A New Runge-Kutta Discontinuous Galerkin Method with Conservation Constraint to Improve CFL Condition for Solving Conservation Laws

Zhiliang Xu [‡], Xu-Yan Chen [¶] and Yingjie Liu [§]

April 24, 2013

Abstract

We present a new formulation of the Runge-Kutta discontinuous Galerkin (RKDG) method [7, 6, 5, 4] for solving conservation Laws. The new formulation requires the computed RKDG solution in a cell to satisfy additional conservation constraint in adjacent cells and does not increase the complexity or change the compactness of the RKDG method. We use this new formulation to solve one-dimensional and two-dimensional conservation laws with piecewise quadratic and cubic polynomial approximation, respectively. The hierarchical reconstruction [13, 25] is applied as a limiter to eliminate spurious oscillations in discontinuous solutions. Numerical computations for scalar and systems of nonlinear hyperbolic conservation laws are performed. We find that: 1) this new formulation improves the CFL number over the original RKDG formulation and thus reduces the overall computational cost; 2) the new formulation improves the robustness of the DG scheme with the current limiting strategy and improves the resolution of the numerical solutions of shock wave problems in multi-dimensions.

1 Introduction

In this paper, we introduce a simple, yet effective technique to improve the Courant-Friedrichs-Lewy (CFL) condition of the Runge-Kutta discontinuous Galerkin (RKDG) method for solving nonlinear conservation laws. The discontinuous Galerkin method (DG) was firstly introduced by Reed and Hill [18] as a technique to solve neutron transport problems. In

[‡](E-mail: zxu2@nd.edu)
Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556. Research was supported in part by NSF grants DMS-1115887 and DMS-0800612 and and NIH grants 1 R01 GM100470-01 and 1 R01 GM095959-01A1.

[§](E-mail: yingjie@math.gatech.edu)
School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332. Research was supported in part by NSF grant DMS-1115671.

[¶](E-mail: xchen@math.gatech.edu)
School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332.

a series of papers by Cockburn, Shu *et al.* [7, 6, 5, 4], the RKDG method has been developed for solving nonlinear hyperbolic conservation laws and related equations. In their formulation, DG is used for spatial discretization with flux values at cell edges computed by either Riemann solvers or monotone flux functions, the total variation bounded (TVB) limiter [19, 7] is employed to eliminate spurious oscillations and the total variation diminishing (TVD) Runge-Kutta (RK) method [21] is used for the temporal discretization to ensure the stability of the numerical approach while simplifying the implementation. The RKDG method has enjoyed great success in solving the Euler equations for gas dynamics, compressible Navier-Stokes equations, viscous MHD equations and many other equations, and motivated many related new numerical techniques.

In [7], the RKDG method is shown to be linearly stable when the CFL factor is bounded by $\frac{1}{2q+1}$ for the second order and the third order schemes in the one-dimensional (1D) space, where q is the degree of the polynomial approximating the solution. It would be desirable if there is a simple technique to increase CFL number of the RKDG method without introducing too much computational overhead while still being compact and maintaining its other nice properties. In this paper, we present such a strategy which is to mix the RKDG method with some of the finite volume reconstruction features (e.g. Abgrall [1]), which are used as the extra constraint imposed on the numerical solution. With this respect, we would like to refer to a recent work of van Leer and Nomura [12], in which the diffusive flux for DG is approximated by using a reconstructed polynomial supported on the union of adjacent cells out of the set of piecewise polynomials originally defined on these cells, respectively. Additionally, it is found in [15] that the central DG scheme on overlapping cells with Runge-Kutta time-stepping can use a CFL number larger than the one that RKDG method can take on non-overlapping cells when the order of accuracy of these schemes is above the first order. Other efforts to improve the CFL condition can be found in [24, 3].

In the present paper, we propose the following idea to improve the CFL number that the RKDG method is allowed to use. We impose additional conservation constraint on the numerical solution computed by the RKDG method in the sense that in addition to let a piecewise approximate solution supported on a cell conserve the cell average of this cell, this solution matches the cell averages of the solution supported on adjacent neighbors of this cell in a least-square sense. This is implemented by introducing a penalty term to the original RKDG formulation. This new idea is a simple (and thus potentially useful) technique connecting the DG and finite volume methods which are both compact. The resulting method is referred to as the constrained RKDG method in the sections that follow. We illustrate the effectiveness of our technique by using the 1D and two-dimensional (2D) third- and fourth-order accurate schemes, respectively. The 2D test cases are solved on triangular meshes. For nonlinear test problems, we compute both smooth and discontinuous solutions. Further study on higher-order accurate cases and theoretical analysis will be reported in the future.

Using finite volume limiting techniques on solutions computed by the RKDG method for conservation laws has been explored by many researchers. In [17, 28], the WENO finite volume reconstruction procedures are used as the limiter on cells where the solutions supported on these cells become oscillatory. In [16], Luo *et al.* developed a Hermite WENO-based limiter for the second order RKDG method on unstructured meshes following [17]. Since the RKDG method is a compact method, it would be ideal to use a compact limiting technique.

It is a challenging task to use adjacent high order information in the limiting procedure to remove spurious oscillations in the vicinities of discontinuities while preserving high resolution. The first of such limiters is the TVB projection limiter by Cockburn and Shu, which uses the lowest and (limited) first Legendre moments locally where non-smoothness is detected. Other compact limiting techniques which are supposed to remove spurious oscillations using information only from adjacent cells for any orders include the moment limiter [2] and the recently developed hierarchical reconstruction (HR) [13]. Besides the above related techniques, there are also many research works of compact limiters for high order schemes on various problems. HR as a limiting technique can be applied without using local characteristic decomposition. One goal of the paper is to verify if our technique for improving the CFL number of RKDG works well with HR. In [25], HR on 2D triangular meshes has been studied for the piecewise quadratic DG method; a partial neighboring cell technique has been developed and a component-wise WENO-type linear reconstruction is used on each hierarchical level. This new technique has good resolution and accuracy on unstructured meshes and is easy to implement since the weights on each hierarchical level are trivial to compute and essentially independent of the mesh.

In this study, we find that the constrained RKDG method increases the CFL number over the original RKDG method, and further improves the resolution of the numerical solutions limited by HR. For the 2D third order accuracy case, it also reduces the magnitude of numerical errors.

We also point out that the computer memory requirement for the constrained RKDG method is the same as that for the RKDG method. The computer memory utilized by both methods is mainly for storing the degrees of freedom for each cell. Thus we do not perform a study on this aspect.

The paper is organized as follows. Section 2 describes the conservation constrained RKDG formulation and summarizes the limiting procedure. Results of numerical tests are presented in Section 3. Concluding remarks and a plan for the future work are included in Section 4.

