Received: 23 October 2018 | Revised: 31 May 2019 | Accepted: 14 June 2019
DOI: 10.1002/fld.4757

RESEARCH ARTICLE

WILEY

A high-order Runge-Kutta discontinuous Galerkin method with a subcell limiter on adaptive unstructured grids for two-dimensional compressible inviscid flows

Pritam Giri | Jianxian Qiu

Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, School of Mathematical Sciences, Xiamen University, Xiamen 361005, China

Correspondence
Jianxian Qiu, Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, School of Mathematical Sciences, Xiamen University, Xiamen 361005, China.
Email: jxqiu@xmu.edu.cn

Funding information
NSAF, Grant/Award Number: U1630247; National Natural Science Foundation of China, Grant/Award Number: 11571290

Summary

A robust, adaptive unstructured mesh refinement strategy for high-order Runge-Kutta discontinuous Galerkin method is proposed. The present work mainly focuses on accurate capturing of sharp gradient flow features like strong shocks in the simulations of two-dimensional inviscid compressible flows. A posteriori finite volume subcell limiter is employed in the shock-affected cells to control numerical spurious oscillations. An efficient cell-by-cell adaptive mesh refinement is implemented to increase the resolution of our simulations. This strategy enables to capture strong shocks without much numerical dissipation. A wide range of challenging test cases is considered to demonstrate the efficiency of the present adaptive numerical strategy for solving inviscid compressible flow problems having strong shocks.

KEYWORDS

adaptive mesh refinement, Euler equations, HWENO method, Runge-Kutta discontinuous Galerkin method, subcell limiter, WENO method

1 | INTRODUCTION

The main purpose of this paper is to develop an accurate and robust numerical method for solving a hyperbolic system of partial differential equations (PDEs) using high-order Runge-Kutta discontinuous Galerkin (RKDG) method with an adaptive mesh refinement (AMR) strategy on two-dimensional (2-D) unstructured grids. Designing a high-order accurate and robust numerical method for hyperbolic PDEs is a difficult problem because hyperbolic equations admit discontinuous solution. This leads to spurious oscillations in the computed solution when approximated by high-degree polynomials. The discontinuous Galerkin (DG) spectral element method (SEM) is one of the most promising available methods to bring robustness and accuracy together to numerically solve hyperbolic problems. Reed and Hill[1] first introduced the DG method to simulate neutron transport phenomena. After that, Cockburn et al laid the profound theoretical background of DG method with Runge-Kutta time integration in their series of well-known papers.[2-6] The most attractive feature of the DG method is that the continuity of approximations for the solution across the faces of an element need not to be assured. As an immediate consequence of that, the approximation over any element does not depend on approximations of neighboring elements, making this method compact and that is why highly parallelizable on a high-performance computing platform. Moreover, such an approximation of solution space, being quite consistent with the admissible solution space for a hyperbolic system of equations, provides a possible way of increasing the order of spatial accuracy of this method by increasing the degree of approximating polynomials.

Designing a suitable limiter plays a crucial role in the success of a high-order RKDG scheme for simulating compressible flows. At the same time, designing a compact limiter in the DG framework is of prime interest to preserve the compactness of the overall method. The compactness of a limiter is also important for an easy integration to the AMR strategy. The slope limiting methodology is one of the most primitive attempts to control spurious oscillations in the computed solutions provided by high-order DG methods. The *minmod*-type limiters,[3-6] the moment-based limiter as proposed by Biswas et al,[7] and an improved moment-based limiter by Burbeau et al[8] are few but not a complete list of limiting techniques that rely on the slope limiting methodology. Such kind of limiting strategies are prone to degrade the order of accuracy of the high-order RKDG if limiting is mistakenly applied to at least a few cells where the solution is reasonably smooth. However, the weighted essentially nonoscillatory (WENO) scheme–based limiting methodology for the DG method along with an appropriate *troubled cell* indicator is relatively more successful. The WENO schemes[9,10] first appeared as a high-order reconstruction strategy in the finite difference and the finite volume methods. Essentially, these schemes adaptively choose the smoothest reconstruction at a cell interface point out of all possible stencils of specified order to avoid spurious oscillations. The WENO-based reconstruction strategy comprises the basic ingredient of the DG limiter in the work of Qiu and Shu[11] and Zhu et al.[12] Such kind of a limiting strategy shows an enormous possibility while keeping the order of accuracy of the DG scheme intact. Similar to the classical WENO method, the Hermite WENO (HWENO)–based reconstruction strategy also serves the same purpose in reconstructing a spurious oscillation-free solution in the finite difference or the finite volume method. Additionally, the HWENO-based reconstruction enables one to achieve the same order of accuracy on a relatively more compact stencil in comparison to the classical WENO-based reconstruction. Qiu and Shu[13,14] first proposed the use of the HWENO-based reconstruction as a limiter for high-order DG methods on the Cartesian grids to simulate one-dimensional and 2-D inviscid compressible flows. Zhu and Qiu[15,16] extended this study to simulate 2-D and three-dimensional flows using unstructured simplex elements and demonstrated the efficacy of the WENO and the HWENO-based limiting strategy for the RKDG. Despite the clear superiority of the WENO-type limiters over available slope limiting methodologies, the WENO-type limiters also have a few drawbacks. Reconstruction of polynomial on any target cell using such limiters requires information not only from the immediate neighbors but also from the neighbors of the next neighbors. This destroys the inherent compactness of the unlimited RKDG methods. The compactness of the DG method is one of the most appealing features that makes this method suitable for parallel computing. However, in addition to the loss of compactness, it is common to encounter the problem of negative linear weights while using such WENO-type limiters, and this problem needs special treatment to get rid of subsequent difficulties (interested readers may refer to the review paper by Shu[17] and references therein). Zhong and Shu[18] first proposed a simple as well as compact WENO-based limiting strategy on the Cartesian mesh, which requires the information from the immediate neighbors of a target cell, and at the same time, this scheme avoids the problem of negative linear weights. Zhu et al[19] extended this study to illustrate the performance of a simple and compact WENO-based limiting strategy for an unstructured grid with triangular elements. Recently, Zhu et al proposed a simple and compact HWENO-based limiting technique for the RKDG method on the structured[20] as well as unstructured[21] grids. Recently, Dumbser et al[22,23] have developed an a posteriori subcell limiter that has a different philosophy than the so-called classical DG limiters. Instead of repairing the solution at any particular instant of time, a posteriori subcell limiters attempt to recompute the solution starting from a set of reliable subcell averages at the previous time step with a more robust numerical method. Those methods are attractive in many ways. First, unlike the classical limiters, a posteriori subcell limiters do not sacrifice the compactness of the underlying DG scheme, irrespective of its order. After-the-fact troubled cell detection and recompute philosophy make such a limiter more controllable, and that is why it is more failure-proof and robust. This is the key to integrate accuracy to the robustness of a high-order DG scheme. In our present work, we implemented a modified a posteriori subcell limiting strategy, which is discussed in Section 3.

An appropriate detection of troubled cells, on which limiting might be needed, is an important ingredient of a successful limiting strategy. The classical minmod-type total variation bounded (TVB) troubled cell indicators were adopted in the earliest related works of Cockburn et al.[3,5,6] The necessity of a problem-dependent positive constant (M) hinders the robustness of a minmod-type TVB limiter. Many other shock detection strategies or troubled cell indicators, some of which perform well for a special kind of hyperbolic PDEs in comparison to other candidates, appear in the DG literature. The moment limiter (BDF) of Biswas et al,[7] a modified moment limiter of Burbeau et al,[8] the monotonicity-preserving limiter of Suresh and Huynh,[24] a modified monotonicity-preserving limiter of Rider and Margolin,[25] and subcell resolution idea–based troubled cell indicator of Harten[26] are some of the interesting propositions to circumvent this problem. Interested readers may refer to the work of Qiu and Shu[27] for an extensive review and comparative study about the performances of various troubled cell indicators in the literature. A clear superiority of the minmod-type TVB limiter with a properly chosen TVB parameter, subcell resolution idea–based troubled cell indicator of Harten, and the shock detection technique (referred to as KXRCF hereafter) by Krivodonova et al[28] over other shock detection methodologies was reported by Qiu and Shu.[27] In the context of a posteriori finite volume subcell limiter, an after-the-fact check is a natural way to design a troubled cell indicator. This strategy is motivated by the so-called "multidimensional optimal order detection (MOOD) loop" in the finite volume framework as proposed by Clain et al[29] and extended to the cases for unstructured grids by Diot et al.[30] The MOOD-based limiting strategy was adopted by Dumbser et al.[22,23] In our present work, a similar strategy is adopted with a few modifications for their application with the AMR.

An admissible discontinuous solution across the cell interfaces is the basic essence and most prominent distinction of DG methods from the conventional continuous Galerkin methods. This privilege makes the algorithm highly parallelizable and favorable to AMR without having any restriction on the conformity of the adapted grids. The first patch-based nonconforming AMR was introduced by Berger and Oliger.[31] They employed a nested, logically rectangular patch-based AMR for second-order accurate finite difference and finite volume method[31,32] to solve for a hyperbolic system of equations. A fifth-order accurate WENO scheme along with the patch-based AMR method was proposed by Baeza and Mulet.[33] The cell-by-cell AMR is a slightly simpler version of mesh adaptation from an algorithmic point of view, and such methods can easily be extended for unstructured grids. The cell-by-cell AMR approach was introduced first by Khokhlov[34] in the finite volume framework. Later, this adaptation strategy was extended by Dumbser et al.[35,36] for the high-order accurate finite volume framework with a high-order accurate time integration scheme to solve hyperbolic systems in conservative as well as nonconservative forms. An algorithmic simplicity of the cell-by-cell AMR made this strategy well suited for a high-order DG method. Boumann and Oden[37] implemented the DG with a cell-by-cell adaptation for the convection-diffusion problem, and they extended this study for the compressible inviscid as well as compressible viscous flow simulations.[38] Houston et al.[39,40] analyzed the cell-by-cell mesh adaptation with hanging nodes for the first-order hyperbolic problems using DG methods. The cell-by-cell AMR approach was successfully extended to the unstructured meshes (see the work of Luo et al[41]). Recently, Fambri and Dumbser[42] extended the application of the cell-by-cell AMR strategy to solve incompressible viscous flow problems on the Cartesian grids. The cell-by-cell AMR strategy on the Cartesian grid along with the DG method was also implemented in our past work[43] to solve hyperbolic conservation laws.

Although there is no scope of any further dispute over the effectiveness of the AMR and high-order methods for hyperbolic problems, an efficient combination of those strategies remains challenging for solving real-world problems. In our present work, a fifth-order RKDG method on unstructured triangular grid elements is implemented. It is immensely difficult for conventional a priori limiters with a fifth-order RKDG to simulate flow problems with strong shocks. However, to our present implementation with a modified subcell limiter, a wide range of such problems remain accessible even with a fifth-order RKDG. The present implementation relies on a new extra set of solution admissibility criteria to detect the troubled cells. This is discussed in details in Section 3.3. Our numerical experiments show a substantial reduction in the troubled cell count for the simulations of flow problems with strong shocks. To the best of our knowledge, a fifth-order accurate RKDG method with an AMR on unstructured triangular grid elements for the compressible Euler equations has not appeared in the literature yet.

The rest of this paper is organized as follows. Section 2 outlines the spatial and time discretization for the Euler equations using a high-order RKDG method on triangular elements. Section 3 briefly describes the subcell limiter, as implemented in our present work, and the related projection operation for transferring data from a cell to its subcells and the reconstruction operations for transferring data from the subcells to the corresponding cell. This section also outlines the detection methodology for troubled cells. Section 4 describes the cell-by-cell AMR strategy suitable for our high-order RKDG on the triangular elements as implemented in the present work. Section 5 demonstrates the efficacy and robustness of our present implementation through a wide range of numerical experiments to simulate inviscid compressible flows with intricate flow features including strong shocks. Finally, Section 6 draws some final remarks on our present work.

2 | AN OVERVIEW OF THE RUNGE-KUTTA DISCONTINUOUS GALERKIN SPECTRAL ELEMENT METHOD FOR THE 2-D EULER EQUATIONS

We consider the time-dependent, 2-D, compressible Euler equations, conservation laws for mass, momentum, and total energy of the flow of an inviscid, compressible fluid, on a finite time interval $(0, T]$ and in an open, connected, bounded Lipschitz domain $\Omega \subset \mathbb{R}^2$. Those equations can be given by the following form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0} \qquad \text{in} \quad \Omega \times (0, T], \tag{1}$$

subject to the initial condition, $\mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x})$, where $\mathbf{x} \in \Omega$ and $t \in \mathbb{R}_+$ denote the spatial coordinate in the domain of interest and time variable, respectively. $\mathbf{u}(\mathbf{x}, t) : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}^4$ is the conservative vector and $\mathbf{F}(\mathbf{u}(\mathbf{x}, t)) : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}^{4\times 2}$ is a second-order tensor denoting the flux function, given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho E \end{bmatrix} \text{ and } \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho v_x & \rho v_y \\ \rho v_x^2 + p & \rho v_x v_y \\ \rho v_y v_x & \rho v_y^2 + p \\ (\rho E + p) v_x & (\rho E + p) v_y \end{bmatrix}. \tag{2}$$

In the previous expressions, ρ, (v_x, v_y), p, and E denote density of the fluid, velocity vector, fluid pressure, and total energy per unit mass, respectively. The total energy per unit mass can be determined from the knowledge of the internal energy per unit mass (e) by the expression, $E = e + \|\mathbf{v}\|^2/2$. In our present case, the system is closed by the equation of state $\rho e = p/(\gamma - 1)$ by assuming the fluid to be an ideal gas. γ denotes the ratio of specific heats.

Two-dimensional domains are partitioned by nonoverlapping, conforming triangular cells, which forms the primary mesh for 2-D simulations in our present work. The primary mesh is adaptively refined with nonconforming triangular elements as explained in details in Section 4. Next, the classical RKDG of order $(m + 1)$ approach will be outlined for the 2-D Euler equations. For a given conforming or nonconforming triangulation τ_h of the domain of interest, we seek for the approximate solution for the conservative vector $\mathbf{u}_h(\mathbf{x}, t)$ at any particular time in the finite element space of the discontinuous functions, given by

$$V_h^m := \left\{ v_h(\mathbf{x}) \in L^\infty(\Omega) : v_h|_T \in \mathbb{P}^m, \forall\, T \in \tau_h \right\}. \tag{3}$$

Here, T denotes any triangular element of the given triangulation τ_h, and $\mathbb{P}^m$ is the set of all polynomials of degree at most m on any arbitrary triangle. We multiply Equation (1) by a test function $v_h(\mathbf{x}) \in V_h^m$ and integrate the equation over the element T to obtain the following equation:

$$\frac{d}{dt} \int_T \mathbf{u}(\mathbf{x}, t) v_h(\mathbf{x}) d\Omega - \int_T \mathbf{F}(\mathbf{u}) \cdot \nabla v_h(\mathbf{x}) d\Omega + \sum_{e \in \partial T} \int_e v_h(\mathbf{x}) \mathbf{F}(\mathbf{u}) \cdot \hat{\boldsymbol{n}}_e d\Gamma = 0, \tag{4}$$

where ∂T denotes the boundary of the triangular element T and $\hat{\boldsymbol{n}}_e$ is the outward unit normal to the edge specified by the symbol e. Equation (4) is the exact representation of the weak form of Equation (1) since no approximation is introduced until this point. Next, we replace $\mathbf{u}$ with the approximate solution $\mathbf{u}_h$ to derive the semidiscrete form of Equation (4). The problem essentially reduces to the following: Find $\mathbf{u}_h \in (V_h^m)^4\ \forall\ v_h \in V_h^m$ on every $T \in \tau_h$ such that

$$\frac{d}{dt} \int_T \mathbf{u}_h(\mathbf{x}, t) v_h(\mathbf{x}) d\Omega - \int_T \mathbf{F}(\mathbf{u}_h) \cdot \nabla v_h(\mathbf{x}) d\Omega + \sum_{e \in \partial T} \int_e v_h(\mathbf{x}) \mathbf{F}(\mathbf{u}_h) \cdot \hat{\boldsymbol{n}}_e d\Gamma = 0. \tag{5}$$

