutta discontinuous Galerkin method for complex geometry problems on Cartesian grid. Int J Numer Methods Fluids 73(10):847–868

18. Pareschi L, Russo G (2005) Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. J Sci Comput 25(1):129–155
19. Wang H, Shu C-W, Zhang Q (2016) Stability analysis and error estimates of local discontinuous Galerkin methods with implicit–explicit time-marching for nonlinear convection–diffusion problems. Appl Math Comput 272:237–258
20. Wang H, Wang S, Zhang Q, Shu C-W (2016) Local discontinuous Galerkin methods with implicit-explicit time-marching for multi-dimensional convection-diffusion problems. ESAIM: Math Model Numer Anal 50(4):1083–1105
21. Xia Y, Xu Y, Shu C-W (2007) Efficient time discretization for local discontinuous Galerkin methods. Discret Contin Dyn Syst-B 8(3):677–693
22. Guo R, Filbet F, Xu Y (2016) Efficient high order semi-implicit time discretization and local discontinuous Galerkin methods for highly nonlinear PDEs. J Sci Comput 68(3):1029–1054
23. Guo R, Xia Y, Xu Y (2017) Semi-implicit spectral deferred correction methods for highly nonlinear partial differential equations. J Comput Phys 338:269–284
24. Boscarino S, Russo G, Scandurra L (2018) All Mach number second order semi-implicit scheme for the Euler equations of gas dynamics. J Sci Comput 77(2):850–884
25. Boscheri W, Dimarco G, Loubère R, Tavelli M, Vignal M-H (2020) A second order all Mach number IMEX finite volume solver for the three dimensional Euler equations. J Comput Phys 415:109486
26. Boscheri W, Pareschi L (2021) High order pressure-based semi-implicit IMEX schemes for the 3D Navier-Stokes equations at all Mach numbers. J Comput Phys 434:110206
27. Busto S, Río-Martín L, Vázquez-Cendón ME, Dumbser M (2021) A semi-implicit hybrid finite volume/finite element scheme for all Mach number flows on staggered unstructured meshes. Appl Math Comput 402:126117
28. Busto S, Tavelli M, Boscheri W, Dumbser M (2020) Efficient high order accurate staggered semi-implicit discontinuous Galerkin methods for natural convection problems. Comput Fluids 198:104399
29. Fambri F, Dumbser M (2017) Semi-implicit discontinuous Galerkin methods for the incompressible Navier–Stokes equations on adaptive staggered Cartesian grids. Comput Methods Appl Mech Eng 324:170–203
30. Tavelli M, Dumbser M (2017) A pressure-based semi-implicit space–time discontinuous Galerkin method on staggered unstructured meshes for the solution of the compressible Navier–Stokes equations at all Mach numbers. J Comput Phys 341:341–376
31. Toro EF (2009) Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer, Berlin, Heidelberg
32. Ascher UM, Ruuth SJ, Spiteri RJ (1997) Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl Numer Math 25(2-3):151–167
33. Calvo M, De Frutos J, Novo J (2001) Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations. Appl Numer Math 37(4):535–549
34. Bristeau MO, Glowinski R, Periaux J, Viviand H (1987) Numerical simulation of compressible Navier-Stokes flows: a GAMM workshop. Vieweg+Teubner Verlag, Braunschweig
35. Shi J, Zhang Y-T, Shu C-W (2003) Resolution of high order WENO schemes for complicated flow structures. J Comput Phys 186(2):690–696
36. Zhang Y-T, Shi J, Shu C-W, Zhou Y (2003) Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high Reynolds numbers. Phys Rev E 68(4):046709

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.