2 Formulation of the method

In this section, we formulate the conservation constrained Runge-Kutta discontinuous Galerkin finite element method for solving time dependent hyperbolic conservation laws (2.1)

$$\begin{cases} \frac{\partial u_k}{\partial t} + \nabla \cdot \mathbf{F}_k(\mathbf{u}) = 0 \ , \quad k = 1, .., p, \quad \text{in } \Omega \times (0, T) \ , \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \ , \end{cases} \tag{2.1}$$

where $\Omega \subset R^d$, $\mathbf{x} = (x_1, ..., x_d)$, d is the dimension, $\mathbf{u} = (u_1, ..., u_p)^T$ and the flux vectors $\mathbf{F}_k(\mathbf{u}) = (F_{k,1}(\mathbf{u}), ..., F_{k,d}(\mathbf{u}))$.

The method of lines approach is used to evolve the solution in time. The third- and fourth-order accurate TVD Runge-Kutta time-stepping methods are used for the test problems presented in the paper, respectively. At each time level the semi-discrete constrained DG method is used for spatial discretization. In the vicinities of discontinuities of the solution, the computed piecewise polynomial solution is reconstructed by the hierarchical reconstruction to remove spurious oscillations.

2.1 Conservation constrained discontinuous Galerkin Method

We describe the conservation constrained DG formulation here. First, the physical domain Ω is partitioned into a collection of $\mathcal{N}$ non-overlapping cells $\mathcal{T}_h = \{\mathcal{K}_i : i = 1, ..., \mathcal{N}\}$ so that $\Omega = \bigcup_{i=1}^{\mathcal{N}} \mathcal{K}_i$. In 2D, we use triangular meshes and for simplicity, we assume that there are no hanging nodes. Let the basis function set which spans the finite element space on cell $\mathcal{K}_i$ be

$$\mathcal{B}_i = \{\phi_m(\mathbf{x}) : m = 0, ..., r\} \ . \tag{2.2}$$

In the present study, we choose the basis function set to be a polynomial basis function set of degree q in a cell $\mathcal{K}_i$, which consists of the monomials of multi-dimensional Taylor expansions about the cell centroid scaled by the area of the cell. For instance, for a 2D triangular cell $\mathcal{K}_i$, the basis function set (2.2) in the (x, y) coordinate is

$$\begin{aligned} \mathcal{B}_i &= \{\phi_m(x - x_i, y - y_i) : m = 0, ..., r\} \\ &= \Big\{1, (x - x_i)/\sqrt{|\mathcal{K}_i|}, (y - y_i)/\sqrt{|\mathcal{K}_i|}, (x - x_i)^2/(\sqrt{|\mathcal{K}_i|})^2, \\ &\quad (x - x_i)(y - y_i)/(\sqrt{|\mathcal{K}_i|})^2, (y - y_i)^2/(\sqrt{|\mathcal{K}_i|})^2, \cdots, (y - y_i)^q/(\sqrt{|\mathcal{K}_i|})^q \Big\} \ , \end{aligned} \tag{2.3}$$

where $\mathbf{x}_i \equiv (x_i, y_i)$ is the centroid of $\mathcal{K}_i$ and $r = (q + 1)(q + 2)/2 + 1$. $|\mathcal{K}_i|$ is the area of cell $\mathcal{K}_i$. The finite element space on cell $\mathcal{K}_i$ is the span of these basis functions.

In each cell $\mathcal{K}_i$, the approximate solution $u_{h,k}$ of the k^{th} equation of (2.1) is expressed as

$$u_{h,k} = \sum_{m=0}^{r} c_m(t)\phi_m(\mathbf{x}) \ . \tag{2.4}$$

Let's assume that the immediate neighbors (sharing same edges or vertices) of $\mathcal{K}_i$ are collected as the set

$$\mathcal{T}_{C,i} = \{\mathcal{K}_J : J = 1, 2, .., M.\} \ . \tag{2.5}$$

We remark that $\mathcal{T}_{C,i}$ also contains cell $\mathcal{K}_i$. The semi-discrete DG formulation of the k^{th} equation of (2.1) is to find an approximate solution u_h of the form (2.4) (neglecting its subscript k for convenience) such that

$$\frac{d}{dt}\int_{\mathcal{K}_i} u_h v_h d\mathbf{x} + \int_{\partial\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_i v_h d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} = 0 \ , \tag{2.6}$$

for any $v_h \in \text{span}\{\mathcal{B}_i\}$, where $\mathbf{n}_i$ is the outer unit normal vector of $\mathcal{K}_i$.

Since the approximate solution $\mathbf{u}_h$ is discontinuous across cell edges, the interfacial fluxes are not uniquely determined. The flux function $\mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_i$ appearing in equation (2.6) can be replaced by the Lax-Friedrich flux function (see e.g. [20]) defined as

$$h_k(\mathbf{x}, t) = h_k(\mathbf{u}_h^{in}, \mathbf{u}_h^{out}) = \frac{1}{2}(\mathbf{F}_k(\mathbf{u}_h^{in}) \cdot \mathbf{n}_i + \mathbf{F}_k(\mathbf{u}_h^{out}) \cdot \mathbf{n}_i) + \frac{\alpha}{2}(u_h^{in} - u_h^{out}) \quad , \quad k = 1, ..., m \ ,$$

where α is the largest characteristic speed,

$$\begin{aligned} \mathbf{u}_h^{in}(\mathbf{x}, t) &= \lim_{\mathbf{y}\to\mathbf{x}, \mathbf{y}\in\mathcal{K}_i^{\text{int}}} \mathbf{u}_h(\mathbf{y}, t) \ , \\ \mathbf{u}_h^{out}(\mathbf{x}, t) &= \lim_{\mathbf{y}\to\mathbf{x}, \mathbf{y}\notin\bar{\mathcal{K}}_i} \mathbf{u}_h(\mathbf{y}, t) \ . \end{aligned}$$

Equation (2.6) then becomes

$$\frac{d}{dt}\int_{\mathcal{K}_i} u_h v_h d\mathbf{x} + \int_{\partial\mathcal{K}_i} h_k v_h d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h)\cdot\nabla v_h d\mathbf{x} = 0 \ . \tag{2.7}$$

The resulting systems of ordinary differential equations can be solved by a TVD Runge-Kutta method [21] which builds on convex combinations of several forward Euler schemes of (2.7). Our additional conservation constraint is performed within each of the component forward Euler scheme. A forward Euler scheme of (2.7) can be written as

$$\int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} = \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \Delta t_n \int_{\partial\mathcal{K}_i} h_k^n v_h d\Gamma + \Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^n(\mathbf{u}_h)\cdot\nabla v_h d\mathbf{x} \ , \tag{2.8}$$

where the superscript n denotes the time level t_n, $\Delta t_n = t_{n+1} - t_n$. In particular, letting $v_h \equiv 1$, we obtain the cell average of u_h^{n+1} over cell $\mathcal{K}_i$, denoted by $\overline{u_i^{n+1}}$, just as with a finite volume scheme.