Next, approximations of different terms of Equation (5) will be outlined. The volume integration $\int_T \mathbf{F}(\mathbf{u}_h) \cdot \nabla v_h(\mathbf{x}) d\Omega$ on the element T can be approximated with a numerical quadrature of sufficiently high order. The approximation can be given by

$$\int_T \mathbf{F}(\mathbf{u}_h) \cdot \nabla v_h(\mathbf{x}) d\Omega \approx 2|T| \sum_{i=1}^{N_q(m)} w_i^q \mathbf{F}\left(\mathbf{u}_i^q\right) \cdot \nabla v_h\left(\mathbf{x}_i^q\right), \tag{6}$$

where $N_q(m)$ is the number of quadrature points to integrate the function on any triangular element when $\mathbf{u}$ is approximated in $(\mathbb{P}^m)^4$, $\mathbf{x}_i^q$ are the $N_q(m)$ number of quadrature points on any triangular element, $\mathbf{u}_i^q$ are the conservative variables evaluated at $\mathbf{x}_i^q$, w_i^q are the weights associated to this integration rule, and $|T|$ denotes the area of the triangular element. In this paper, a cardinal function algorithm of Taylor et al[44] is adopted to calculate the quadrature points and weights to numerically integrate a function on a triangular element (see Appendix B for more details). The line integration as featuring in Equation (4) can be calculated by a sufficiently high-order Gauss quadrature given by the following form:

$$\sum_{e \in \partial T} \int_e v_h(\mathbf{x}) \mathbf{F}(\mathbf{u}_h) \cdot \hat{\boldsymbol{n}}_e d\Gamma \approx \sum_{e \in \partial T} \frac{|e|}{2} \sum_{i=1}^{N_q^G(m)} w_i^G v_h\left(\mathbf{x}_i^G\right) \mathbf{F}\left(\mathbf{u}_i^G\right) \cdot \hat{\boldsymbol{n}}_e, \tag{7}$$

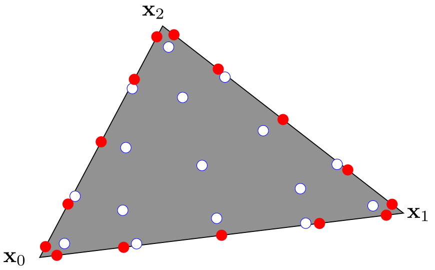


FIGURE 1 The quadrature points on a typical triangular element for a fifth-order discontinuous Galerkin method. ● points show the edge quadrature points for approximating Equation (7). ○ points show the quadrature points for approximating Equation (6) [Colour figure can be viewed at wileyonlinelibrary.com]

where $N_q^G(m)$ is the number of Gauss-Legendre quadrature points to integrate the function on any edge of a triangular element when $\mathbf{u}_h$ is in $(\mathbb{P}^m)^4$, $\mathbf{x}_i^G$ are the $N_q^G(m)$ number of Gauss-Legendre quadrature points on any edge, and $\mathbf{u}_i^G$ denotes the state variable evaluated at $\mathbf{x}_i^G$. It was proven by Cockburn et al[5] that for an $(m+1)$th-order RKDG, the quadrature rules over each of the faces of T have to be exact for polynomials of degree $(2m+1)$, and the quadrature rules over T have to be exact for polynomials of degree $2m$. The number of quadrature points $N_q(m)$ and $N_q^G(m)$ are chosen accordingly. Figure 1 shows the quadrature points on a typical triangular element for evaluating integrals (6) and (7). Note that $\mathbf{F}(\mathbf{u}_i^G)\cdot\hat{\boldsymbol{n}}_e$ needs to be approximated through any standard approximate Riemann solver. In our present work, a simple local Lax-Friedrichs (LLF) solver is employed. The LLF solver, being a function of the left and right states at a point on an edge and the outward unit normal vector to that edge, can be expressed by the following form:

$$\mathbf{F}\left(\mathbf{u}_i^G\right)\cdot\hat{\boldsymbol{n}}_e \approx \tilde{\mathbf{f}}\left(\mathbf{u}_i^G\right) = \frac{1}{2}\left[\mathbf{F}\left(\mathbf{u}_i^{G+}\right)+\mathbf{F}\left(\mathbf{u}_i^{G-}\right)\right]\cdot\hat{\boldsymbol{n}}_e - \frac{\alpha}{2}\left[\mathbf{u}_i^{G+}-\mathbf{u}_i^{G-}\right]. \tag{8}$$

In Equation (8), α is taken to be the maximum magnitude of eigenvalues of flux Jacobian matrix at $\mathbf{x}_i^G$ in the direction of $\hat{\boldsymbol{n}}_e$. $\mathbf{u}_i^{G+}$ and $\mathbf{u}_i^{G-}$ are the conserved variables at $\mathbf{x}_i^G$ are taken from the interior and exterior information on the triangle, respectively.

2.1 | Spatial discretization

This section describes the spatial discretization and approximations of different terms of Equation (5). Unlike the nodal DG,[45] the modal RKDG approach, which is pursued in this work, relies on the approximation of conserved variables on each triangular element T with a linear combination of $N(m)$ number of orthogonal functions: $\Psi_i^{(T)}(\mathbf{x}) \in \mathbb{P}^m(T)$:

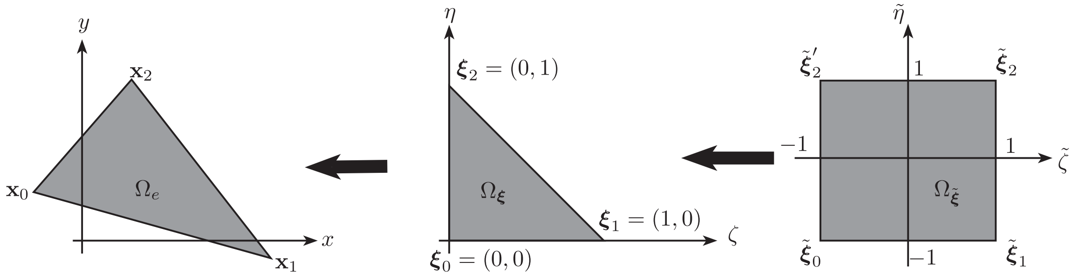
$$\mathbf{u}_h^{(T)}(\mathbf{x},t) = \sum_{i=1}^{N(m)} \hat{\mathbf{u}}_i^{(T)}(t)\Psi_i^{(T)}(\mathbf{x}). \tag{9}$$

The superscript (T) is used to differentiate the approximations on different triangular elements. Also, it is worth noting that the time dependence of the solution is encapsulated in the moments $\hat{\mathbf{u}}_i^{(T)}$ of the orthogonal functions in the above approximation and hence facilitates the subsequent implementation. However, normalized $\Psi_i^{(T)}(\mathbf{x})$ functions construct the basis for the finite element space V_h^m. Unlike the nodal DG approach, $\Psi_i^{(T)}(\mathbf{x})$ are chosen as test functions $v_h(\mathbf{x})$ while deriving the weak form of Equation (1). Consequently, the first term of Equation (5) simplifies further as

$$\frac{d}{dt}\int_T \mathbf{u}_h(\mathbf{x},t)\Psi_i^{(T)}(\mathbf{x})d\Omega = 2|T|\psi_i\frac{d\hat{\mathbf{u}}_i^{(T)}(t)}{dt}. \tag{10}$$

In Equation (10), the orthogonality relation for the orthogonal functions $\Psi_i^{(T)}(\mathbf{x})$ on the triangular element T is used: $\int_T \Psi_i^{(T)}(\mathbf{x})\Psi_j^{(T)}(\mathbf{x})d\Omega = 2|T|\psi_i$ if $i=j$ and 0 otherwise. ψ_i is a scalar constant that comes from the orthogonality relation for the chosen modal basis functions on the standard triangle, which is discussed later in this section and in Appendix A. Next, combining the approximations for all the terms of Equation (5) yields the semidiscrete form of equations governing the time evolution of the *ith* moment of the conserved variables on a triangular element T.

$$2|T|\psi_i\frac{d\hat{\mathbf{u}}_i(t)}{dt} = 2|T|\sum_{j=1}^{N_q(m)} w_j^q\mathbf{F}\left(\mathbf{u}_j^q\right)\cdot\nabla\Psi_i\left(\mathbf{x}_j^q\right) - \sum_{e\in\partial T}\frac{|e|}{2}\sum_{j=1}^{N_q^G(m)} w_j^G\Psi_i\left(\mathbf{x}_j^G\right)\tilde{\mathbf{f}}\left(\mathbf{u}_j^G\right). \tag{11}$$



y
$\mathbf{x}_2$
$\mathbf{x}_0$
Ω_e
x
$\mathbf{x}_1$
η
$\boldsymbol{\xi}_2 = (0,1)$
$\Omega_{\boldsymbol{\xi}}$
$\boldsymbol{\xi}_1 = (1,0)$
ζ
$\boldsymbol{\xi}_0 = (0,0)$
$\tilde{\eta}$
$\tilde{\boldsymbol{\xi}}_2'$
1
$\tilde{\boldsymbol{\xi}}_2$
-1
1
$\tilde{\zeta}$
$\Omega_{\tilde{\boldsymbol{\xi}}}$
$\tilde{\boldsymbol{\xi}}_0$
-1
$\tilde{\boldsymbol{\xi}}_1$

FIGURE 2 The mapping of any triangle to the standard triangle given by Equations (13a) and (13b) and the mapping of the standard triangle to the standard square given by Equation (14)

In Equation (11), superscript (T) is removed from all the terms as this is not causing any ambiguity. The above equation can be solved for the moments $\hat{\mathbf{u}}_i(t)$ on all triangular elements with a given initial condition and consistent boundary conditions on the domain boundaries using a time integration scheme.

To facilitate the numerical implementation, the orthogonal functions are defined on a standard triangle having scaled coordinate $\boldsymbol{\xi} \equiv (\zeta, \eta)$. An isosceles right triangle in the $\zeta - \eta$ plane is taken to be the standard triangle such that the three vertices $\mathbf{x}_0$, $\mathbf{x}_1$, and $\mathbf{x}_2$ (specified in an anti-clockwise sense) of any triangle T are mapped to the vertices of the standard triangle $(0,0)$, $(1,0)$, and $(0,1)$, respectively, as shown in Figure 2. In this way, one-to-one mapping between a physical triangle and the standard triangle can be derived as follows:

$$\mathbf{x} = \mathbf{x}_0(1 - \zeta - \eta) + \mathbf{x}_1\zeta + \mathbf{x}_2\eta. \tag{12}$$

Similarly, an inverse mapping is obtained to find the coordinate of any point in any physical triangle from its vertices and the corresponding coordinate in the $\zeta - \eta$ plane as follows:

$$\zeta = \frac{(x - x_0)(y_2 - y_0) - (x_2 - x_0)(y - y_0)}{|\mathbf{J}|}, \tag{13a}$$

$$\eta = \frac{(x_1 - x_0)(y - y_0) - (x - x_0)(y_1 - y_0)}{|\mathbf{J}|}, \tag{13b}$$

where $\mathbf{x}_0 \equiv (x_0, y_0)$, $\mathbf{x}_1 \equiv (x_1, y_1)$, and $\mathbf{x}_2 \equiv (x_2, y_2)$ are the three vertices of T and $|\mathbf{J}|$ is the Jacobian associated with the mapping given by Equation (12). For any triangular element, $|\mathbf{J}|$ turns out to be a constant (see Appendix A). Both the forward mapping and its inverse mapping are frequently used during the present computation. On the standard triangle, a complete set of orthogonal functions can be defined in many ways to develop a generalized numerical methodology for an RKDG of arbitrary order. In this work, Proriol polynomials[46] are employed as the basis functions. Being dependent only on the Jacobi polynomials which are orthogonal functions of a single variable, the Proriol polynomials can be computed in a straightforward way by computing the Jacobi polynomials. The Proriol polynomials are defined in the standard square, ie, in the $\tilde{\zeta} - \tilde{\eta}$ plane. On this plane, a standard square is defined by $\{(\tilde{\zeta}, \tilde{\eta}) : |\tilde{\zeta}| \leqslant 1, |\tilde{\eta}| \leqslant 1\}$, as shown in Figure 2. The mapping from the standard isosceles triangle to the standard square is known as Duffy's transform and is given by

$$\tilde{\zeta} = \frac{2\zeta}{1 - \eta} - 1 \text{ and } \tilde{\eta} = 2\eta - 1. \tag{14}$$

One may notice that this transformation is not unique at $\boldsymbol{\xi} = (0, 1)$ and that is why extra care is needed while computing the Proriol polynomials accurately at that singular point. The Proriol functions are defined on the standard square by

$$\Phi_{kl}(\tilde{\zeta}, \tilde{\eta}) = \left(\frac{1 - \tilde{\eta}}{2}\right)^k P_k^{(0,0)}(\tilde{\zeta}) P_l^{(2k+1,0)}(\tilde{\eta}), \tag{15}$$

where $P_k^{(0,0)}(\tilde{\zeta})$ and $P_l^{(2k+1,0)}(\tilde{\eta})$ are the Jacobi polynomials of degree k and l, respectively. Explicit expressions of the first few Proriol polynomials are given in Appendix A. A function $\mathbf{u}_h^{(T)}(\mathbf{x}, t) \in (\mathbb{P}^m)^4$ on any triangular cell T can be approximated in terms of the Proriol polynomials as

$$\mathbf{u}_h^{(T)}(\mathbf{x},t) = \sum_{k=0}^{m}\sum_{l=0}^{m-k}\hat{\mathbf{u}}_{kl}^{(T)}(t)\Phi_{kl}(\tilde{\xi}). \tag{16}$$

Equivalence of the above approximation of $\mathbf{u}^{(T)}(\mathbf{x},t)$ and the approximation by Equation (9) determines the number of required polynomials to approximate a function in $(\mathbb{P}^m)^4$, which is $N(m) = (m+1)(m+2)/2$. It is worth noting that $\tilde{\xi}$ can be uniquely determined for a given coordinate $\mathbf{x}$ in T by two consecutive mappings, namely Equations (13a) and (13b) and (14). Also, the double-indexed moment $\hat{\mathbf{u}}_{kl}^{(T)}(t)$ and the single-indexed moment $\hat{\mathbf{u}}_i^{(T)}(t)$ are actually the same if the indices are related by the following relation:

$$i = \frac{k(2m-k+3)}{2} + l. \tag{17}$$

Furthermore, a scheme with spatial order $(m+1)$ is obtained if the numerical solution on each triangular element are approximated in $(\mathbb{P}^m)^4$.

2.1.1 | Initialization

To evolve the moments of the state variables according to Equation (11), we need to provide the initial condition that is consistent with Equation (9). For a given initial condition $\mathbf{u}_0(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t = 0)$, corresponding moments can be approximated by performing an L^2-projection, which leads to the computations of the following integrals to find the initial moments on any triangular element T:

$$\hat{\mathbf{u}}_i^{(T)}(t=0) = \frac{1}{|T|}\int_T \mathbf{u}_0(\mathbf{x})\Psi_i^{(T)}(\mathbf{x})d\Omega, \qquad \forall\, i \in \{1,2,\ \dots\ ,N(m)\}. \tag{18}$$

The integral of Equation (18) can be approximated by choosing a quadrature rule exact for polynomials of degree $2m$ as for an $(m+1)$th-order RKDG discretization, where $\Psi_i^{(T)}$ is in $\mathbb{P}^m$ and approximation of $\mathbf{u}_0(\mathbf{x})$ is in $(\mathbb{P}^m)^4$. After initializing the moments of the conservative vector of each triangular element, a suitable time integration scheme is employed to evolve those moments in time. This will be discussed in the following.

2.2 | The Runge-Kutta time integration and calculation of time step

The spatial discretization over the physical domain of interest reduces the Euler equations to a set of coupled ordinary differential equations given by Equation (11). A third-order explicit strong stability preserving Runge-Kutta (SSP-RK3) scheme[47] is applied for the time integration. On a triangular element, the vector of moments of any state variable $\hat{\mathbf{u}}_n^{(T)}$ is updated for the next time level by the following stages:

$$\hat{\mathbf{u}}_{(1)}^{(T)} = \hat{\mathbf{u}}_n^{(T)} + \delta t\mathcal{L}\left(\hat{\mathbf{u}}_n^{(T)}, t_n\right), \tag{19a}$$

$$\hat{\mathbf{u}}_{(2)}^{(T)} = \frac{3}{4}\hat{\mathbf{u}}_n^{(T)} + \frac{1}{4}\hat{\mathbf{u}}_{(1)}^{(T)} + \frac{1}{4}\delta t\mathcal{L}\left(\hat{\mathbf{u}}_{(1)}^{(T)}, t_n + \delta t\right), \tag{19b}$$

$$\hat{\mathbf{u}}_{n+1}^{(T)} = \frac{1}{3}\hat{\mathbf{u}}_n^{(T)} + \frac{2}{3}\hat{\mathbf{u}}_{(2)}^{(T)} + \frac{2}{3}\delta t\mathcal{L}\left(\hat{\mathbf{u}}_{(2)}^{(T)}, t_n + \frac{\delta t}{2}\right), \tag{19c}$$

where $\mathcal{L}$ denotes the spatial discretization.