Now suppose the cell averages $\{\overline{u_i^{n+1}}\}$ have been computed on all cells. We do not compute the rest of the moments of u_h^{n+1} on cell $\mathcal{K}_i$ by using equation (2.8). Instead, we let u_h^{n+1} on cell $\mathcal{K}_i$ minimize an energy functional (variational to (2.8)) subject to that it conserves additional given cell averages not only in cell $\mathcal{K}_i$ but also in some of its neighbors. Rewrite (2.8) in cell $\mathcal{K}_i$ as

$$\int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} = \mathcal{L}(v_h) \ , \tag{2.9}$$

where $\mathcal{L}(v_h)$ represents the right-hand-side of (2.8), which is a linear bounded functional defined on the finite element space on $\mathcal{K}_i$. The variational form of (2.9) is to find u_h^{n+1} in the finite element space on $\mathcal{K}_i$ such that it minimizes the energy functional

$$E(v_h) = \frac{1}{2}\int_{\mathcal{K}_i} (v_h)^2 d\mathbf{x} - \mathcal{L}(v_h) \ . \tag{2.10}$$

In [], the conservation constrained RKDG formulation on cell $\mathcal{K}_i$ can be described as replacing each component forward Euler scheme by finding u_h^{n+1} in the finite element space on $\mathcal{K}_i$, such that

$$\begin{array}{rl} E(u_h^{n+1}) = & \text{Minimum of } \{E(v_h) : v_h \in \text{span}\{\mathcal{B}_i\}\}, \\ & \text{subject to } \ \frac{1}{|\mathcal{K}_J|}\int_{\mathcal{K}_J} v_h d\mathbf{x} = \overline{u_J^{n+1}} \ , \quad J = 1, ..., M \ . \end{array} \tag{2.11}$$

Here the set $\{\mathcal{K}_J\}$ consists of cell $\mathcal{K}_i$ and its immediate neighbors.

This constrained minimization problem can be solved by the method of Lagrangian multiplier as follows

$$\begin{array}{l} \int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} - \mathcal{L}(v_h) = \sum_{J=1}^{M} \frac{\lambda_J}{|\mathcal{K}_J|}\int_{\mathcal{K}_J} v_h d\mathbf{x}, \ \forall\ v_h \in \text{span}\{\mathcal{B}_i\} \\ \frac{1}{|\mathcal{K}_J|}\int_{\mathcal{K}_J} u_h^{n+1} d\mathbf{x} = \overline{u_J^{n+1}} \ , \quad J = 1, ..., M \ , \end{array} \tag{2.12}$$

where $\{\lambda_J\}$ are Lagrangian multipliers. Coefficients $\{c_m\}$ of u_h^{n+1} (see equation (2.4)) are determined by the above linear system. Note that the left-hand-side of the first equation

of (2.12) is in the same form as equation (2.9) or (2.8), and $M = 4$ for the 2D triangular meshes since the set $\{\mathcal{K}_J\}$ contains the cell $\mathcal{K}_i$ and its three adjacent neighbors (sharing common edges with $\mathcal{K}_i$). ($M = 3$ in 1D, the set $\{\mathcal{K}_J\}$ contains the cell $\mathcal{K}_i$ and its left and right neighbors.)

Even though this technique increases the CFL number, the use of Lagrangian multipliers also increases the dimensions of the linear system. In order to overcome this problem, we introduce a new minimization problem without any constraint.

Find $\tilde{u}_h^{n+1} \in \text{span}\{\mathcal{B}_i\}$ such that

$$E_2(\tilde{u}_h^{n+1}) = \text{Min}\left\{E_2(v_h) : v_h \in \text{span}\{\mathcal{B}_i\}\right\}, \tag{2.13}$$

where

$$E_2(v_h) = (1/|\mathcal{K}_i|)\, E(v_h) + \frac{\mu}{2} \sum_{j \in N(i)} \left(\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} v_h d\mathbf{x} - \overline{u_j^{n+1}} \right)^2, \tag{2.14}$$

$N(i)$ consists of indices of cell $\mathcal{K}_i$ and its immediate neighbors, and $\mu \geq 0$ is a constant. Note that when $\mu = 0$ the formulation returns to the standard DG.

Remark 1. It's easy to verify that energy function E_2 (2.14) is invariant (subject to a constant factor) under the scaling of the coordinates.

The variational formulation of problem (2.13) is to find $\tilde{u}_h^{n+1} \in \text{span}\{\mathcal{B}_i\}$ such that

$$\begin{array}{l} (1/|\mathcal{K}_i|) \left(\int_{\mathcal{K}_i} \tilde{u}_h^{n+1} v_h d\mathbf{x} + \int_{\partial \mathcal{K}_i} h_k v_h d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \right) + \\ \mu \sum_{j \in N(i)} \left(\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} v_h d\mathbf{x} \right) \left(\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} \tilde{u}_h^{n+1} d\mathbf{x} - \overline{u_j^{n+1}} \right) = 0, \end{array} \tag{2.15}$$

for any $v_h \in \text{span}\{\mathcal{B}_i\}$. In order to keep the cell averages of $\overline{u_j^{n+1}}$, we define $u_h^{n+1} \in \text{span}\{\mathcal{B}_i\}$ as

$$u_h^{n+1} = \tilde{u}_h^{n+1} + \overline{u_i^{n+1}} - \frac{1}{|\mathcal{K}_i|} \int_{\mathcal{K}_i} \tilde{u}_h^{n+1} d\mathbf{x} \tag{2.16}$$

Remark 2. Note that the linear system (2.15) consists of the same number of equations as in a RKDG step (2.8). Therefore the complexity of the new method is close to that of the standard RKDG (without using orthogonal basis functions).

The linear system (2.15) has a unique solution. In fact, consider the associate homogeneous system with $h_k, \mathbf{F}_k = 0$ and $\overline{u_j^{n+1}} = 0$ for all $j \in N(i)$,

$$(1/|\mathcal{K}_i|) \left(\int_{\mathcal{K}_i} \tilde{u}_h^{n+1} v_h d\mathbf{x} \right) + \mu \sum_{j \in N(i)} \left(\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} v_h d\mathbf{x} \right) \left(\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} \tilde{u}_h^{n+1} d\mathbf{x} \right) = 0. \tag{2.17}$$

Let $v_h = \tilde{u}_h^{n+1}$. We conclude that $\int_{\mathcal{K}_i} |\tilde{u}_h^{n+1}|^2 d\mathbf{x} = 0$, which implies $\tilde{u}_h^{n+1} \equiv 0$.