To achieve numerical stability of the scheme, time step is chosen by the following condition:

$$\delta t < \min_{T\in\tau_h}\frac{\text{CFL}\ r^{(T)}}{\left(|v_x| + |v_y| + 2a\right)}, \tag{20}$$

where $a = \sqrt{\gamma p/\rho}$ is the local acoustic velocity, v_x and v_y are x and y components of velocity field, respectively, and $r^{(T)}$ is the radius of the circumscribed circle of a triangular element T. *CFL* is the specified Courant-Friedrichs-Lewy (CFL) number. $(|v_x| + |v_y| + 2c)_{\max}^{(T)}$ is calculated based on the subcell averaged values among all $4m^2$ number of subcells of a triangular cell. The condition (20) can be derived from a von Neumann stability analysis of the method for the linear advection equation to ensure the L^2-stability. This condition is equally valid for nonlinear equations to prevent the amplification of round-off errors. For the stability of the scheme, $CFL = 1/(2m+1)$ is chosen in practice for the RKDG method where spatial approximations are in $\mathbb{P}^m$, and $(m+1)$th-order $(m+1)$ stages SSP-RK time integration are employed. This can be proved trivially for $m = 0$ and the same for $m = 1$ was proved by Cockburn and Shu.[2] For $m \geqslant 2$, numerical experiments confirm even smaller magnitude of CFL number than its estimated value from the analysis (ie, $CFL = 1/(2m+1)$). For more details on this, readers may refer to table 2.2 of the review article of Cockburn and Shu.[48] In our work, we presented the simulation results for the DG discretization in $\mathbb{P}^3$ and $\mathbb{P}^4$ with the SSP-RK3 time integration scheme. From the work of Cockburn and Shu,[48] the estimated CFL number for the L^2-stability of our scheme is 0.089. Our numerical experiments for the solution of the compressible Euler equations in this work show that the scheme is stable for the CFL number between 0.1 and 0.05, which is very close to the estimated CFL number in the work of Cockburn and Shu[48] for the L^2-stability of the scheme.

3 | THE SUBCELL FINITE VOLUME LIMITER

This section describes the concept of the a posteriori subcell limiter, which was first proposed by Dumbser and Loubere.[23] This subcell limiter is adopted in our present work with required modifications for its implementation with AMR. A posteriori limiters are essentially different from the classical DG limiters such as TVB limiter and other DG limiters, which rely on the concepts of WENO or HWENO. In the high-order RKDG, the major role of a limiter is to assure spurious oscillation-free solution at any instant of time, while preserving the high-order of accuracy at the smooth regions. Hence, the essential feature of a limiter is to introduce the ability to differentiate the physical oscillations and the oscillations that appear due to some numerical artifact. The key step in achieving that goal for both the classical and a posteriori limiters is to identify the cells on which limiting might be needed, ie, troubled cells through an appropriate troubled cell indicator or shock detection technique. This step, of course, becomes an important ingredient of any limiting technique. The troubled cell indicator, which is used in our work, is discussed in the next section.

Before discussing on a posteriori subcell limiter, it is important to realize the necessity of this alternative limiting strategy even when classical limiters, especially those which rely on the concepts of the WENO and the HWENO, work reasonably well for DG with AMR. Classical limiters reconstruct the DG solution on any troubled cell by using the solution on itself and mass conservative extrapolated solutions on this cell from its neighbors. While combining all possible provisional solutions on the troubled cell, classical limiters adaptively assign a small weight to a bad candidate, assuring a spurious oscillation-free final solution on the troubled cell. The only assumption associated with the classical limiters is that the provisional solution on the troubled cell or the provisional solutions provided by extrapolated solutions from its neighbors may not be good solutions but should not contain any floating point error (NaN or Inf). Then, only it is possible to combine those solutions with properly chosen weights to get rid of any bad component of the solution. However, very high-order RKDG may produce a solution on a troubled cell, which is not even meaningful. Another motivation to choose an a posteriori subcell limiter is the compactness of the stencil used for the limiting. A compact limiter is desirable especially for the implementation of the RKDG with an AMR. In contrast to that, the robustness of the classical limiter relies on the wider span of the selected stencil. However, the WENO and the HWENO-based compact limiters work well for DG methods up to a certain order for the flows having strong shocks.

A general implementation strategy for a posteriori finite volume subcell limiter is discussed next. First, a provisional solution $\mathbf{u}_*^{(T)} \in (\mathbb{P}_m)^4$ on a cell T for the next time level $(t + \delta t)$ is obtained using the RKDG from $\mathbf{u}_n^{(T)}$ (the solution at time t) without applying any limiter. Next, $\mathbf{u}_*^{(T)}$ is checked against some numerical and physical admissibility criteria as discussed in the next section. If this provisional solution qualifies against those preliminary checks, then $\mathbf{u}_*^{(T)}$ is further checked against some shock detection criteria to determine whether this cell is a troubled cell or not. If $\mathbf{u}_*^{(T)}$ fails against those preliminary numerical and physical admissibility checks, then this cell is identified as a troubled cell without any further investigation. After identifying a cell T as a troubled cell, the provisional solution $\mathbf{u}_*^{(T)}$ is discarded. In place of that discarded solution, subcell averages at time t, ie, $\overline{\mathbf{u}}_n^{(T)}$ is evolved with a first-order finite volume method and with the same time integration scheme as the RKDG to obtain subcell averages $\overline{\mathbf{u}}_{n+1}^{(T)}$ at time $(t + \delta t)$. An appropriate choice of the CFL number assures reliable subcell averages at time $(t + \delta t)$. Subsequently, the DG solution $\mathbf{u}_{n+1}^{(T)}$ at time $(t + \delta t)$ is reconstructed from $\overline{\mathbf{u}}_{n+1}^{(T)}$ to resume the time integration by the RKDG. To perform an a posteriori subcell limiting, any triangular cell is divided into a fixed number of uniform subcells. In our present work, number of subcells $N_s(m) = 4m^2$ is considered for an RKDG of order $(m+1)$ as suggested by Dumbser and Loubere.[23] This choice of $N_s(m)$ for a triangular element assures exactly the same number of degrees of freedom for the RKDG solution and for the solution in terms of subcell averages. Figure 3 shows the division of a typical troubled cell into a specified $N_s(m)$ number of subcells and the

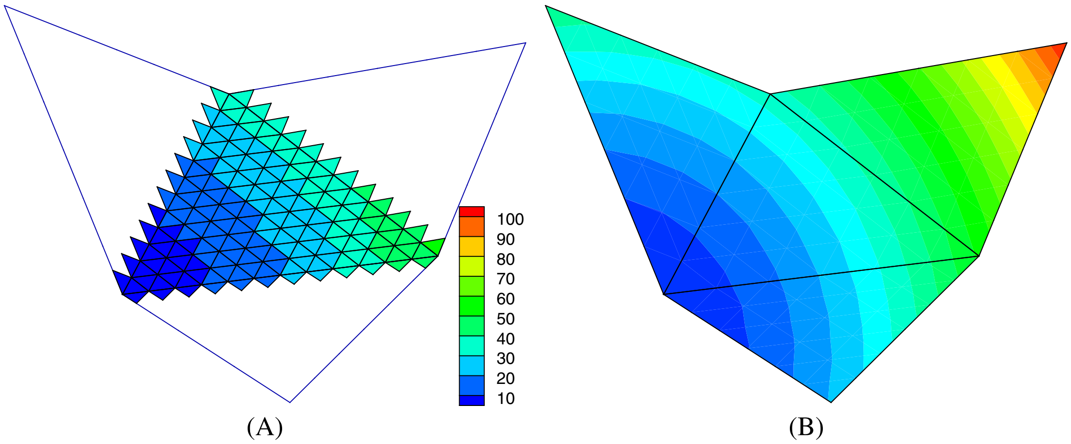


FIGURE 3 Subcells used for the a posteriori subcell finite volume limiter on a typical troubled cell for a seventh-order discontinuous Galerkin scheme. To update the subcell averages of the subcells of the target cell, boundary subcells are found from the neighboring cells of the target cell. The right-hand figure shows the troubled cell along with its immediate neighbors, which may need to be constructed for an application with an adaptive mesh refinement [Colour figure can be viewed at wileyonlinelibrary.com]

boundary subcells from the immediate neighboring cells for applying an a posteriori finite volume limiter. Algorithm 1 summarizes the present numerical strategy.

Algorithm 1 DG-SEM with an a posteriori finite volume sub-cell limiter

```
Input : initial state variable(s) on an element $T$ at time $t_n$, $\mathbf{u}_n^{(T)}$
Calculate sub-cell averages $\overline{\mathbf{u}}_n^{(T)}$ and store
Evolve $\mathbf{u}_n^{(T)} \rightarrow \mathbf{u}_*^{(T)}$ by the DG-SEM without any limiter
if $\mathbf{u}_*^{(T)}$ is an admissible solution then
    $\mathbf{u}_{n+1}^{(T)} = \mathbf{u}_*^{(T)}$
else
    $T$ is a troubled cell
    Evolve $\overline{\mathbf{u}}_n^{(T)} \rightarrow \overline{\mathbf{u}}_{n+1}^{(T)}$ by a first order finite volume method
    Reconstruct DG solution $\mathbf{u}_{n+1}^{(T)}$ from $\overline{\mathbf{u}}_{n+1}^{(T)}$
end if
```

The calculation of subcell averages from the moments on a triangular cell T and the calculation of moments with respect to the Proriol polynomials from the given set of subcell averages of a function are two major components for the implementation of an a posteriori finite volume subcell limiter. Those two steps are discussed in details next.

3.1 | Computation of subcell averages from the DG solution in a cell

In this section, a computation procedure of subcell averages from the $N(m)$ number of moments of any scalar-valued function $u_h^{(T)}$ on any triangular element T is discussed. Exactly same steps are followed for each component of the conservative vector. We denote the $N(m)$ number of moments of this function with respect to the Proriol basis function as $\hat{\mathbf{u}}^{(T)} \equiv \{\hat{u}_1^{(T)}, \hat{u}_2^{(T)} \ldots \hat{u}_{N(m)}^{(T)}\}$. Also, the $N_s(m)$ number of subcell averages is denoted by $\overline{\mathbf{u}}^{(T)} \equiv \{\overline{u}_1^{(T)}, \overline{u}_2^{(T)} \ldots \overline{u}_{N_s(m)}^{(T)}\}$. Here, the superscript T indicates the fact that this function is restricted to T. The subcell averages on $N_s(m)$ subcells can be calculated from the following equation:

$$\overline{\mathbf{u}}^{(T)} = \mathbf{P}_s \hat{\mathbf{u}}^{(T)}, \tag{21}$$

where $\mathbf{P}_s \in \mathbb{R}^{N_s(m) \times N(m)}$ is the projection matrix. The $(i,j)th$ entry of the projection matrix can be derived from the following equation:

$$p_{i,j} = \frac{1}{|T_{s_i}|} \int_{T_{s_i}} \Psi_j(\mathbf{x}) d\Omega, \tag{22}$$

which can be derived from Equation (9). T_{s_i}, which is featured in Equation (22), denotes the *ith* subcell of the triangular element T and clearly $T = \cup_{i=1}^{N_s(m)} T_{s_i}$. The computation of this projection matrix and its pseudoinverse turns out to be cumbersome if it is evaluated on the physical triangle. However, in the standard triangle, Equation (22) becomes

$$p_{i,j} = \frac{1}{|T_{s_i}|} \int_{\Omega_{\xi_{s_i}}} \Psi_j(\mathbf{x}) |\mathbf{J}| d\zeta d\eta, \tag{23}$$

where $\mathbf{J}$ is the Jacobian matrix of the transformation given by Equation (12) and $\Omega_{\xi_{s_i}}$ denotes the subcell in the standard triangle corresponding to T_{s_i}. It turns out that for a triangular element, $|\mathbf{J}| = 2|T_{s_i}|$ (see Appendix A for more details). Consequently, Equation (23) reduces to the following equation:

$$p_{i,j} = 2 \int_{\Omega_{\xi_{s_i}}} \Psi_j(\mathbf{x}) d\zeta d\eta. \tag{24}$$

Note that, unlike a quadrilateral element, the projection matrix $\mathbf{P}_s$ turns out to be independent of the coordinates of vertices of any triangular element. Hence, this projection matrix can be precomputed once in the preprocessing stage for its repeated usage on the fly.

3.2 | Reconstruction of the DG solution from the subcell averages

This problem is opposite to the previous one. We seek $N(m)$ number of moments of any scalar-valued function $u_h^{(T)}$ on a specific triangular cell T when $N_s(m)$ number of subcell averages $\overline{\mathbf{u}}_h^{(T)}$ are given on the subcells of T. Following Equation (16), $u_h^{(T)}$ can be expanded in terms of the Proriol polynomials as

$$u_h^{(T)} = \sum_{i=1}^{N(m)} \hat{\mathrm{u}}_i^{(T)} \Psi_i^{(T)}(\mathbf{x}). \tag{25}$$

The first set of equations is obtained by the fact that $u_h^{(T)}$ should recover the subcell averages exactly on the $N_s(m)$ subcells of T, ie, on the *jth* subcell T_{s_j} $\forall\ 1 \leqslant j \leqslant N_s(m)$

$$\begin{aligned} \frac{1}{|T_{s_j}|} \int_{T_{s_j}} u_h^{(T)} d\Omega &= \overline{u}_j^{(T)} \\ \Rightarrow \sum_{i=1}^{N(m)} \hat{\mathrm{u}}_i^{(T)} \frac{1}{|T_{s_j}|} \int_{T_{s_j}} \Psi_i^{(T)}(\mathbf{x}) d\Omega &= \overline{u}_j^{(T)}. \end{aligned} \tag{26}$$

Equations (26) gives rise to a set of $N_s(m)$ number of equations for $N(m)$ number of unknown moments. This overdetermined system can be solved in a least squares sense while retaining the total mass fixed. So, the equality constraint can be represented with the following equations:

$$\begin{aligned} \int_T u_h^{(T)} d\Omega &= \sum_{l=1}^{N_s(m)} \overline{u}_l^{(T)} |T_{s_l}| \\ \Rightarrow N_s(m) \hat{\mathrm{u}}_1^{(T)} &= \sum_{l=1}^{N_s(m)} \overline{u}_l^{(T)}. \end{aligned} \tag{27}$$

A general procedure to efficiently solve this constrained optimization problem is outlined in Appendix B.

3.3 | Detection of the troubled cells

For the Euler equations, state variables at time t_n, denoted by $\mathbf{u}_n^{(T)} \equiv \{\rho,\ \rho v_x,\ \rho v_y,\ \rho E\}_n^{(T)}$ on any cell T, is evolved with the high-order RKDG first to obtain provisional state variables $\mathbf{u}_*^{(T)}$ at time $(t_n + \delta t)$. These provisional state variables are tested against a few admissibility conditions, as proposed by Dumbser and Loubere,[23] before accepting this as a solution at $(t_n + \delta t)$. The first condition that needs to be checked is the numerical admissibility of all the components of $\mathbf{u}_*^{(T)}$. If any of the components of $\mathbf{u}_*^{(T)}$ produces a floating point error (NaN or Inf), this cell is marked as a troubled cell without any further test. If $\mathbf{u}_*^{(T)}$ satisfies numerical admissibility criteria, then the provisional state variables are checked further against a few physical admissibility criteria. In the case of the Euler equations, density ρ and pressure p are two variables which cannot be negative. So, the positivity of provisional density and pressure are checked. If $\mathbf{u}_*^{(T)}$ does not satisfy the physical admissibility criteria, then this cell is marked as a troubled cell. If any exception is not found against the physical admissibility criteria, then this provisional solution is finally checked for the presence of any discontinuity in the density and entropy $s = (p/\rho^\gamma)$ variable on that cell T.