Remark 3. A compromised formulation with only one Lagrangian multiplier can be written as follows. Find $\tilde{u}_h^{n+1} \in \text{span}\{\mathcal{B}_i\}$ such that

$$\begin{array}{rl} E_2(\tilde{u}_h^{n+1}) = & \text{Min}\{E_2(v_h) : v_h \in \text{span}\{\mathcal{B}_i\}\} \\ & \text{subject to } \frac{1}{|\mathcal{K}_i|} \int_{\mathcal{K}_i} v_h d\mathbf{x} = \overline{u_i^{n+1}} \ . \end{array} \tag{2.18}$$

This method has similar complexity and CFL numbers to that of the formulation (2.13), (2.14) and (2.16).

2.2 Implementation

To summarize, assume we employ a s-stage TVD Runge-Kutta method to solve equation (2.7), which can be written in the form:

$$\begin{array}{rcl} \int_{\mathcal{K}_i} u_h^{(j)} v_h d\mathbf{x} & = & \sum_{l=0}^{j-1} \alpha_{jl} \left(\int_{\mathcal{K}_i} u^{(l)} v_h d\mathbf{x} + \Delta t_n \beta_{jl} L(u_h^{(l)}, v_h) \right) \ , \quad j = 1, ..., s \\ & \equiv & \sum_{l=0}^{j-1} \alpha_{jl} \int_{\mathcal{K}_i} u_h^{(j,l+1)} v_h d\mathbf{x} \ , \end{array} \tag{2.19}$$

with

$$u_h^{(0)} = u_h^n, \qquad u_h^{(s)} = u_h^{n+1} \ . \tag{2.20}$$

Here α_{jl} and β_{jl} are coefficients of the Runge-Kutta method at the j^{th} stage, and

$$L(u_h, v_h) = -\int_{\partial \mathcal{K}_i} h_k v_h d\Gamma + \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \ .$$

In particular, $u_h^{(j,l+1)}$ is determined by

$$\int_{\mathcal{K}_i} u_h^{(j,l+1)} v_h d\mathbf{x} = \int_{\mathcal{K}_i} u^{(l)} v_h d\mathbf{x} + \Delta t_n \beta_{jl} L(u_h^{(l)}, v_h) \ , \ \forall \ v_h \in \text{span}\{\mathcal{B}_i\} \ .$$

This is a forward Euler scheme as in (2.8) with the time step size $\Delta t_n \beta_{jl}$, and will be replaced similarly by the modification as in (2.15) and (2.16). This technique can also be applied to the classical 4th order Rung-Kutta method with the DG component (2.8). The 4 stages are written as follows

$$\begin{array}{rcl} \int_{\mathcal{K}_i} u_h^{n+1/2-} v_h d\mathbf{x} & = & \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \frac{1}{2}\Delta t_n \int_{\partial \mathcal{K}_i} h_k^n v_h d\Gamma + \frac{1}{2}\Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^n(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \\ & \equiv & \mathcal{L}_1(v_h) \ , \\ \int_{\mathcal{K}_i} u_h^{n+1/2+} v_h d\mathbf{x} & = & \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \frac{1}{2}\Delta t_n \int_{\partial \mathcal{K}_i} h_k^{n+1/2-} v_h d\Gamma + \frac{1}{2}\Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^{n+1/2-}(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \\ & \equiv & \mathcal{L}_2(v_h) \ , \\ \int_{\mathcal{K}_i} u_h^{n+1-} v_h d\mathbf{x} & = & \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \Delta t_n \int_{\partial \mathcal{K}_i} h_k^{n+1/2+} v_h d\Gamma + \Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^{n+1/2+}(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \\ & \equiv & \mathcal{L}_3(v_h) \ , \\ \int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} & = & \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \Delta t_n \int_{\partial \mathcal{K}_i} h_k^* v_h d\Gamma + \Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^*(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} \\ & \equiv & \mathcal{L}_4(v_h) \ , \end{array} \tag{2.21}$$

where $h_k^{n+1/2-}$ denotes the numerical flux evaluated with $u_h^{n+1/2-}$ from the previous stage and

$$h_k^* = \frac{1}{6}\{h_k^n + 2h_k^{n+1/2-} + 2h_k^{n+1/2+} + h_k^{n+1-}\},$$

similarly for $\mathbf{F}_k^{n+1/2-}$, $\mathbf{F}_k^{n+1/2+}$ and other fluxes. The modification (2.13) and (2.16) that has been applied to $\mathcal{L}$ in equation (2.9) can be applied t0 $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_4$ to modify the values of $u_h^{n+1/2-}$, $u_h^{n+1/2+}$, u_h^{n+1-} and u_h^{n+1} respectively.

2.2.1 Choices of cells being used as constraints

Here we give several possible choices of the set of cells that can be used as constraints. For the 1D third- and fourth-order accurate constrained RKDG schemes, the set of cells $\mathcal{T}_{C,i}$ defined in Eq. (2.5) for solving for the solution on $\mathcal{K}_i$ cell is $\mathcal{T}_{C,i} = \{\mathcal{K}_{i-1}, \mathcal{K}_i, \mathcal{K}_{i+1}\}$. The resulting 1D third- and fourth-order accurate constrained RKDG schemes are denoted as "1D Constrained RKDG3-2Cell" and "1D Constrained RKDG4-2Cell" in the following sections, respectively.

2.3 Analytical estimates of the CFL numbers

Table 1: CFL numbers with $\alpha = 0$ (no constraint).

Temporal order	p-w linear	p-w quadratic	p-w cubic
2nd	0.33	0.11	0.050
3rd	0.40	0.20	0.13
4th	-	-	0.14

Table 2: CFL numbers with $\alpha = 1$.

Temporal order	p-w linear	p-w quadratic	p-w cubic
2nd	0.97	0.86	0.18
3rd	1.2	1.6	0.49
4th	-	-	0.56

Table 3: CFL numbers with $\alpha = 100$.

Temporal order	p-w linear	p-w quadratic	p-w cubic
2nd	1.0	0.88	0.18
3rd	1.1	1.6	0.49
4th	-	-	0.56

2.4 Limiting by hierarchical reconstruction

To prevent non-physical oscillations in the vicinity of discontinuities, we apply HR [25] at each Runge-Kutta stage to the DG solution. Since shock waves or contact discontinuities are all local phenomena, we apply the HR limiting procedure to a small region covering discontinuities. Specifically, we employ a local limiting procedure by using a detector [4]

to identify cells which may contain oscillatory solutions. HR is then applied to solutions supported on these cells. We give a brief description of the HR limiting procedure here. More details can be found in [25].