To check for the presence of any discontinuity on a particular cell T, provisional density $(\rho_*^{(T)})$ and provisional entropy variable $(s_*^{(T)})$ are checked against the so-called discrete maximum principle, which is given by the following conditions on that cell:

$$\max\left(\overline{\rho}_*^{(T)}\right) < \max\left(\overline{\rho}_n^{(T)}\right) + \epsilon, \qquad \max\left(\overline{s}_*^{(T)}\right) < \max\left(\overline{s}_n^{(T)}\right) + \epsilon, \tag{28a}$$

$$\min\left(\overline{\rho}_*^{(T)}\right) > \min\left(\overline{\rho}_n^{(T)}\right) - \epsilon, \qquad \min\left(\overline{s}_*^{(T)}\right) > \min\left(\overline{s}_n^{(T)}\right) - \epsilon, \tag{28b}$$

where $\max(\overline{\rho}_*^{(T)})$ and $\min(\overline{\rho}_*^{(T)})$ denote the maximum and minimum subcell averages of $\rho_*^{(T)}$, respectively. Similarly, $\max(\overline{\rho}_n^{(T)})$ and $\min(\overline{\rho}_n^{(T)})$ denote the maximum and minimum subcell averages of $\rho_n^{(T)}$, respectively. If all of the conditions, given by Equations (28a) and (28b), are satisfied on a cell, this cell is ensured to be a problem-free cell, and the provisional solution provided by the high-order RKDG is considered to be the solution at the next time level. Note that if the solution should remain constant in time over any cell, a small positive value of ϵ ensures that this not to be detected as a troubled cell. Furthermore, a strict verification of the discrete maximum principle requires an evaluation of the optimal values of a function in any triangular cell. This is not at all a feasible option for an arbitrarily high-order method. We perform those checks by using the subcell averages of the density and entropy variables. That is why, it is necessary to introduce a small positive number ϵ in the conditions (28a) and (28b) to avoid the detection of a problem-free cell as a troubled cell. In our experience, $\epsilon = 10^{-10}$ works well for all the simulations performed in our present work. If a provisional solution does not satisfy conditions (28a) and (28b), then this solution is checked further before declaring this cell as a troubled cell. These final conditions check for any drastic change in the second derivatives of the provisional density or entropy function in comparison to the solution at the previous time step. Those extra conditions are motivated by the so-called 'MOOD loop', first proposed by Clain et al[29] in the context of a high-order multidimensional finite volume method. The provisional solution on a problem-free cell should additionally satisfy the following conditions:

$$\max\left(\overline{\rho}_{*_{xx}}^{(T)}\right) < \max_{\forall\tilde{T}}\left(\overline{\rho}_{n_{xx}}^{(\tilde{T})}\right) + \epsilon, \qquad \max\left(\overline{s}_{*_{xx}}^{(T)}\right) < \max_{\forall\tilde{T}}\left(\overline{s}_{n_{xx}}^{(\tilde{T})}\right) + \epsilon, \tag{29a}$$

$$\min\left(\overline{\rho}_{*_{xx}}^{(T)}\right) > \min_{\forall\tilde{T}}\left(\overline{\rho}_{n_{xx}}^{(\tilde{T})}\right) - \epsilon, \qquad \min\left(\overline{s}_{*_{xx}}^{(T)}\right) > \min_{\forall\tilde{T}}\left(\overline{s}_{n_{xx}}^{(\tilde{T})}\right) - \epsilon, \tag{29b}$$

$$\max\left(\overline{\rho}_{*_{yy}}^{(T)}\right) < \max_{\forall\tilde{T}}\left(\overline{\rho}_{n_{yy}}^{(\tilde{T})}\right) + \epsilon, \qquad \max\left(\overline{s}_{*_{yy}}^{(T)}\right) < \max_{\forall\tilde{T}}\left(\overline{s}_{n_{yy}}^{(\tilde{T})}\right) + \epsilon, \tag{29c}$$

$$\min\left(\overline{\rho}_{*_{yy}}^{(T)}\right) > \min_{\forall\tilde{T}}\left(\overline{\rho}_{n_{yy}}^{(\tilde{T})}\right) - \epsilon, \qquad \min\left(\overline{s}_{*_{yy}}^{(T)}\right) > \min_{\forall\tilde{T}}\left(\overline{s}_{n_{yy}}^{(\tilde{T})}\right) - \epsilon, \tag{29d}$$

where $\tilde{T}$ denotes the set of all immediate neighboring cells of T including T itself. $\max(\overline{\rho}_{*_{xx}}^{(T)})$ and $\min(\overline{\rho}_{*_{xx}}^{(T)})$ denote maximum and minimum values of subcell averages of the second derivative of $\rho_*^{(T)}$ with respect to x. Similarly, $\max(\overline{\rho}_{n_{xx}}^{(\tilde{T})})$ and $\min(\overline{\rho}_{n_{xx}}^{(\tilde{T})})$ denote the maximum and minimum values of subcell averages of the second derivative of $\rho_*^{(\tilde{T})}$ with respect to x. One may note that extra set of conditions (29a) to (29d) relax the strict maximum principle (conditions (28a) and (28b)) and allow any oscillatory physical solution to evolve by the unlimited RKDG. It is worth noting that there is a major difference in the implementation of the conditions (29a) to (29d) in our work from that of Clain et al.[29] Clain et al derived those conditions from the subcell averages of the state variables. In this present work, moments of the second derivatives are first calculated from the moments of the state variables on any triangular cell through the computation of the derivatives of the Proriol polynomials on a triangular element (see Equation A7 of Appendix A) and two consecutive L^2-projections. Next, the subcell averages of the second derivatives (ie, $\rho_{*_{xx}}^{(T)}$, $\rho_{*_{yy}}^{(T)}$, $s_{*_{xx}}^{(T)}$, $s_{*_{yy}}^{(T)}$, $\rho_{n_{xx}}^{(T)}$, $\rho_{n_{yy}}^{(T)}$, $s_{n_{xx}}^{(T)}$, and $s_{n_{yy}}^{(T)}$) are computed from their moments on the triangular element T by using Equation (21). This modification facilitates the implementation by maintaining the compactness of the DG-SEM. Detection of troubled cells is summarized in Algorithm 2.

Algorithm 2 Troubled cell detection

```
if any of the moments of $\mathbf{u}_*^{(T)}$ is NaN or Inf then
    $T$ is a troubled cell
else
    if any of the $\overline{\rho}_*^{(T)}$ and $\overline{p}_*^{(T)}$ becomes zero or negative then
        $T$ is a troubled cell
    else
        if $\mathbf{u}_*^{(T)}$ does not satisfy any of the conditions (28) then
            if $\mathbf{u}_*^{(T)}$ does not satisfy any of the conditions (29) then
                $T$ is a troubled cell
            else
                Limiting is not needed on $T$
            end if
        else
            Limiting is not needed on $T$
        end if
    end if
end if
```

4 | ADAPTIVE MESH REFINEMENT STRATEGY

The dynamic mesh adaption is an economic and efficient way to cope with a simulation of flows having a wide range of spatial scales. An AMR is advantageous in the RKDG in many ways in comparison to the other adaption strategies. First as there is no restriction on the conformity of the grid, a refinement strategy that even allows hanging nodes also works well in the DG framework. In addition to that, an AMR strategy is preferred when primary mesh quality needs to be maintained. In our present implementation, any triangular cell T is split into four triangular cells by introducing three new vertices which are the midpoints of three edges. Hence, the quality of newly generated child elements does not deteriorate in comparison to the same of the parent element. A reverse methodology is followed if the four adjacent child cells of the same parent node need to be merged. The refinement and derefinement of any triangular cell, as adopted in our present work, are shown in Figure 4. In our approach, a quadtree data structure is employed to serve this purpose. In the initial conforming primary mesh, data of each element is stored in a dynamically allocated structure, and all the cells are designated as level $= 0$. In any case, primary cells or root cells are not allowed to be deleted but only can be refined, thus assuring a volume-filling triangulation at any instant of time. A data structure, which is associated with every cell, consists of the following information about that cell.

- The link to its parent cell. This piece of information is NULL for the primary cells or root cells.
- The links to its four child cells. Those information are NULL if a cell is a leaf cell, ie, this cell does not have any child cell.
- All the leaf cells contain a DG solution in terms of the moments of the state variables.
- It is also necessary to store the subcell averages of the state variables. The subcell averages are useful to take corrective measures if the moments are found to be erroneous owing to the presence of flow discontinuities or flow features having very high gradients of state variables. After each time step, the subcell averages are updated from the DG solution if this particular cell is not detected as a troubled cell. For a troubled cell, the subcell averages are evolved from the subcell averages of the previous time step, and DG solutions are reconstructed from the subcell averages obtained from the subcell limiting.

Next, the refinement/derefinement criteria and the methodology for transferring data to the newly created leaf element(s) due to a refinement or derefinement will be addressed.

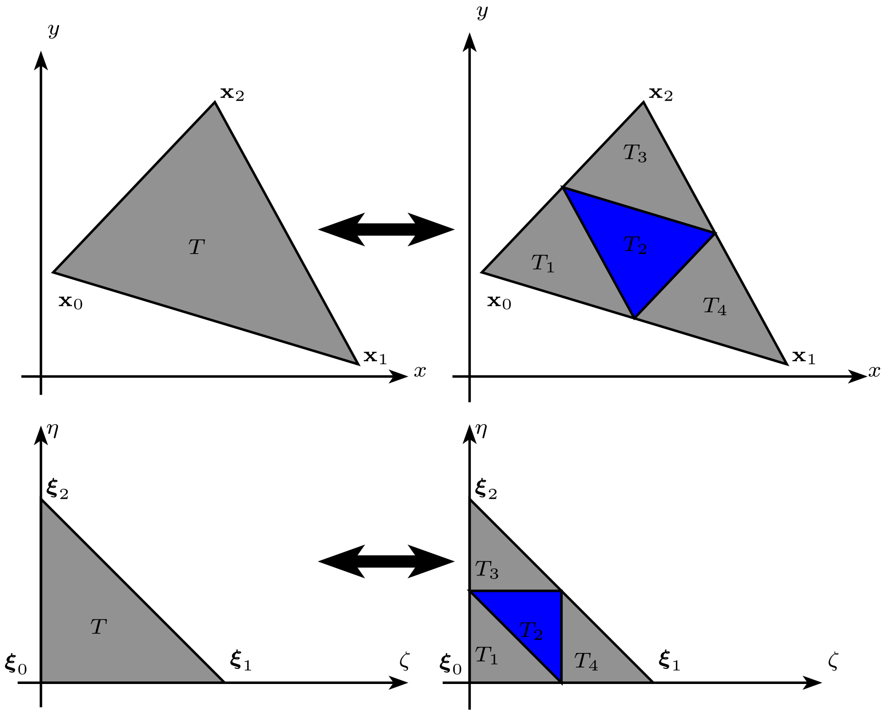


FIGURE 4 Refinement (arrow to the left) and derefinement (arrow to the right) of a triangular element. Above figures show the refinement and derefinement operations on a typical triangular element in the physical plane. Bottom figures show the refinement and derefinement operations for the standard triangle. Solution(s) on the newly created four child cells during refinement or newly created parent cell due to merging of four child cells can be computed efficiently by multiplying a precomputed matrix and the vector of moments of the solution(s) [Colour figure can be viewed at wileyonlinelibrary.com]

4.1 | The mesh refinement/derefinement criteria

The refinement/derefinement of cells, in our present work, ensures equidistribution of some quantitie(s) over each cell. For compressible flow simulations, the dilatation of velocity field ($\nabla \cdot \mathbf{v}$) is to a great extent a reliable indicator of the presence of strong shocks and, hence, adopted as a key determining quantity of refinement to assure cell clustering near to the shock locations. In addition to the velocity dilatation, the gradient of entropy variable ($\nabla p - a^2\nabla\rho$) also is taken as an additional parameter which senses contact discontinuities as well. To be precise, a cell is refined if the following condition is satisfied:

$$\alpha_d^{(T)} > \alpha_d^{\min} + \sigma_d^r \left(\alpha_d^{\max} - \alpha_d^{\min}\right), \tag{30a}$$

$$\text{or,}\ \alpha_s^{(T)} > \alpha_s^{\min} + \sigma_s^r \left(\alpha_s^{\max} - \alpha_s^{\min}\right), \tag{30b}$$

where σ_d^r and σ_s^r are two specified constants and the superscript r emphasizes that those parameters are associated with the refinement of cells. Both of those values are chosen between 0.6 and 0.8 for all the simulations in this paper. In Equations (30a) and (30b), other notations are defined as

$$\alpha_d^{(T)} = \frac{1}{|T|}\int_T |\nabla \cdot \mathbf{v}| d\Omega, \tag{31a}$$

$$\alpha_s^{(T)} = \frac{1}{|T|}\int_T \|\nabla p - a^2\nabla\rho\| d\Omega. \tag{31b}$$

The superscript (T) is used to emphasize that those refinement parameters are associated with a particular triangular element T. Two subscripts d and s stand for velocity dilatation and entropy, respectively. The symbol a, as featured in the last equation, denotes the local acoustic velocity in the fluid medium and computed by $\sqrt{\gamma p/\rho}$. Here, $\alpha_d^{\max}$, $\alpha_d^{\min}$, $\alpha_s^{\max}$, and $\alpha_s^{\min}$ are four global quantities on a specific triangulation τ_h^n and are defined by

$$\alpha_d^{\max} = \max_{\forall\, T\in\tau_h^n} \alpha_d^{(T)}, \qquad \alpha_d^{\min} = \min_{\forall\, T\in\tau_h^n} \alpha_d^{(T)}, \tag{32a}$$

$$\alpha_s^{\max} = \max_{\forall\, T\in\tau_h^n} \alpha_s^{(T)}, \qquad \alpha_s^{\min} = \min_{\forall\, T\in\tau_h^n} \alpha_s^{(T)}. \tag{32b}$$

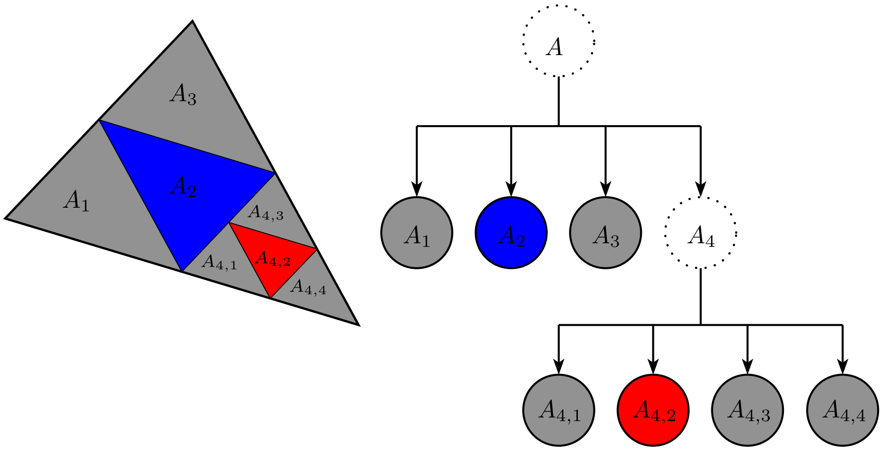


FIGURE 5 Left-hand figure shows a volume-filling splitting of a typical triangular cell of level = 0 and right-hand figure demonstrates the representation of the data structure for this particular refinement of a root cell or a cell of level = 0. Leaf nodes are represented by solid circles. Any node may have either four nodes or maybe a leaf node itself [Colour figure can be viewed at wileyonlinelibrary.com]

In a similar way, four child leaf cells of the same parent cell need to be merged if the following conditions are met:

$$\max\left[\alpha_d^{(T_1)}, \alpha_d^{(T_2)}, \alpha_d^{(T_3)}, \alpha_d^{(T_4)},\right] < \alpha_d^{\min} + \sigma_d^m\left(\alpha_d^{\max} - \alpha_d^{\min}\right), \tag{33a}$$

$$\max\left[\alpha_s^{(T_1)}, \alpha_s^{(T_2)}, \alpha_s^{(T_3)}, \alpha_s^{(T_4)}\right] < \alpha_s^{\min} + \sigma_s^m\left(\alpha_s^{\max} - \alpha_s^{\min}\right), \tag{33b}$$

where σ_d^m and σ_s^m are two specified constants and the superscript m emphasizes that those parameters are associated with the merging of child cells of the same parent cell.

4.2 | The mesh refinement/derefinement and associated data structures

The dynamic mesh refinement/derefinement needs a dynamic data structure for the efficient implementation of AMR algorithm. When child nodes of the same parent need to be merged, a dynamic memory allocation is needed for the nonleaf parent node to accommodate data from its child leaf nodes. At the same time, the data in those child leaf nodes also need to be deleted as soon as all the necessary data are transferred to their parent node. Exactly reverse actions are performed while refining a cell. A quadtree data structure is employed to perform the dynamic cell splitting and merging operations. The employed data structure is illustrated in Figure 5. A cell of level = 0 (ie, root cell) has the direct access to its direct child cells. So, any leaf cell, which is the direct or indirect child of the root cell, can be accessed from the address of the root cell itself. So, all the root nodes are traversed to perform refinement or derefinement in a similar manner. Next, data transfer to newly created leaf cells as a result of a refinement or derefinement is discussed.