HR decomposes the job of limiting a high-order polynomial supported on a cell (which may contain spurious oscillations) into a series of smaller jobs, each of which only involves the non-oscillatory reconstruction of a linear polynomial, which can be easily achieved through classical processes such as the MUSCL reconstruction [9, 10, 11] used in [13], or a WENO-type combination used in [25]. Since the reconstruction of a linear polynomial can only use information from adjacent cells, HR can be formulated in multi-dimensions on a compact stencil. Using the basis function set (2.3), the approximate solution $u_h(\mathbf{x}-\mathbf{x}_i)$ on cell $\mathcal{K}_i$ is in the Taylor expansion around cell centroid $\mathbf{x}_i$

$$u_h(\mathbf{x}-\mathbf{x}_i) = \sum_{m=0}^{q} \sum_{|\mathbf{m}|} \frac{1}{\mathbf{m}!} u_h^{(\mathbf{m})}(0)(\mathbf{x}-\mathbf{x}_i)^{\mathbf{m}} \ . \tag{2.22}$$

$u_h(\mathbf{x}-\mathbf{x}_i)$ may contain spurious oscillations. The hierarchical reconstruction procedure is to recompute the coefficients of polynomial $u_h(\mathbf{x}-\mathbf{x}_i)$ by using polynomials in cells adjacent to $\mathcal{K}_i$ (or partial neighboring cells [25]). These adjacent cells (or partial cells) are collected as the set $\{\mathcal{K}_j\}$ (which also contains cell $\mathcal{K}_i$) and the polynomials (of degree q) supported on them are denoted as $\{u_{h,j}(\mathbf{x}-\mathbf{x}_j)\}$ respectively. HR recomputes a set of new coefficients

$$\frac{1}{\mathbf{m}!}\widetilde{u}_h^{(\mathbf{m})}(\mathbf{0}), \qquad |\mathbf{m}| = q, q-1, \ldots, 0$$

to replace the original coefficients $\frac{1}{\mathbf{m}!}u_h^{(\mathbf{m})}(\mathbf{0})$ of $u_h(\mathbf{x}-\mathbf{x}_i)$ iteratively from the highest to the lowest degree terms without losing the order of accuracy if the piecewise polynomial solution is locally smooth, and eliminates spurious oscillations of $u_h(\mathbf{x}-\mathbf{x}_i)$ otherwise.

To obtain $\widetilde{u}_h^{(\mathbf{m})}(\mathbf{0})$, we first compute *candidates* of $u_h^{(\mathbf{m})}(\mathbf{0})$, and then let the new value for $u_h^{(\mathbf{m})}(\mathbf{0})$ be

$$\widetilde{u}_h^{(\mathbf{m})}(\mathbf{0}) = F\big(\text{ candidates of } u_h^{(\mathbf{m})}(\mathbf{0})\ \big),$$

where F is a convex limiter of its arguments (e.g., the center biased minmod function used in [14], or the WENO-type combination in [25]), $F(a_1, a_2, \cdots, a_l) = \sum_{i=1}^{l} \theta_i a_i$, for some $\theta_i \geq 0$ and $\sum_{i=1}^{l} \theta_i = 1$.

In order to find these candidates of $u_h^{(\mathbf{m})}(\mathbf{0})$, $|\mathbf{m}| = m$, we take a $(m-1)^{th}$ order partial derivative of $u_h(\mathbf{x}-\mathbf{x}_i)$ (and also polynomials in adjacent cells), and express

$$\partial^{m-1} u_h(\mathbf{x}-\mathbf{x}_i) = L_h(\mathbf{x}-\mathbf{x}_i) + R_h(\mathbf{x}-\mathbf{x}_i),$$

Table 4: CFL numbers with $\alpha = 1000$.

Temporal order	p-w linear	p-w quadratic	p-w cubic
2nd	1.0	0.88	0.18
3rd	1.1	1.6	0.49
4th	-	-	0.56

where L_h is the linear part (containing the zeroth and first degree terms) and R_h is the remainder. Clearly, every coefficient in the first degree terms of L_h is in the set $\{u_h^{(\mathbf{m})}(\mathbf{0}) : |\mathbf{m}| = m\}$. And for every $\mathbf{m}$ subject to $|\mathbf{m}| = m$, one can always take some $(m-1)^{th}$ order partial derivatives of $u_h(\mathbf{x}-\mathbf{x}_i)$ so that $u_h^{(\mathbf{m})}(\mathbf{0})$ is a coefficient in a first degree term of L_h. Thus, a "candidate" for a coefficient in a first degree term of L_h is also the candidate for the corresponding $u_h^{(\mathbf{m})}(\mathbf{0})$.

In order to find a set of candidates for all coefficients in the first degree terms of $L_h(\mathbf{x}-\mathbf{x}_i)$, we need to know the new approximate cell averages of $L_h(\mathbf{x}-\mathbf{x}_i)$ on $d+1$ distinct mesh cells adjacent to cell $\mathcal{K}_i$, which is a key step. Assume $\mathcal{K}_{j_0}, \mathcal{K}_{j_1}, \cdots, \mathcal{K}_{j_d} \in \{\mathcal{K}_j\}$ are these cells and $\overline{L}_{j_0}, \overline{L}_{j_1}, \cdots, \overline{L}_{j_d}$ are the corresponding new approximate cell averages. For example, in order to obtain $\overline{L}_{j_1}$, we first compute

$$A_{j_1} = \frac{1}{|\mathcal{K}_{j_1}|} \int_{\mathcal{K}_{j_1}} \partial^{m-1} u_{h,j_1}(\mathbf{x}-\mathbf{x}_{j_1}) d\mathbf{x},$$

then

$$D_{j_1} = \frac{1}{|\mathcal{K}_{j_1}|} \int_{\mathcal{K}_{j_1}} \widetilde{R_h}(\mathbf{x}-\mathbf{x}_i) d\mathbf{x},$$

where $\widetilde{R_h}(\mathbf{x}-\mathbf{x}_i)$ is the $R_h(\mathbf{x}-\mathbf{x}_i)$ with its coefficients replaced by previously computed new values. Finally we can set $\overline{L}_{j_1} = A_{j_1} - D_{j_1}$.

More details of the HR implementation for one-dimensional test problems in this paper can be found in [26].