4.2.1 | Data transfer during refinement

Four new leaf cells, resulting from a refinement of a parent leaf cell, get their own DG solution from the DG solution of the corresponding parent cell. Any component of solution on a parent leaf cell T, which needs to be refined, is given by $u_h^{(T)} = \sum_{i=1}^{N(m)} \hat{\mathrm{u}}_i^{(T)}\Psi_i^{(T)}$. This representation of the solution is exactly the same as Equation (9), with $\Psi_i^{(T)}$ and $\hat{\mathrm{u}}_i$ being the Proriol polynomials defined on T and moments with respect to the chosen basis, respectively. We seek similar solution representations on the newly created child cells as given by the following equation:

$$u_h^{(T_k)} = \sum_{i=1}^{N(m)} \hat{\mathrm{u}}_i^{(T_k)}\Psi_i^{(T_k)}, \tag{34}$$

where $k \in \{1, 2, 3, 4\}$. This problem can be solved without losing any information as provided by the parent cell. We outline the projection methodology for the first child node T_0. The similar procedure follows for other child nodes. On the standard triangle, the coordinates of three vertices of the first child cell are $(0, 0)$, $(1/2, 0)$, and $(0, 1/2)$. For the sake of an efficient computation, $N_q(m)$ number of quadrature points on the $\zeta - \eta$ plane inside the first child T_0 to compute the integral of any function of degree $2m$ are precomputed at the preprocessing stage. Those $N_q(m)$ number of quadrature points are denoted by $\{\boldsymbol{\xi}_1^{(T_0)}, \boldsymbol{\xi}_2^{(T_0)}, \ldots, \boldsymbol{\xi}_{N_q(m)}^{(T_0)}\}$. In the physical plane, those coordinates are $\{\mathbf{x}_1^{(T_0)}, \mathbf{x}_2^{(T_0)}, \ldots, \mathbf{x}_{N_q(m)}^{(T_0)}\}$. It turns out that $N(m)$ number of basis functions $\Psi_i^{(T)}$ on any of those quadrature points are independent of the vertices of T. This motivates one to precompute all possible basis functions on the quadrature points in the first child.

$$\mathbf{P}^{(T_1)} = \begin{bmatrix} \Psi_1^T\left(\mathbf{x}_1^{(T_1)}\right) & \Psi_2^T\left(\mathbf{x}_1^{(T_1)}\right) & \dots & \Psi_{N(m)}^T\left(\mathbf{x}_1^{(T_1)}\right) \\ \Psi_1^T\left(\mathbf{x}_2^{(T_1)}\right) & \Psi_2^T\left(\mathbf{x}_2^{(T_1)}\right) & \dots & \Psi_{N(m)}^T\left(\mathbf{x}_2^{(T_1)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_1^T\left(\mathbf{x}_{N_q(m)}^{(T_1)}\right) & \Psi_2^T\left(\mathbf{x}_{N_q(m)}^{(T_1)}\right) & \dots & \Psi_{N(m)}^T\left(\mathbf{x}_{N_q(m)}^{(T_1)}\right) \end{bmatrix}, \tag{35}$$

where $\mathbf{P}^{(T_1)} \in \mathbb{R}^{N_q(m)\times N(m)}$. Function values on the quadrature points of the first child cell can be obtained by multiplying the vector of moments of the solution $\{\hat{\mathrm{u}}_1^{(T)}, \hat{\mathrm{u}}_2^{(T)}, \dots, \hat{\mathrm{u}}_{N(m)}^{(T)}\}$ to the precomputed matrix $\mathbf{P}^{(T_1)}$. Thus, the moments of the local representation of the solution for the first child are obtained by an L_2 projection from the function values on the quadrature points. Similar projection matrices $\mathbf{P}^{(T_2)}$, $\mathbf{P}^{(T_3)}$, and $\mathbf{P}^{(T_4)}$ are precomputed for finding the solution on the second, third, and fourth child cells, respectively, from their parent cell.

4.2.2 | Data transfer during derefinement

The problem statement for data transfer to a parent cell from its child cells is exactly the opposite to the cell refinement. More precisely, this problem can be described as follows.

The DG solutions are given on the child cells in terms of the moments $u_h^{(T_k)} = \sum_{i=1}^{N(m)} \hat{\mathrm{u}}_i^{(T_k)}\Psi_i^{(T_k)}$, where $k \in \{1, 2, 3, 4\}$. We need to find out the moments of the solution on the parent cell before deleting its child cells. This process is associated with an inevitable loss of information. The first step for transferring data to the parent node T from its child cells, namely T_1, T_2, T_3, and T_4, is to compute the function values on the quadrature points of T. In a matrix form, function values on quadrature points of T can be calculated from the following:

$$\mathbf{u}_h^{(T)} = \tilde{\mathbf{P}}^{(T_1)}\hat{\mathbf{u}}^{(T_1)} + \tilde{\mathbf{P}}^{(T_2)}\hat{\mathbf{u}}^{(T_2)} + \tilde{\mathbf{P}}^{(T_3)}\hat{\mathbf{u}}^{(T_3)} + \tilde{\mathbf{P}}^{(T_4)}\hat{\mathbf{u}}^{(T_4)}, \tag{36}$$

where $\mathbf{u}_h^{(T)} \in \mathbb{R}^{N_q(m)}$ is the vector of function values at the quadrature points of T when $u_h \in \mathbb{P}^m$, $\hat{\mathbf{u}}^{(T_1)}$, $\hat{\mathbf{u}}^{(T_2)}$, $\hat{\mathbf{u}}^{(T_3)}$, and $\hat{\mathbf{u}}^{(T_4)}$ are the vectors of moments of the solution of corresponding child cells. The projection matrices $\tilde{\mathbf{P}}^{(T_1)}$, $\tilde{\mathbf{P}}^{(T_2)}$, $\tilde{\mathbf{P}}^{(T_3)}$, and $\tilde{\mathbf{P}}^{(T_4)}$ are precomputed and stored as those do not change when child tags are fixed on the standard triangle. Figure 4 shows the child numbering convention on the standard triangle, which is adopted in the present work. However, after gathering function values on the quadrature points of T, moments of the DG solution can be obtained through an L^2-projection.

5 | NUMERICAL RESULTS AND DISCUSSIONS

In this section, we present 2-D test cases to assess our numerical scheme. More specifically, we would like to show the effectiveness of the modified subcell limiter in eliminating spurious oscillations in the solution. The onset of spurious oscillations at any instant of time may be proved to be detrimental in the later stage of a simulation. Such spurious oscillations always contaminate the smooth and fine solution features and, in its worst consequence, result in an abrupt termination of the simulation due to floating point errors. For a higher-order RKDG schemes, the solution is more prone to produce spurious oscillations especially when the solution contains intricate flow features or flow discontinuities like shocks. That is why for a high-order RKDG method, the limiter has a more important role for a successful simulation. In the later sections, we present the results for a fifth-order RKDG method. This RKDG scheme with a subcell limiter can be extended to even higher-order methods without having any extra difficulties from the implementation point of view. Also, note that the value of the ratio of specific heats, $\gamma = 1.4$ is chosen in all our simulations unless it is specified otherwise.

5.1 | Two-dimensional cases

5.1.1 | Isentropic vortex test case

In our first 2-D test problem, the correctness in terms of convergence rate towards the exact or reference solution of our RKDG implementation on the unstructured triangular mesh is tested. To test that, we consider the isentropic vortex test

TABLE 1 The rate of convergence for fourth-order and fifth-order discontinuous Galerkin (DG) schemes without any mesh refinement for two-dimensional problems

	Fourth-order DG				**Fifth-order DG**			
N	$\|\|e^h\|\|_{L^2}$	O_{L^2}	$\|\|e^h\|\|_{L^\infty}$	O_{L^∞}	$\|\|e^h\|\|_{L^2}$	O_{L^2}	$\|\|e^h\|\|_{L^\infty}$	O_{L^∞}
124	5.998E-02	-	1.243E-01	-	1.383E-02	-	1.343E-02	-
518	2.648E-03	4.50	3.052E-03	5.35	3.230E-04	5.42	2.914E-04	5.53
1926	1.272E-04	4.38	1.865E-04	4.03	1.334E-05	4.60	2.464E-05	3.56
7668	5.079E-06	4.65	1.274E-05	3.87	2.937E-07	5.51	8.269E-07	4.90

problem which first appeared in the work of Yee et al.[49] The primitive variables are initialized with the following functions:

$$\rho_0 = \left[1 - \frac{(\gamma-1)\beta^2}{8\pi^2\gamma} e^{(1-x^2-y^2)}\right]^{\frac{1}{\gamma-1}}, \tag{37a}$$

$$u_0 = M\cos\alpha - \frac{\beta(y-y_0)}{2\pi} e^{(1-x^2-y^2)/2}, \tag{37b}$$

$$v_0 = M\sin\alpha + \frac{\beta(x-x_0)}{2\pi} e^{(1-x^2-y^2)/2}, \tag{37c}$$

$$p_0 = \rho_0^\gamma. \tag{37d}$$

In Equations (37a) to (37d), $\beta = 5$, $x_0 = y_0 = 0$, $M = 0.5$, and $\alpha = \pi/4$ are chosen. The Euler equations are solved with this initial condition on the domain $[-5, 5]^2$. The periodic boundary condition is applied at all the boundaries of the domain. The CFL number = 0.1 is chosen for all the runs of this particular test problem. This problem is run up to the final time, $T_f = 20\sqrt{2}$ units, yielding the identical exact solution at the final time and the initial condition. This happens due to that particular choice of initial flow field, which assures an exact balance between the centrifugal force and the pressure gradient at any point in the domain. However, in this particular case, the initial flow field is divergence-free and the entropy variable vanishes everywhere in the domain. That is why refinement based on equidistribution of the entropy and the divergence of the velocity field do not have any influence on the primary grid refinement.

The convergence rates towards the exact solution of fourth-order and fifth-order RKDG schemes are tabulated in Table 1. L^2 and L^∞ errors are inferred by calculating the following quantities:

$$L^2 \text{ error, } \|e^h\|_{L^2} = \sum_{i=0}^{N-1} \int_{\Omega_i} \left[\rho_0(x,y) - \rho(x,y,T_f)\right]^2 d\Omega, \tag{38a}$$

$$L^\infty \text{ error, } \|e^h\|_{L^\infty} = \max_{\forall\ (x,y)} \left|\rho_0(x,y) - \rho(x,y,T_f)\right|. \tag{38b}$$

In Equation (38a), Ω_i is the *ith* element, and N denotes the total number of leaf elements. The numerical order of accuracy is defined by

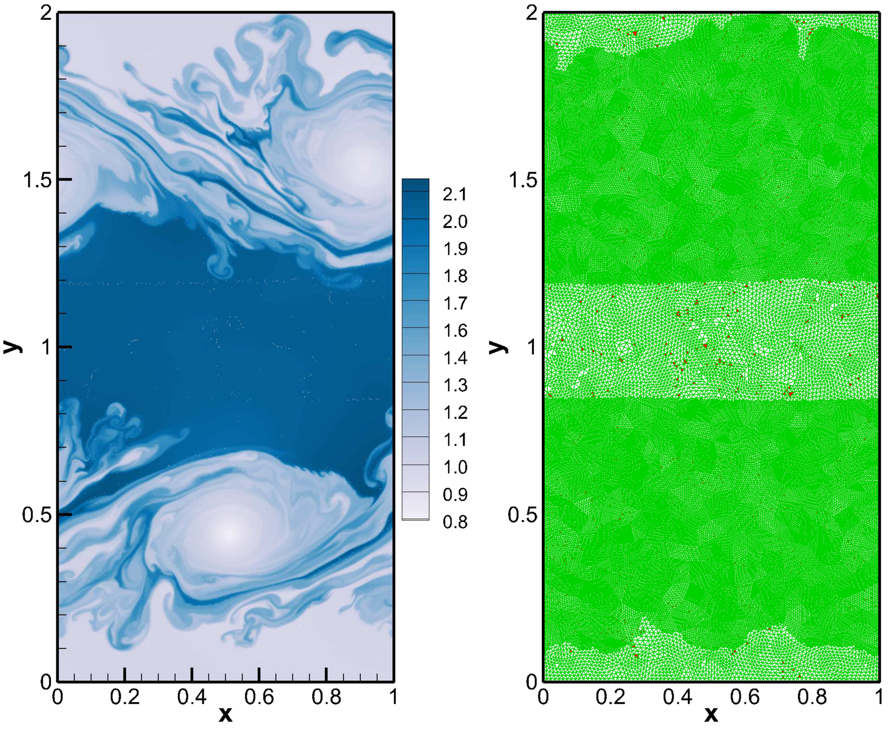
$$O_{L^2} = \log_2\left(\frac{\|e^h\|_{L^2}}{\|e^{h/2}\|_{L^2}}\right), \tag{39a}$$

$$O_{L^\infty} = \log_2\left(\frac{\|e^h\|_{L^\infty}}{\|e^{h/2}\|_{L^\infty}}\right). \tag{39b}$$

We see that the L^2 and L^∞ errors decrease with the increased number of primary elements yielding the expected convergence rate.

5.1.2 | Kelvin-Helmholtz instability

The Kelvin-Helmholtz instability is one of the most ubiquitous shear-driven hydrodynamic instabilities occurs in nature. However, such kind of instability may also occur between two adjacent flow streams having different fluid densities and



(A) (B)

FIGURE 6 A, Density contour for Kelvin-Helmholtz problem at the final time, *t*=5 by a fifth-order discrete Galerkin method on 2130 primary triangular elements and maximum three levels of adaptive mesh refinement;
B, Corresponding adaptively refined grid layout; C, Magnified view of the panel A; D, Corresponding magnified adaptively refined grid layout. Red patches indicate the troubled cells [Colour figure can be viewed at wileyonlinelibrary.com]

streamwise velocities in presence of disturbances in the form of a cross-stream velocity. This is achieved by the following initialization for the primitive variables on the domain $[0,1] \times [0,2]$:

$$\rho_0 = 1 + \frac{\rho_\infty}{2}\left[\tanh\left(\frac{y-y_1}{\epsilon}\right) - \tanh\left(\frac{y-y_2}{\epsilon}\right)\right], \tag{40a}$$

$$u_0 = u_\infty\left[\tanh\left(\frac{y-y_1}{\epsilon}\right) - \tanh\left(\frac{y-y_2}{\epsilon}\right)\right], \tag{40b}$$

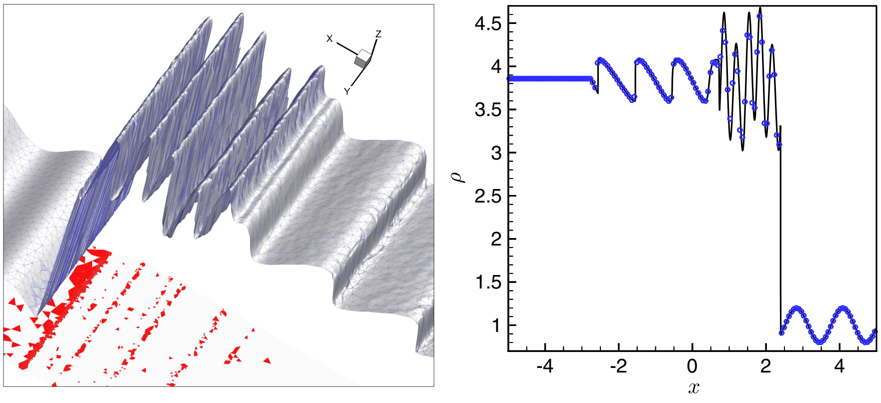
$$v_0 = A\sin(2\pi x)\left[e^{-(y-y_1)^2/\sigma^2} + e^{-(y-y_2)^2/\sigma^2}\right], \tag{40c}$$

$$p_0 = p_\infty, \tag{40d}$$

where $\rho_\infty = u_\infty = 1$, $p_\infty = 10$, $\epsilon = 0.05$, $\sigma = 0.2$, $A = 0.01$, $y_1 = 0.5$, and $y_2 = 1.5$. The chosen initial condition is similar to that in the work of Springel.[50] The periodic boundary condition is applied along both the directions. Here, the small amplitude disturbance acts as an initial stimulus for the generation of vortical structures at the interfaces between the two streams and subsequent turbulent mixing. Our demonstration computation is performed on only 2130 numbers of primary triangular elements in the physical domain and maximum three levels of refinements of any primary element corresponding to an effective resolution of 136 320 triangular cells. All associated grid refinement and derefinement parameters, namely, σ_d^r, σ_s^r, σ_d^m, and σ_s^m are chosen to be 0.3. The CFL number = 0.1 is chosen for a stable computation up to a final time, $T_f = 5.0$ units. In Figure 6A, we illustrate the density profile obtained from the solution of the Euler equations by a fifth-order RKDG scheme along with the subcell limiter. With three levels of refinement, secondary billows are effectively captured in our simulation. Figure 6B depicts the adapted grid layout along with the troubled cells in the whole domain, which shows a maximum level of refinements around the turbulent region. Here the number of leaf cells at the final time is 50 438, which is about 37% of the effective resolution.