2.4.1 Hierarchical reconstruction with local iteration

On the 2D triangular grids, for limiting the 4^{th} order solutions, we observed the returning of small overshoots/undershoots when using the partial neighboring cell technique developed for the 3^{rd} order case. Here we introduce a local iteration technique to further reduce the small overshoots/undershoots. Assume we need to reconstruct the approximate solution $u_h(\mathbf{x}-\mathbf{x}_i)$ supported on cell $\mathcal{K}_i$ by HR, which uses information from neighboring cells (more precisely, partial neighboring cells [25]). When all new values of the coefficients of $u_h(\mathbf{x}-\mathbf{x}_i)$ have been computed, we will update $u_h(\mathbf{x}-\mathbf{x}_i)$ while keeping the solution on its neighboring cells unchanged, and apply HR to reconstruct $u_h(\mathbf{x}-\mathbf{x}_i)$ again. In other words, we apply HR twice to update $u_h(\mathbf{x}-\mathbf{x}_i)$ with the solutions on its neighboring cells being fixed. The new iteration reduces the possible remaining overshoots/undershoots without spreading out the diffusion.

3 Numerical Examples

3.1 Accuracy test using 1D linear advection equation

We first test the capability of the constrained RKDG method to achieve the desired order of accuracy with a large CFL number, using the 1D linear advection equation

$$u_t + u_x = 0\ , \qquad (x,t) \in (-1,1) \times (0,T) \tag{3.1}$$

with periodic boundary conditions and the initial condition

$$u(x, t = 0) = \frac{1}{2} + \sin(\pi x) \ , \qquad -1 \leq x \leq 1 \ . \tag{3.2}$$

The uniform mesh is used to solve this test problem. The solution is computed up to $T = 2.0$. The cell size, denoted by $\triangle x$, is listed in tables shown in this section. Table 5 shows that the third-order accurate constrained RKDG method is stable with that the CFL number is equal to 1.6; while table 6 shows that the fourth-order accurate constrained RKDG method is stable with the CFL number equals to 0.6. We also tested the maximum values of the CFL number that the third-order and fourth-order accurate RKDG methods can use numerically. We observed that these maximum values are around 0.2 and 0.1 for the third-order and fourth-order accurate RKDG methods, respectively. Tables 5 and 6 show these test results as well.

We also studied the how the conservation penalty weight γ affects the CFL numbers that can be used numerically. Tables 7 and 8 list the accuracy test results with $\gamma = 1, 10, 1000$ for the third-order and fourth-order accurate constrained RKDG schemes, respectively. It is clear that the allowed maximum CFL numbers for these two schemes are not affected by the choice of γ values. Additionally, the L_1 and L_∞ errors do not seem to be affected by the γ values as well.

Table 5: Accuracy test results of solving 1D linear advection equation (3.1) by using the third-order accurate schemes. L^1 and L^∞ errors. 2 cells are used for conservation penalty. $T = 2.0$.

$\triangle x$	1D Constrained RKDG3-2Cell, $\mu = 1$, CFL = 1.6				1D RKDG3, CFL = 0.2			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{400}$	4.74E-7	-	7.45E-7	-	1.48E-9	-	3.84E-9	-
$\frac{1}{800}$	5.92E-8	3.00	9.31E-8	3.00	1.85E-10	3.64	4.79E-10	3.00
$\frac{1}{1600}$	7.40E-9	3.00	1.16E-8	3.00	2.31E-11	3.00	6.00E-11	3.00
$\frac{1}{3200}$	9.25E-10	3.00	1.45E-9	3.00	3.29E-12	2.81	9.21E-12	2.70
$\frac{1}{6400}$	1.16E-10	3.00	1.82E-10	2.99	3.42E-13	3.27	1.08E-12	3.09
$\frac{1}{12800}$	1.45E-11	3.00	2.28E-11	3.00	-	-	-	-

3.2 Accuracy test using using 1D Burgers' equation with a smooth solution

Here we test the maximum CFL number that the constrained RKDG method can achieve by using the 1D scalar Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0 \ , \qquad (x, t) \in (-1, 1) \times (0, T) \ , \tag{3.3}$$

with a periodic boundary condition and the initial condition

$$u(x, t = 0) = \frac{1}{2} + \sin(\pi x) \ , \qquad -1 \leq x \leq 1 \ . \tag{3.4}$$

Table 6: Accuracy test results of solving 1D linear advection equation (3.1) by using the fourth-order accurate schemes. L^1 and L^∞ errors. 2 cells are used for conservation penalty. $T = 2.0$.

$\triangle x$	1D Constrained RKDG4-2Cell, $\mu = 1$, CFL = 0.6				1D RKDG4, CFL = 0.1			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{50}$	1.34E-7	-	2.65E-7	-	2.06E-9	-	5.33E-9	-
$\frac{1}{100}$	8.30E-9	4.01	1.63E-8	4.02	1.28E-10	4.01	3.33E-10	4.00
$\frac{1}{200}$	5.23E-10	3.99	1.05E-9	3.96	8.02E-12	4.00	2.08E-11	4.00
$\frac{1}{400}$	3.25E-11	4.01	6.42E-11	4.03	7.52E-13	3.41	1.66E-12	3.65
$\frac{1}{800}$	2.13E-12	3.93	4.23E-12	3.92	-	-	-	-

Table 7: Solving 1D linear advection equation (3.1) by using the 1D third-order accurate Constrained RKDG (1D RKDG3-2Cell) method with different μ values. L^1 and L^∞ errors. 2 cells are used for conservation penalty. CFL = 1.6. $T = 2.0$.

$\triangle x$	1D Constrained RKDG3-2Cell					
	$\mu = 1$		$\mu = 10$		$\mu = 1000$	
	L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
$\frac{1}{1600}$	7.40E-9	1.16E-8	7.67E-9	1.20E-8	7.69E-9	1.21E-8
$\frac{1}{3200}$	9.25E-10	1.45E-9	9.58E-10	1.51E-9	9.62E-10	1.51E-9
$\frac{1}{6400}$	1.16E-10	1.82E-10	1.20E-10	1.88E-10	1.20E-10	1.89E-10
$\frac{1}{12800}$	1.45E-11	2.28E-11	1.50E-11	2.36E-11	1.50E-11	2.38E-11

Table 8: Solving 1D linear advection equation (3.1) by using the 1D fourth-order accurate Constrained RKDG (1D RKDG4-2Cell) method with different μ values. L^1 and L^∞ errors. 2 cells are used for conservation penalty. CFL = 1.6. $T = 2.0$.

$\triangle x$	1D Constrained RKDG4-2Cell					
	$\mu = 1$		$\mu = 10$		$\mu = 1000$	
	L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
$\frac{1}{100}$	8.30E-9	1.63E-8	8.31E-9	1.63E-8	8.31E-9	1.63E-8
$\frac{1}{200}$	5.23E-10	1.05E-9	5.24E-10	1.05E-9	5.24E-10	1.05E-9
$\frac{1}{400}$	3.25E-11	6.42E-11	3.25E-11	6.42E-11	3.25E-11	6.60E-11
$\frac{1}{800}$	2.13E-12	4.23E-12	2.13E-12	4.24E-12	2.32E-12	9.66E-12

The uniform mesh is used to solve this test problem. The solution is computed up to $T = 0.5/\pi$, when it is still smooth. $\triangle x$ listed in tables shown in this section is the cell size.