5.1.3 | Shu-Osher problem

Performance of a limiting strategy can be better tested if the solution contains both the intricate smooth solution features as well as strong discontinuities. The Shu-Osher test problem[51] is one of the examples of such kind. In this test problem,



(A) (B)

FIGURE 7 A, Surface plot of density for the two-dimensional Shu-Osher problem at a final time $t = 1.8$ by a fifth-order discontinuous Galerkin method on 5224 primary triangular elements and maximum two levels of primary grid refinements. Red patches show the troubled cells at the final time; B, This panel plots the computed density values (blue circles) on the $y = 0$ line to compare with the reference solution (plotted using a black solid line) obtained from the solution of the one-dimensional Shu-Osher problem [Colour figure can be viewed at wileyonlinelibrary.com]

we solve for the interaction of a moving Mach 3 shock front with a sinusoidally varying density profile, given by the following initial state of primitive variables:

$$(\rho_0, u_0, p_0) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if} \quad x \leqslant -4 \\ (1 + 0.2\sin(5x), 0, 1) & \text{otherwise.} \end{cases} \tag{41}$$

For the 2-D case, this problem is considered on the domain $[-5, 5] \times [-1, 1]$, which is discretized with 5224 triangular elements. Maximum two levels of refinement are employed, corresponding to an effective resolution of 334 336 number of cells. All four refinement and derefinement parameters are chosen to be 0.3. At both the boundaries at $x = -5$ and $x = 5$, the nonreflecting boundary condition is applied, whereas a periodic boundary condition is applied at the other two boundaries. For a stable computation of this specified problem setup, the CFL number = 0.05 is chosen. Figure 7A highlights the density variation near to the shock front at the final time along with the regions on which limiter acts. It is worth noting that AMR uses only 17% of the effective grid resolution at the final time, achieving a high level of computational efficiency by this AMR approach. Figure 7B compares the density profile on the $y = 0$ line in the domain with the reference solution obtained from the solution of the one-dimensional Euler equations. In Figure 7B, numerical values of density at the final time are obtained by sampling on 100 uniformly distributed points on the $y = 0$ line between $x = -5$ and $x = 5$.

5.1.4 | Double Mach reflection problem

This problem consists of the evolution of a self-similar shock structure with two triple points when a vertical shock front of Mach 10 hits a ramp, inclined at an angle 30^o with the horizontal line.[52] At the primary triple point, a primary mach stem and a reflected shock front meet with the incident shock. Due to the breaking up of the reflected shock front, a secondary triple point appears where a bow structured secondary mach stem, and primary and secondary reflected shock fronts meet. When the secondary reflected shock front meets the primary slip line (which emanates from the primary triple point), an interesting fine vortex sheet roll-up appears on the primary slip line. It is challenging for any higher order method to capture all the shock structures sharply and at the same time this fine vortex sheet on the primary slip line. We solve the Euler equations with a fifth-order RKDG method on the domain $[0, 4] \times [0, 1]$. A Mach 10 shock, inclined at an angle 60^o with the horizontal line is placed at the point $(1/6, 0)$. At the top boundary of this domain, time-dependent Dirichlet boundary condition tracks exactly the location of the incident shock front. At the left and right boundaries, inflow and outflow boundary conditions are applied. At the bottom boundary, the postshock boundary condition is applied before the initial location of the incident shock. A reflecting boundary condition is applied on the rest of the bottom boundary (for more details of the setup for this problem and analytical treatment, refer to the work of Kem[53] and references therein). The number of primary triangles used to run this simulation is 5947. Maximum three levels of refinement of any primary element are employed, which corresponds to an effective resolution of 380 608 triangular elements. All four grid refinement and derefinement parameters are chosen to be 0.2. The CFL number = 0.05 is assigned for this problem. The most significant portion of the domain is presented for the visualization of the density profile at a final time, $T_f = 0.2$ in Figure 8A, showing the small-scale structures on the primary slip line. Figure 8B depicts the adaptively refined primary grid layout in the domain at the final time along with the troubled cells. The number of leaf elements at the final time is

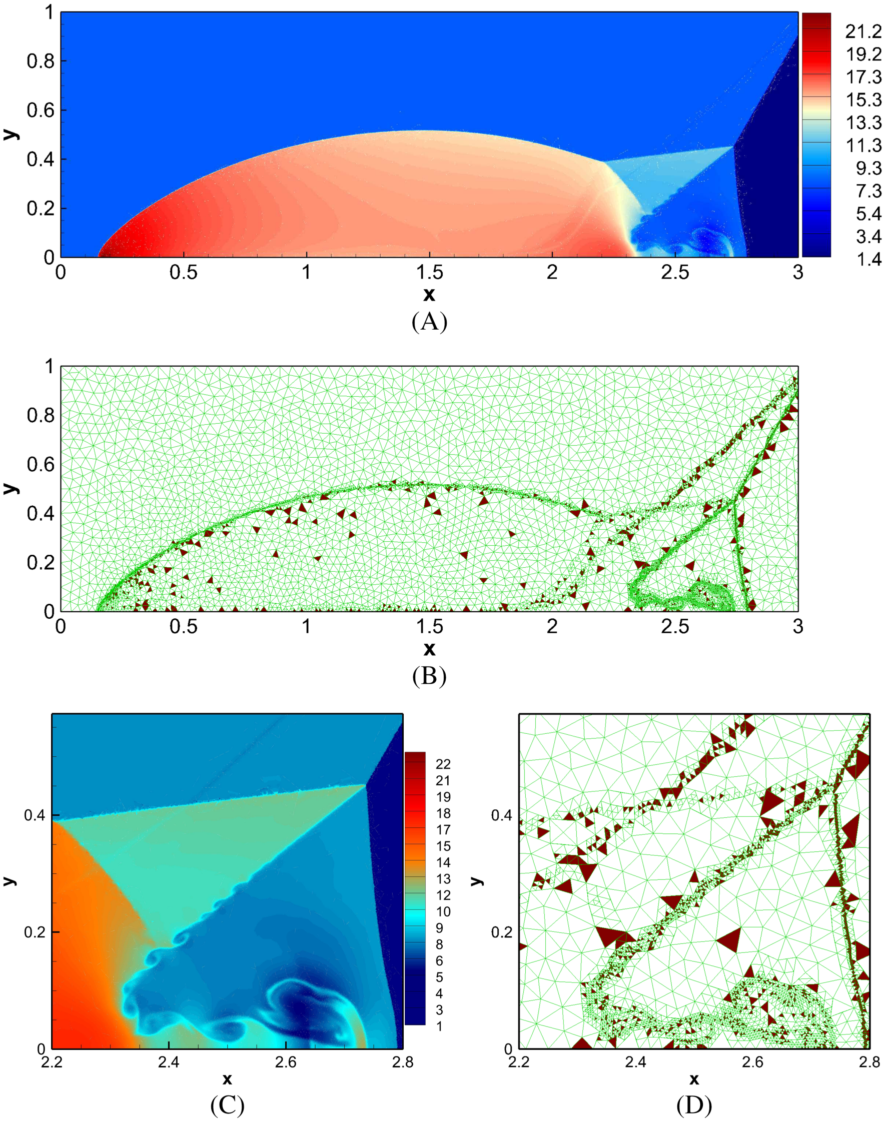


FIGURE 8 A, Density contour for the double Mach reflection problem at the final time $t = 0.2$ by a fifth-order discontinuous Galerkin method on the 5947 primary triangular elements and maximum three levels of adaptive mesh refinement; B, Corresponding adaptively refined grid layout; C, Magnified view of panel A around the primary slip line; D, Corresponding magnified adaptively refined grid layout. Red patches indicate the troubled cells [Colour figure can be viewed at wileyonlinelibrary.com]

13 294, which is approximately only 3.493% of the effective grid resolution in this particular case. Figures 8C and 8D show the magnified view of the density contour around the primary slip line and grid layout on that portion of the domain, respectively.

5.1.5 | Forward facing step

In this test problem, we investigate a supersonic flow over a forward-facing step in a Mach 3 wind tunnel. This is an extensively studied benchmark problem, proposed by Woodward and Colella.[52] A triple point structure appears after a bow shock reflects from the forward step and then from the top wall. After a short time, a vortex sheet emanates from that triple point and sustains. Similar to the previous test case, it is challenging to capture sharp shock structures as well as vortex sheet roll-up by a high-order method. We set up this problem exactly in the same way as in the work of Woodward and Colella.[52] This simulation is run up to a final time of four units and on only 6962 primary triangles. Maximum three levels of grid refinement of any primary cell are specified, which corresponds to 445 568 effective triangular cells. All four grid refinement and derefinement parameters are chosen to be 0.2. The CFL number to simulate this problem is chosen to be 0.1. Figure 9A shows the density contour at the final time illustrating the ability of our AMR approach and the subcell limiting to capture vortex sheet roll-up nicely along with strong shock fronts. Figure 9B demonstrates the refined primary grid layout at the final time along with the troubled cells. For this particular case, only 3.39% of effective resolution is used at the final time by this AMR strategy.

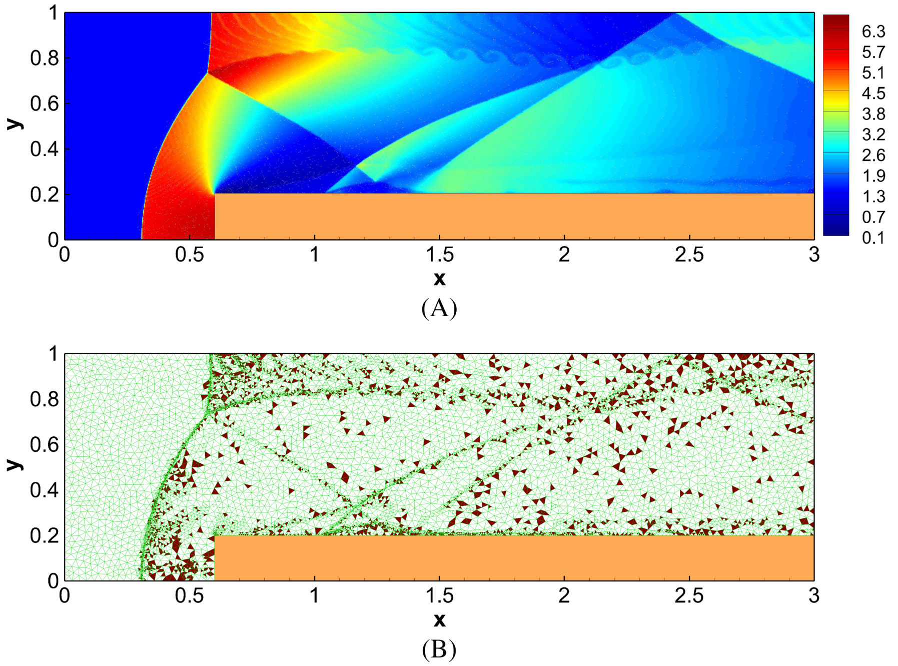


FIGURE 9 A, Density contour for the forward facing step problem at the final time $t = 4$ by a fifth-order discontinuous Galerkin method on 6962 primary triangular elements and maximum three levels of adaptive mesh refinement. B, Corresponding adaptively refined grid layout. Red patches indicate the troubled cells [Colour figure can be viewed at wileyonlinelibrary.com]

5.1.6 | Two-dimensional Riemann problems

The 2-D Riemann problem is one of the most extensively studied problems both analytically and numerically. Unlike one-dimensional Riemann problem, 2-D Riemann problem exhibits much more intricate flow structures in gas dynamics simulations. Numerically those are good tests to check the ability of a higher-order scheme to capture complex interactions between various wave patterns (shocks, contact, discontinuities, and rarefaction waves). In this section, we consider configurations (6) and (12) of the work of Kurganov and Tadmov.[54] The initial conditions on $[0, 1]^2$ square domain are given by

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (1, 0.75, -0.5, 1) & \text{if} \quad x \geqslant 0.5 \ \text{ and } \ y \geqslant 0.5 \\ (2, 0.75, 0.5, 1) & \text{if} \quad x < 0.5 \ \text{ and } \ y \geqslant 0.5 \\ (1, -0.75, 0.5, 1) & \text{if} \quad x < 0.5 \ \text{ and } \ y < 0.5 \\ (3, -0.75, -0.5, 1) & \text{otherwise.} \end{cases} \tag{42a}$$

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (0.5313, 0, 0, 0.4) & \text{if} \quad x \geqslant 0.5 \ \text{ and } \ y \geqslant 0.5 \\ (1, 0.7276, 0, 1) & \text{if} \quad x < 0.5 \ \text{ and } \ y \geqslant 0.5 \\ (0.8, 0, 0, 1) & \text{if} \quad x < 0.5 \ \text{ and } \ y < 0.5 \\ (1, 0, 0.7276, 1) & \text{otherwise.} \end{cases} \tag{42b}$$

To simulate those cases, 6768 primary triangular elements are used along with maximum three levels of mesh refinement, which corresponds to an effective resolution of 433 152 cells. All four refinement and derefinement parameters are chosen to be 0.3 for those two cases. The CFL = 0.1 is specified for both those cases. Right panels of Figure 10 show the density contours for the first and second Riemann problems at final times = 3.5 and 2.5, respectively. The left panels of Figure 10 show the corresponding adaptive refined grid layouts along with the troubled cells. It turns out that, to simulate the first Riemann problem, only about 4.79% of effective grid resolution needs to be used by the AMR strategy. For the second Riemann problem that is only about 3.61% of the effective grid resolution. We see our adaptive mesh refinement efficiently refines the grid locally where the solution develops intricate flow features.

5.1.7 | Sedov-Taylor blast wave problem

We now consider a simulation of the strong cylindrical blast wave, which propagates supersonically in a uniform gas medium. To set up a strong blast wave, a large amount of energy E_0 is injected into a pointlike region. In our present

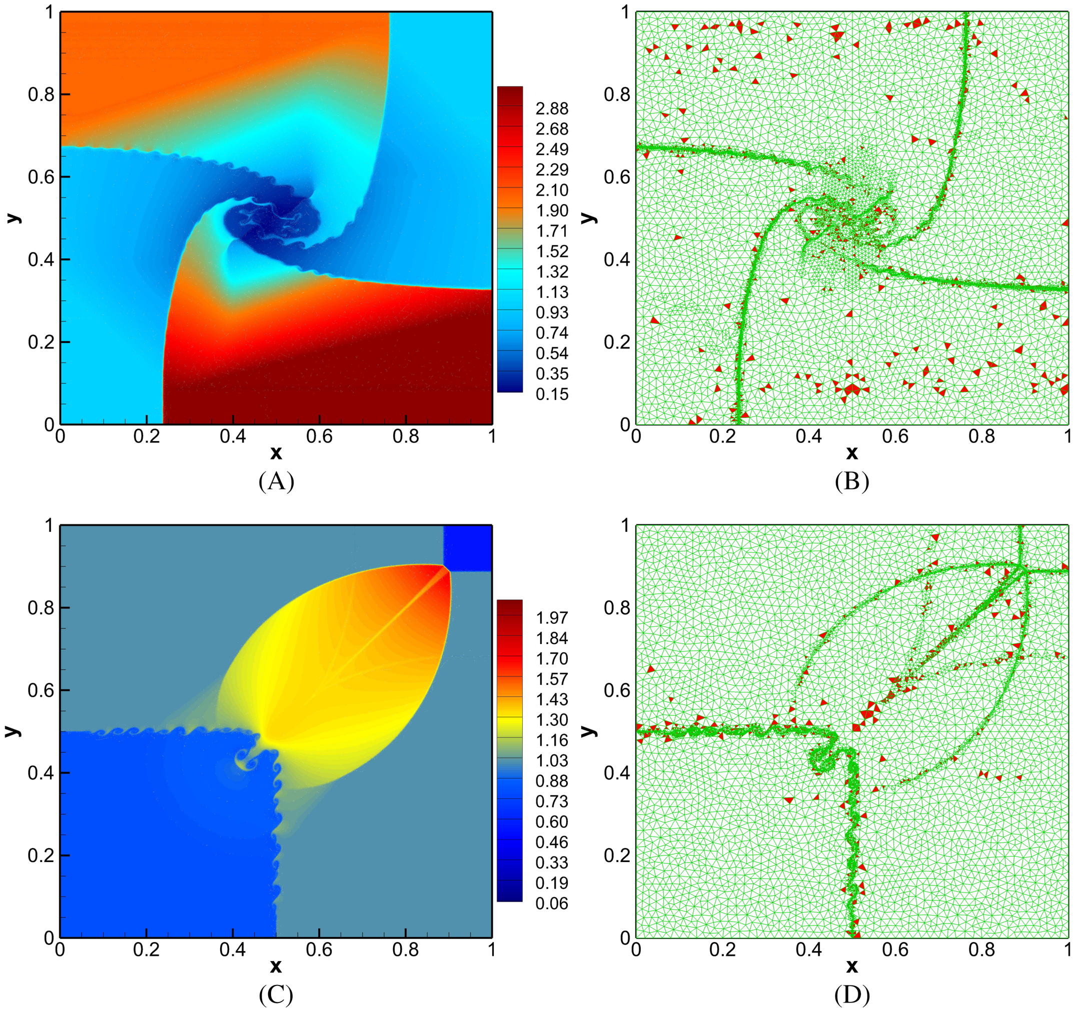


FIGURE 10 A, Density contour for the first two-dimensional Riemann problem at the final time $t = 0.35$ by a fifth-order discontinuous Galerkin (DG) method; B, Corresponding adaptively refined grid layout; C, Density contour for the second two-dimensional Riemann problem at the final time t=0.25 by a fifth-order DG method; D, Corresponding adaptively refined grid layout of the panel A; D, Corresponding magnified adaptively refined grid layout. Both the problems are run on 6768 primary triangular elements and maximum three levels of adaptive mesh refinement. Red patches indicate the troubled cells [Colour figure can be viewed at wileyonlinelibrary.com]

simulation, E_0 is set to be 1.5 erg within a small region at the center of the domain having radius 0.02 unit. To assure supersonic expansion of the blast wave front, ambient pressure p_0 is chosen to be low (4×10^{-13} in our present simulation). Density ρ_0 is set to unity everywhere in the domain of interest. This problem admits a self-similar solution, derived first independently by John von Neumann, Leonid Sedov, and Sir G.I. Taylor. Theoretically, the cylindrical shock front location at any time instant t is predicted by the expression, $r_s = (\alpha E_0 t^2/\rho_0)^{1/4}$, where α is a constant depending on the geometry of the problem (planer, cylindrical or spherical) and the ratio of specific heats of the surrounding media γ (for more details on the analytical treatment, see the work of Sedov[55]). In this particular case, $\alpha = 1.0162$ as we consider γ to be 1.4. The final time for our simulation is set to be unity.

This problem is set with 4542 triangular cells having smaller elements near to the center to facilitate initial large pressure at the center of the square domain $[-2, 2]^2$. Maximum two levels of refinement are specified with all four refinement and derefinement parameters as 0.2. As our simulation is stopped well before the supersonic shock front reaches the outer boundary of the domain, boundary conditions are irrelevant in this present problem. The CFL number is taken to be 0.1 to stabilize the computation of a fifth-order RKDG scheme.

Figure 11A shows the density contour at the final time, and Figure 11B illustrates the adaptive grid layout along with the troubled cells at the same time showing a high degree of grid refinements near to the shock location. In the bottom panels of Figure 11, we compare the primitive variables obtained from our simulation on the $y = 0$ line with the analytical

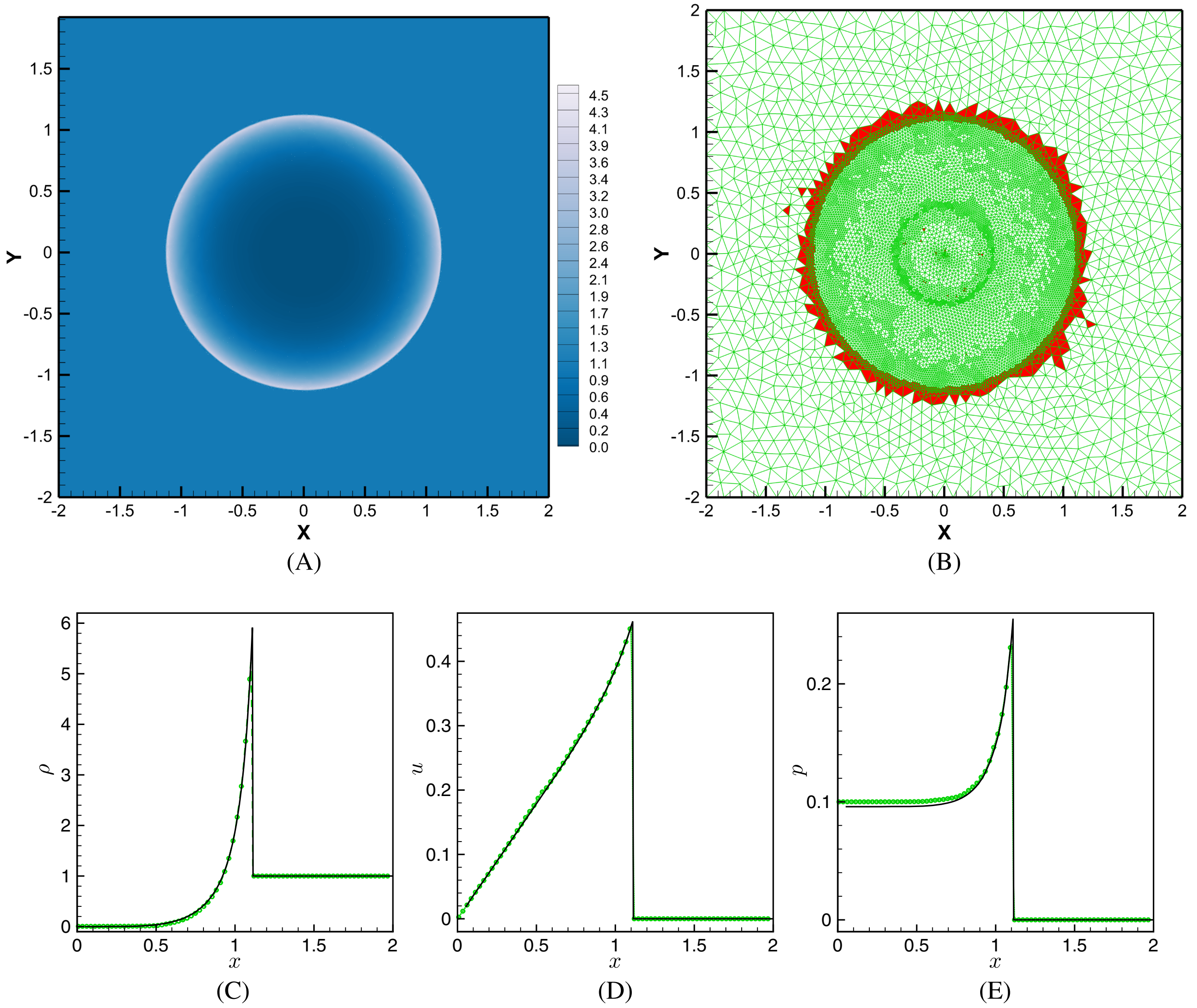


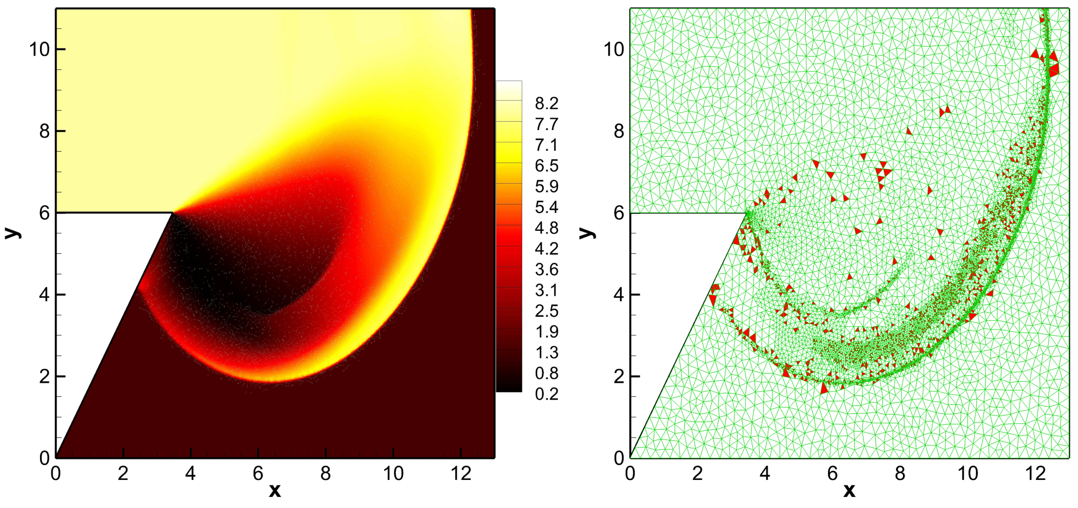
FIGURE 11 A, Density contour for the Sedov-Taylor blast wave problem at the final time, t=1 on 4542 number of primary triangular elements and maximum three levels of adaptive mesh refinement by a fifth order DG method; B, corresponding adaptively refined grid layout. Bottom panels compare the density, x-component of velocity and pressure (plotted with green circles) obtained numerically with the exact solutions (black solid lines). Red patches indicate the troubled cells [Colour figure can be viewed at wileyonlinelibrary.com]

solution obtained using the codes of Kamm and Timmes.[56] We see that the numerically calculated pressure does not agree well with the analytical one towards the center of the cylindrical shock front. We suspect that the inaccuracy associated with the initialization of the pressure field within a small number of triangular cells near the domain center is the reason.

5.1.8 | Shock wave diffraction at a convex corner

We simulate a Mach 10 shock wave diffraction at a convex corner, which is a well-studied benchmark problem in computational fluid dynamics. It is challenging to develop a stable numerical scheme for a high-order RKDG due to the development of a low pressure region near to the 120^o convex corner.[57] We consider a right-moving Mach 10 shock initially located at $x = 3.4$, moving in an undisturbed fluid medium with $\rho = 1.4$ and $p = 1$. The domain of interest is shown in Figure 12.

To simulate this problem, 5081 primary triangular elements are used along with maximum three levels of mesh refinement, which corresponds to an effective resolution of 325 184 cells. All four refinement and derefinement parameters are chosen to be 0.2 for this case. The CFL = 0.1 is specified. Right panels of Figure 12 show the density contour for this problem at a final time = 0.9. The left panel of Figure 12 shows the corresponding adaptively refined grid layout along with the troubled cells. It turns out that, to simulate this problem, only about 3.76% of effective grid resolution needs to be used by the AMR strategy. The result matches well with the same reported in the work of Liu et al.[57]



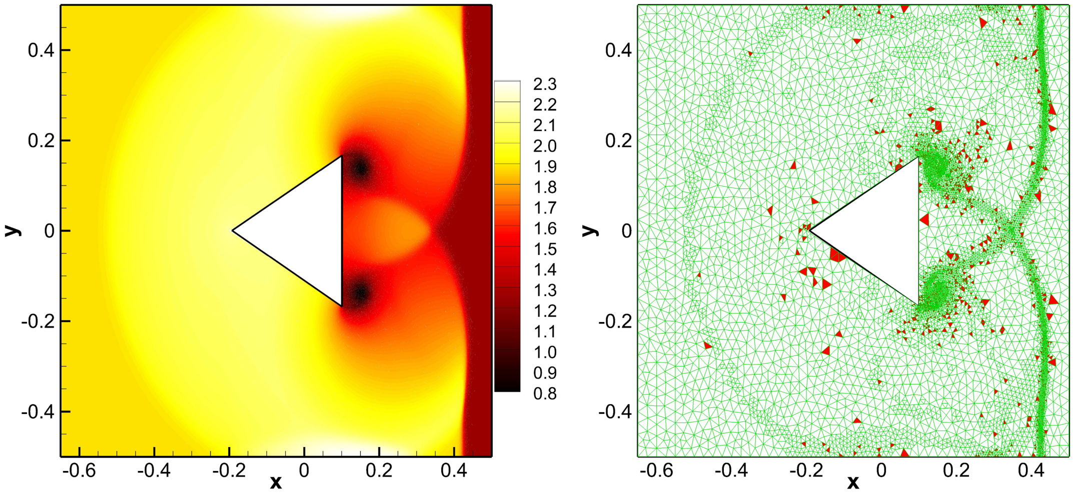
(A) (B)

FIGURE 12 A, Density contour for the Mach 10 shock diffraction at a convex corner at the final time $t = 0.9$ on 5081 primary triangular elements and maximum three levels of adaptive mesh refinement by a fifth-order discontinuous Galerkin method; B, Corresponding adaptively refined grid layout [Colour figure can be viewed at wileyonlinelibrary.com]

5.1.9 | Schardin's Problem: Shock waves pass a finite wedge

The purpose of this test case is to demonstrate the potential of the present scheme with adaptively refined unstructured grid for solving problems on complex domains. In this problem, we simulate a Mach 1.3 shock passing an equilateral triangular solid body.[57] The domain of interest is the region [-0.65,0.5]×[-0.5,0.5] with a triangular body having the coordinates of its vertices (-0.2,0), (0.1,-1/6) and (0.1,1/6). The undisturbed upstream condition of the Mach 1.3 shock is given by $\rho = 1.225\, kg/m^3$ and $p = 101\,325 Pa$. Inflow and outflow boundary conditions are applied at the leftmost and rightmost boundaries, respectively. The remaining boundaries are considered reflecting walls. The initial shock location is at $x = -0.55$.

To simulate this problem, 4674 primary triangular elements are used along with maximum three levels of mesh refinement, which corresponds to an effective resolution of 299 136 cells. All four refinement and derefinement parameters are chosen to be 0.2 for this case. The CFL = 0.1 is specified. The right panels of Figure 13 show the density contour for this problem at a final time = 2.2 $\times 10^{-3}$. The left panel of Figure 13 shows the corresponding adaptively refined grid layout



(A) (B)

FIGURE 13 A, Density contour for the Schardin's problem at the final time $t = 2.2 \times 10^{-3}$ on 4674 primary triangular elements and maximum three levels of adaptive mesh refinement by a fifth-order discontinuous Galerkin method; (B) corresponding adaptively refined grid layout [Colour figure can be viewed at wileyonlinelibrary.com]

along with the troubled cells. It turns out that, to simulate this problem only about 5.14% of effective grid resolution needs to be used by the AMR strategy.

6 | CONCLUSION

In this work, we implemented and investigated the performance of a fourth-order and fifth-order RKDG method with an AMR to solve inviscid compressible flow problems. Simulations are performed with unstructured triangular grid elements, which make our implementation suitable for compressible flow problems on 2-D complex domains. A finite volume subcell limiter is used to suppress spurious oscillations due to the presence of shocks. Troubled cells are detected based on a few after-the-fact detection criteria, which was suggested by Dumbser and Loubere[23] and a few extra new conditions enabling a reduction in troubled cell count in expense of only a slight increase in computational cost. Moreover, this modification of the solution admissibility criterion in our present work enables us to retain the compactness of the overall scheme. Although there seems to have an everlasting room for improvements, our tests show some promising results with this modified subcell limiter along with a cell-by-cell AMR.

ACKNOWLEDGEMENTS

The research is partly supported by NSAF grant U1630247 and National Natural Science Foundation of China, grant 11571290. The authors thank the anonymous reviewers for their many constructive criticisms to improve this paper.

ORCID

Jianxian Qiu https://orcid.org/0000-0002-3479-0536

REFERENCES

1. Reed WH, Hill TR. *Triangular Mesh Methods for Neutron Transport Equation*. Technical Report LA-UR-73-479. New Mexico: Los Alamos Scientific Laboratory; 1973.
2. Cockburn B, Shu C-W. The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *J Comput Phys*. 1991;25:337-361.
3. Cockburn B, Shu C-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math Comput*. 1989;52:411-435.
4. Cockburn B, Lin S-Y, Shu C-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *J Comput Phys*. 1989;84:90-113.
5. Cockburn B, Hou S, Shu C-W. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math Comput*. 1990;54:545-581.
6. Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *J Comput Phys*. 1998;141:199-224.
7. Biswas R, Devine KD, Flaherty JE. Parallel, adaptive finite element methods for conservation laws. *Appl Numer Math*. 1994;14:255-283.
8. Burbeau A, Sagaut P, Bruneau CH. A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods. *J Comput Phys*. 2001;169:111-150.
9. Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. *J Comput Phys*. 1996;126:202-228.
10. Liu X-D, Osher S, Chan T. Weighted essentially non-oscillatory schemes. *J Comput Phys*. 1994;115:200-212.
11. Qiu J, Shu C-W. Runge–Kutta discontinuous Galerkin method using WENO limiters. *SIAM J Sci Comput*. 2005;26(3):907-929.
12. Zhu J, Qiu J, Shu C-W, Dumbser M. Runge–Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. *J Comput Phys*. 2008;227:4330-4353.
13. Qiu J, Shu C-W. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one dimensional case. *J Comput Phys*. 2003;193:115-135.
14. Qiu J, Shu C-W. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two dimensional case. *Comput Fluids*. 2005;34:642-663.
15. Zhu J, Qiu J. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: unstructured meshes. *J Sci Comput*. 2009;39:293-321.
16. Zhu J, Qiu J. Runge-Kutta discontinuous Galerkin method using WENO-type limiters: three-dimensional unstructured meshes. *Commun Comput Phys*. 2012;11(3):985-1005.
17. Shu C-W. High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review*. 2009;51:82-126.
18. Zhong X, Shu C-W. A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods. *J Comput Phys*. 2013;232:397-415.