Table 9 shows that the third-order accurate constrained RKDG method can use a CFL number being equal to 1.6; while table 10 shows that the fourth-order accurate constrained RKDG method is stable with a CFL number being equal to 0.6 for this 1D nonlinear problem test case.

Similar to the 1D linear advection equation test case demonstrated in Sec. 3.1, the numerical study of the choice of γ values for this 1D nonlinear test case shows that the allowed maximum CFL numbers and L_1 and L_∞ errors of numerical solutions are not sensitive to γ values as well. These results are summarized in tables 11 and 12, respectively.

Table 9: Accuracy test results of solving 1D Burgers' equation (3.3) by using the third-order accurate schemes. L^1 and L^∞ errors. 2 cells are used for conservation penalty. $T = 0.5/\pi$.

$\triangle x$	1D Constrained RKDG3-2Cell, $\mu = 1$, CFL $= 1.6$				1D RKDG3, CFL $= 0.2$			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{400}$	9.32E-8	-	6.17E-7	-	3.51E-9	-	5.78E-8	-
$\frac{1}{800}$	1.18E-8	2.98	8.11E-8	2.93	4.36E-10	3.01	7.30E-9	2.99
$\frac{1}{1600}$	1.47E-9	3.00	1.00E-8	3.02	5.44E-11	3.00	9.20E-10	2.99
$\frac{1}{3200}$	1.85E-10	2.99	1.29E-9	2.95	6.84E-12	2.99	1.17E-10	2.98
$\frac{1}{6400}$	2.30E-11	3.01	1.60E-10	3.01	1.11E-12	2.62	1.81E-11	2.69
$\frac{1}{12800}$	3.00E-12	2.94	2.01E-11	2.99	-	-	-	-

Table 10: Accuracy test results of solving 1D Burgers' equation (3.3) by using the fourth-order accurate schemes. L^1 and L^∞ errors. 2 cells are used for conservation penalty. $T = 0.5/\pi$.

$\triangle x$	1D Constrained RKDG4-2Cell, $\mu = 1$, CFL $= 0.6$				1D RKDG4, CFL $= 0.1$			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{50}$	4.56E-7	-	1.18E-5	-	2.27E-8	-	3.64E-7	-
$\frac{1}{100}$	2.99E-8	3.93	7.67E-7	3.94	1.42E-9	4.00	2.27E-8	4.00
$\frac{1}{200}$	1.93E-9	3.95	4.90E-8	3.97	8.89E-11	4.00	1.44E-9	3.98
$\frac{1}{400}$	1.21E-10	4.00	3.08E-9	3.99	5.81E-12	3.94	9.27E-11	3.96
$\frac{1}{800}$	7.87E-12	3.94	1.95E-10	3.98	-	-	-	-

3.3 Accuracy test using 2D linear advection equation

To assess maximum CFL number that the constrained RKDG methods can use on 2D triangular meshes, we use the following definition of the CFL number, which is the maximum of

$$\frac{\triangle t(|\mathbf{u}| + c)}{\mathcal{D}}\ , \tag{3.5}$$

Table 11: Solving 1D Burgers' equation (3.3) by using the third-order accurate Constrained RKDG (1D RKDG3-2Cell) method with different μ values. L^1 and L^∞ errors. 2 cells are used for conservation penalty. CFL = 1.6. $T = 0.5/\pi$.

$\triangle x$	1D Constrained RKDG3-2Cell					
	$\mu = 1$		$\mu = 10$		$\mu = 1000$	
	L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
$\frac{1}{1600}$	1.47E-9	1.00E-8	1.49E-9	1.05E-9	1.49E-9	1.06E-8
$\frac{1}{3200}$	1.85E-10	1.29E-9	1.86E-10	1.33E-10	1.87E-10	1.34E-9
$\frac{1}{6400}$	2.30E-11	1.60E-10	2.34E-11	1.70E-11	2.34E-11	1.71E-10
$\frac{1}{12800}$	3.00E-12	2.01E-11	3.04E-12	2.13E-12	3.05E-12	2.15E-11

Table 12: Solving 1D Burgers' equation (3.3) by using the fourth-order accurate Constrained RKDG (1D RKDG4-2Cell) method with different μ values. L^1 and L^∞ errors. 2 cells are used for conservation penalty. CFL = 0.6. $T = 0.5/\pi$.

$\triangle x$	1D Constrained RKDG4-2Cell					
	$\mu = 1$		$\mu = 10$		$\mu = 1000$	
	L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
$\frac{1}{200}$	1.93E-9	4.90E-8	1.94E-9	4.91E-08	1.94E-9	4.91E-08
$\frac{1}{400}$	1.21E-10	3.08E-9	1.21E-10	3.08E-09	1.21E-10	3.08E-09
$\frac{1}{800}$	7.87E-12	1.95E-10	7.87E-12	1.95E-10	8.03E-12	1.97E-10

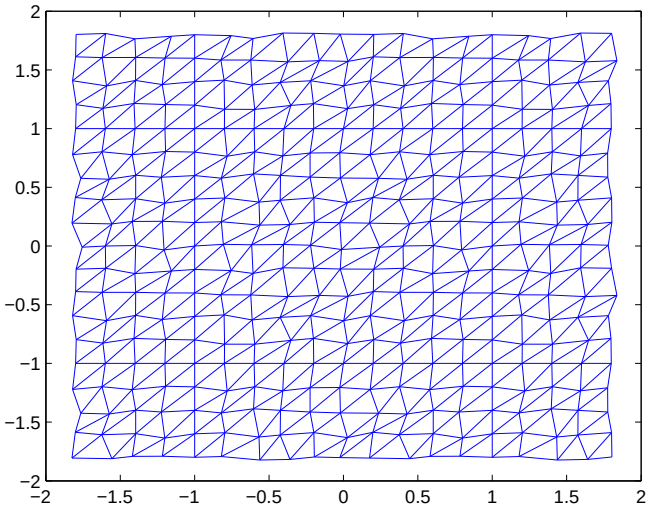


Figure 1: Representative mesh for 2D accuracy tests.

where $\mathcal{D}$ is the diameter of the inscribed circle of a triangle, c is the speed of sound for compressible flow and $|\mathbf{u}|$, defined by ($|\mathbf{u}| = \sqrt{u^2 + v^2}$), is the speed of flow. Both c and $|\mathbf{u}|$ are evaluated by the local cell average value.