19. Zhu J, Zhong X, Shu C-W, Qiu J. Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *J Comput Phys*. 2013;248:200-220.
20. Zhu J, Zhong X, Shu C-W, Qiu J. Runge-Kutta discontinuous Galerkin method with a simple and compact Hermite WENO limiter. *Commun Comput Phys*. 2016;19(4):944-969.
21. Zhu J, Zhong X, Shu C-W, Qiu J. Runge-Kutta discontinuous Galerkin method with a simple and compact Hermite WENO limiter on unstructured meshes. *Commun Comput Phys*. 2017;21(3):623-649.
22. Dumbser M, Zanotti O, Loubre R, Diot S. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J Comput Phys*. 2014;278:47-75.
23. Dumbser M, Loubere R. A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *J Comput Phys*. 2016;319:163-199.
24. Suresh A, Huynh HT. Accurate monotonicity-preserving schemes with Runge–Kutta time stepping. *J Comput Phys*. 1997;136:83-99.
25. Rider WJ, Margolin LG. Simple modifications of monotonicity-preserving limiters. *J Comput Phys*. 2001;174:473-488.
26. Harten A. ENO schemes with subcell resolution. *J Comput Phys*. 1989;83:148-184.
27. Qiu J, Shu C-W. A comparison of troubled cell indicators for Runge–Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters. *SIAM J Sci Comput*. 2005;27(3):995-1013.
28. Krivodonova L, Xin J, Remacle J-F, Chevaugeon N, Flaherty JE. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl Numer Math*. 2004;48:323-338.
29. Clain S, Diot S, Loubère R. A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (MOOD). *J Comput Phys*. 2011;230(10):4028-4050.
30. Diot S, Clain S, Loubère R. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Comput Fluids*. 2012;64:43-63.
31. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *J Comput Phys*. 1984;53:484-512.
32. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J Comput Phys*. 1989;82:64-84.
33. Baeza A, Mulet P. Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations. *Int J Numer Methods Fluids*. 2006;52:455-471.
34. Khokhlov AM. Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations. *J Comput Phys*. 1998;143(2):519-543.
35. Dumbser M, Zanotti O, Hidalgo A, Balsara DS. ADER-WENO finite volume schemes with space-time adaptive mesh refinement. *J Comput Phys*. 2013;248:257-286.
36. Dumbser M, Hidalgo A, Zanotti O. High order space–time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. *Comput Methods Appl Mech Eng*. 2014;268:359-387.
37. Baumann CE, Oden JT. A discontinuous *hp* finite element method for convection—diffusion problems. *Comput Methods Appl Mech Eng*. 1999;175:311-341.
38. Baumann CE, Oden JT. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *Int J Numer Methods Fluids*. 1999;31:79-95.
39. Houston P, Schwab C, Sli E. Stabilized *hp*-finite element methods for first-order hyperbolic problems. *SIAM J Numer Anal*. 2000;37(5):1618-1643.
40. Houston P, Sli E. *hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM J Sci Comput*. 2002;23(4):1226-1252.
41. Luo H, Baum JD, Löhner R. A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *J Comput Phys*. 2008;227:8875-8893.
42. Fambri F, Dumbser M. Semi-implicit discontinuous Galerkin methods for the incompressible Navier–Stokes equations on staggered adaptive Cartesian grids. *Comput Methods Appl Mech Eng*. 2017;324:170-203.
43. Zhu H, Qiu J. An *h*-adaptive RKDG method with troubled-cell indicator for two-dimensional hyperbolic conservation laws. *Adv Comput Math*. 2013;39:445-463.
44. Taylor MA, Wingate BA, Bos LP. A cardinal function algorithm for computing multivariate quadrature points. *SIAM J Numer Anal*. 2007;45:193-205.
45. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. New York, NY: Springer; 2007.
46. Dubiner M. Spectral methods on triangles and other domains. *J Sci Comput*. 1991;6:345-390.
47. Gottlieb S, Shu C-W. Total variation diminishing Runge-Kutta schemes. *Math Comput*. 1998;67:73-85.
48. Cockburn B, Shu C-W. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J Sci Comput*. 2001;16: 173-261.
49. Yee HC, Sandham ND, Djomehri MJ. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *J Comput Phys*. 1999;150:199-238.
50. Springel V. *E pur si muove*: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon Notices Royal Astron Soc*. 2010;401:791-851.
51. Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J Comput Phys*. 1989;83:32-78.
52. Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. *J Comput Phys*. 1984;54:115-173.
53. Kemm F. On the proper setup of the double Mach reflection as a test case for the resolution of gas dynamics codes. *Comput Fluids*. 2016;132:72-75.

54. Kurganov A, Tadmor E. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers. *Numer Methods Partial Differ Equ*. 2002;18(5):584-608.
55. Sedov LI. *Similarity and Dimensional Methods in Mechanics*. New York, NY: Academic Press; 1959.
56. Kamm JR, Timmes FX. *On Efficient Generation of Numerically Robust Sedov Solutions*. Technical Report LA-UR-07-2849. New Mexico: Los Alamos National Laboratory; 2007.
57. Liu J, Qiu J, Goman M, Li X, Liu M. Positivity-preserving Runge-Kutta discontinuous Galerkin method on adaptive Cartesian grid for strong moving shock. *Numer Math: Theory Methods Appl*. 2016;9(5):87-110.
58. Golub GH, Van Loan CF. *Matrix Computations*. 4th ed. Baltimore, MD: Johns Hopkins University Press; 2013.

How to cite this article: Giri P, Qiu J. A high-order Runge-Kutta discontinuous Galerkin method with a subcell limiter on adaptive unstructured grids for two-dimensional compressible inviscid flows. *Int J Numer Meth Fluids*. 2019;91:367–394. https://doi.org/10.1002/fld.4757

APPENDIX A

THE PRORIOL POLYNOMIALS AND QUADRATURE ON A TRIANGULAR ELEMENT

Equation (15) defines the Proriol polynomials on the standard square. On the standard triangle, (ie, on the $\zeta - \eta$ plane), explicit expressions for first few Proriol polynomials are given by

$$\Phi_{00} = 1, \tag{A1a}$$

$$\Phi_{10} = 2\zeta + \eta - 1, \tag{A1b}$$

$$\Phi_{01} = 3\eta - 1, \tag{A1c}$$

$$\Phi_{20} = 6\zeta^2 + 6\zeta\eta + \eta^2 - 6\zeta - 2\eta + 1, \tag{A1d}$$

$$\Phi_{11} = (2\zeta + \eta - 1)(5\eta - 1), \tag{A1e}$$

$$\Phi_{02} = 10\eta^2 - 8\eta + 1. \tag{A1f}$$

The orthogonality relation for those Proriol polynomials on the standard triangle is given by

$$\int_{\Omega_{\boldsymbol{\xi}}} \Phi_{ij}\Phi_{kl}\, d\zeta d\eta = \frac{\delta_{ik}\delta_{jl}}{2(2i+1)(i+j+1)}, \tag{A2}$$

where δ_{ik} and δ_{jl} are the Kronecker delta.

The numerical integration of a function on any triangular element is approximated by first changing the variable of integration from $\mathbf{x}$ to $\boldsymbol{\xi}$.

$$\int_T f(\mathbf{x})\, dxdy = \int_{\Omega_{\boldsymbol{\xi}}} f(\mathbf{x})|\mathbf{J}|\, d\zeta d\eta = 2|T| \int_{\Omega_{\boldsymbol{\xi}}} f(\mathbf{x})\, d\zeta d\eta. \tag{A3}$$

Next, the integration on the standard triangle can be approximated by using an approximate quadrature rule.

$$\int_{\Omega_{\boldsymbol{\xi}}} f(\mathbf{x})\, d\zeta d\eta \approx \sum_{i=1}^{N_q(m)} f\left(\mathbf{x}_i^q\right) w_i^q, \tag{A4}$$

where $\mathbf{x}_i^q$ and w_i^q are the quadrature points in T and corresponding quadrature weights. In the present paper, we have calculated the quadrature rule for integrating a function of arbitrary order by following the *cardinal function algorithm* by Taylor et al.[44] Note that $N_q(m)$ here is chosen such that the integration is exact for $f(\mathbf{x}) \in \mathbb{P}^{2m}$.

Determinant of the Jacobian matrix for the transformation (12) can be derived by

$$|\mathbf{J}| = \begin{vmatrix} \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \begin{vmatrix} (x_1 - x_0) & (y_1 - y_0) \\ (x_2 - x_0) & (y_2 - y_0) \end{vmatrix} = 2|T|. \tag{A5}$$

Finally, to calculate the derivative of a function given in terms of linear combination of the Proriol polynomials on any triangular cell T, at first, we need to derive the formula for calculating the gradient of the Proriol polynomials on the standard triangle. By the chain rule, derivatives of the Proriol polynomials can be calculated from the following equations:

$$\frac{\partial\Phi_{kl}}{\partial\zeta} = \begin{cases} 0 & \text{if} \quad k = 0, \\ 2\left(\frac{1-\tilde{\eta}}{2}\right)^{k-1} \frac{dP_k^{(0,0)}(\tilde{\zeta})}{d\tilde{\zeta}} P_l^{(2k+1,0)}(\tilde{\eta}) & \forall\, k > 0, \end{cases} \tag{A6a}$$

$$\frac{\partial\Phi_{kl}}{\partial\eta} = \begin{cases} 2\frac{dP_l^{(1,0)}(\tilde{\eta})}{d\tilde{\eta}} \qquad \text{if } k = 0, & \\ \left(\frac{1-\tilde{\eta}}{2}\right)^{k-1} \left((1+\tilde{\zeta})\frac{dP_k^{(0,0)}(\tilde{\zeta})}{d\tilde{\zeta}} - kP_k^{(0,0)}(\tilde{\zeta})\right) P_l^{(2k+1,0)}(\tilde{\eta}) & \\ \qquad +2\left(\frac{1-\tilde{\eta}}{2}\right)^{k} P_k^{(0,0)}(\tilde{\zeta}) \frac{dP_l^{(2k+1,0)}(\tilde{\eta})}{d\tilde{\eta}} & \forall\, k > 0. \end{cases} \tag{A6b}$$

Subsequently, the gradient of a Proriol polynomial in physical domain is calculated from the following equation:

$$\begin{Bmatrix} \frac{\partial\Phi_{kl}}{\partial x} \\ \frac{\partial\Phi_{kl}}{\partial y} \end{Bmatrix} = \frac{1}{2|T|} \begin{bmatrix} (y_2 - y_0) & -(y_1 - y_0) \\ -(x_2 - x_0) & (x_1 - x_0) \end{bmatrix} \begin{Bmatrix} \frac{\partial\Phi_{kl}}{\partial\zeta} \\ \frac{\partial\Phi_{kl}}{\partial\eta}, \end{Bmatrix} \tag{A7}$$

where (x_0, y_0), (x_1, y_1), and (x_2, y_2) are three vertices of T.

APPENDIX B

RECONSTRUCTION OF A FUNCTION ON T FROM THE SUBCELL AVERAGES

In Section 3, a modified subcell limiter for DG-SEM is outlined. In this section, the reconstruction of DG solution from the subcell averages on a cell T is detailed. More precisely, the problem can be stated as follows. We seek for the vector of moments $\hat{\mathbf{u}} \in \mathbb{R}^n$ such that it minimizes $||\mathbf{A}\hat{\mathbf{u}}{-}\mathbf{b}||$ and at the same time satisfies exactly $\mathbf{B}\hat{\mathbf{u}} = \mathbf{d}$ where $\mathbf{A} \in \mathbb{R}^{k\times n}$, $\mathbf{B} \in \mathbb{R}^{p\times n}$, $\mathbf{b} \in \mathbb{R}^k$, and $\mathbf{d} \in \mathbb{R}^p$. It is also important to note that $k > n$, rank($\mathbf{B}$)=p, and $p < n$. $\hat{\mathbf{u}}$ denotes the vector of moments of the reconstructed DG solution on a specific triangular element and n is the number of moments = $(m+1)(m+2)/2$. The first set of equations, namely $\mathbf{A}\hat{\mathbf{u}} = \mathbf{b}$ arises by imposing the condition: the DG solution recovers the subcell averages in the least squares sense (see Equation (26)). The second set of equations, namely $\mathbf{B}\hat{\mathbf{u}} = \mathbf{d}$, comes from the condition that the cell average calculated from subcell averages and reconstructed DG solution on any triangular element have to be the same (see Equation (27)). We need the second set of equations to be satisfied exactly, whereas the first set of equations, the number of equations being more than the number of unknowns, needs to be satisfied in a least squares sense. The solution methodology is adopted from the work of Golub.[58] We start by calculating the QR decomposition of $\mathbf{B}^T$ of the following form:

$$\mathbf{B}^T = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \Theta \end{bmatrix}, \tag{B1}$$

where $\mathbf{Q}$ is an orthogonal matrix and an element of $\mathbb{R}^{n\times n}$, $\mathbf{R} \in \mathbb{R}^{p\times p}$ is an upper triangular matrix, and $\boldsymbol{\Theta}$ is the null matrix in $\mathbb{R}^{(n-p)\times p}$. With this notation, the second set of equations further simplifies to

$$\mathbf{B}\hat{\mathbf{u}} = \left(\mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \Theta \end{bmatrix}\right)^T \hat{\mathbf{u}} = \begin{bmatrix} \mathbf{R}^T & \Theta^T \end{bmatrix} \mathbf{Q}^T\hat{\mathbf{u}}. \tag{B2}$$

We introduce a new notation, ie,

$$\mathbf{Q}^T\hat{\mathbf{u}} = \begin{Bmatrix} \mathbf{y} \\ \mathbf{z}, \end{Bmatrix} \text{ where } \quad \mathbf{y} \in \mathbb{R}^p \quad \text{and} \quad \mathbf{z} \in \mathbb{R}^{(n-p)}. \tag{B3}$$

Introducing the above notation, Equation (B2) reduces to

$$\mathbf{B}\hat{\mathbf{u}} = \begin{bmatrix} \mathbf{R}^T & \Theta^T \end{bmatrix} \begin{Bmatrix} \mathbf{y} \\ \mathbf{z} \end{Bmatrix} = \mathbf{R}^T\mathbf{y} = \mathbf{d}. \tag{B4}$$

Equation (B4) can be solved conveniently by a direct solver. However, the first set of equation can be reduced to the following set of equations in terms of the transformed variables.

$$\begin{aligned} \mathbf{A}\hat{\mathbf{u}} = (\mathbf{AQ})(\mathbf{Q}^T\hat{\mathbf{u}}) = (\mathbf{AQ})\begin{Bmatrix} \mathbf{y} \\ \mathbf{z} \end{Bmatrix} = \mathbf{A}_1\mathbf{y} + \mathbf{A}_2\mathbf{z} = \mathbf{b}, \\ \Rightarrow \mathbf{A}_2\mathbf{z} = \mathbf{b} - \mathbf{A}_1\mathbf{y}, \end{aligned} \tag{B5}$$

where the matrix $\mathbf{AQ}$ is partitioned into two matrices $\mathbf{A}_1 \in \mathbb{R}^{k\times p}$ and $\mathbf{A}_2 \in \mathbb{R}^{k\times(n-p)}$ as $\mathbf{AQ} = [\mathbf{A}_1|\mathbf{A}_2]$. So, this problem reduces to an unconstrained minimization problem: Find a $\mathbf{z} \in \mathbb{R}^{(n-p)}$ that minimizes $||\mathbf{A}_2\mathbf{z} - (\mathbf{b} - \mathbf{A}_1\mathbf{y})||$ as $\mathbf{y}$ is already known from the solution of Equation (B4). The solution of this unconstrained minimization problem determines $\mathbf{z}$. Finally, $\hat{\mathbf{u}}$ can be calculated from Equation (B3) as

$$\hat{\mathbf{u}} = \mathbf{Q}\begin{Bmatrix} \mathbf{y} \\ \mathbf{z} \end{Bmatrix}. \tag{B6}$$

To accelerate our computation, QR decompositions of $\mathbf{B}^T$ and A_2 are precomputed and stored beforehand for their frequent usage.