We start with the initial-boundary-value problem of the 2D linear advection equation

$$\begin{array}{ll} u_t + u_x + u_y = 0, & (x, y, t) \in \Omega \times (0, T) \\ u(x, y, t = 0) = \frac{1}{4} + \frac{1}{2}\sin(\pi(x + y)), & (x, y) \in \Omega \ . \end{array} \tag{3.6}$$

to assess the limit of the allowed CFL numbers for the third-order and fourth-order accurate constrained RKDG methods for solving 2D problems. The domain Ω is the square $[-1, 1] \times [-1, 1]$. The periodic boundary condition is used in both directions. For the convenience of implementing the periodic boundary condition, the triangular mesh is obtained by perturbing a uniform triangulation, see Fig. 1 for a typical mesh used for the accuracy test.

We compute the solution up to $T = 2.0$. The typical triangle edge length, denoted by h, is listed in tables shown in this section. The errors presented are for u. For this test case, $|\mathbf{u}|$ in Eq. (3.5) is equal to 1 and $c = 0$.

Table 14 shows that the fourth-order accurate constrained RKDG method using 13 cells (2D Constrained RKDG4-13Cell) as constraints is stable when CFL number is equal to 1.3; while the numerical test shows that the fourth-order accurate RKDG method is stable when CFL number is around 0.2.

We studied how penalty constant μ affects magnitudes of errors for this test case. Table 15 shows that the L^1 and L^∞ errors computed by the 2D Constrained RKDG4-13Cell method increase slightly when μ varies between 1 and 1000. Thus the scheme is not sensitive for the choice of μ value.

Moreover, we tested how choice of constraint cells affect the CFL number. When 7 cells are used as constraints, the maximum CFL number the 2D fourth-order accurate Constrained RKDG method can take is around 0.35. However, the differences between magnitudes of L^1

and L^∞ errors computed by the 2D fourth-order accurate Constrained RKDG methods with 13 constraint cells and 7 constraint cells are small. See Table 16 for this result.

Table 13: Accuracy test results of solving 2D linear advection equation (3.6). CFL = 0.80. L^1 and L^∞ errors. 7 Cells are used for conservation penalty.

h	2D Constrained RKDG3-7Cell, $\mu =$				RKDG3			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
1/20	xxx	-	xxx	-	xxx	-	xxx	-

Table 14: Accuracy test results of solving 2D linear advection equation (3.6) by the fourth-order accurate schemes. L^1 and L^∞ errors. 13 cells are used for conservation penalty. $T = 2.0$.

h	2D Constrained RKDG4-13Cell, $\mu = 1$, CFL=1.3				RKDG4, CFL = 0.2			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{80}$	3.28E-7	-	2.62E-7	-	5.42E-9	-	4.80E-9	-
$\frac{1}{160}$	1.93E-8	4.09	1.79E-8	3.87	3.35E-10	-	3.34E-10	-
$\frac{1}{320}$	1.29E-9	3.90	1.38E-9	3.70	2.13E-11	-	2.09E-11	-
$\frac{1}{640}$	8.02E-11	4.01	8.76E-11	3.98	2.73E-12	-	3.57E-12	-

Table 15: Accuracy test results of solving 2D linear advection equation (3.6) by the fourth-order accurate Constrained RKDG (2D RKDG4-13Cell) method with different μ values. L^1 and L^∞ errors. 13 cells are used for conservation penalty. CFL = 1.3. $T = 2.0$.

h	2D Constrained RKDG4-13Cell					
	$\mu = 1$		$\mu = 10$		$\mu = 1000$	
	L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
$\frac{1}{160}$	1.93E-8	1.79E-8	4.97E-8	2.85E-08	5.57E-8	3.08E-8
$\frac{1}{320}$	1.29E-9	1.38E-9	3.11E-9	2.03E-9	3.49E-9	2.31E-9
$\frac{1}{640}$	8.02E-11	8.76E-11	1.96E-10	1.34E-10	2.20E-10	1.46E-10

3.4 Accuracy test using 2D Burgers' equation with a smooth solution

To assess the limit of the permissible CFL number for the constrained RKDG methods for the 2D nonlinear scalar conservation laws, we solve the following initial-boundary-value problem of the 2D Burgers' equation

$$\begin{aligned} &u_t + \left(\tfrac{1}{2}u^2\right)_x + \left(\tfrac{1}{2}u^2\right)_y = 0, && (x,y) \in \Omega \times (0,T) \\ &u(x,y,t=0) = \tfrac{1}{4} + \tfrac{1}{2}\sin(\pi(x+y)), && (x,y) \in \Omega\ , \end{aligned} \tag{3.7}$$

where $\Omega = [-1, 1] \times [-1, 1]$. The periodic boundary condition is used in both directions. The solution is computed up to $T = 0.5/\pi$, when it is still smooth. The triangular meshes utilized for the 2D linear advection equation test are also used for this convergence test. We use the definition of the CFL number given by Eq. (**??**). For the 2D Burgers' equation test case, $|\mathbf{u}| = \sqrt{u^2 + u^2} = \sqrt{2}|u|$. $|u|$ is evaluated by the local cell average value. The errors presented in tables shown in this section are for u. h listed in these tables is the edge length of the triangle.

Table 17 shows the L^1 and L^∞ errors and numerical orders of accuracy for using the third-order accurate constrained RKDG method with 7 constraint cells (2D Constrained RKDG3-7Cell)) for solving Eq. (3.7). With $\mu = 1.0$, we are able to use a CFL number = 0.8 for computing the solution while achieving the desired order of accuracy. In this table, we also show that the maximum CFL number that the 2D third-order accurate RKDG method can use is about 0.22 by our numerical test. Table 18 shows that when 4 cells are used as constraints, the 2D third-order accurate constrained RKDG method (2D Constrained RKDG3-4Cell) is permitted to use a CFL number = 0.3 for the nonlinear equation test case. In addition, the L^1 and L^∞ errors of the numerical solutions to Eq. 3.7 computed by the 2D Constrained RKDG3-4Cell method is about $3 \sim 4$ times smaller than the ones computed by the 2D Constrained RKDG3-7Cell method.

In Table 19, we demonstrate the L^1 and L^∞ errors and numerical orders of accuracy for using the fourth-order accurate constrained RKDG method with 13 constraint cells (2D Constrained RKDG4-13Cell) for solving Eq. (3.7). With $\mu = 1.0$, the 2D Constrained RKDG4-13Cell method is capable of using a CFL number = 1.4 for achieving the fourth-order accuracy. In this table, we also show that the maximum CFL number that the 2D fourth-order accurate RKDG method can use is about 0.25 by our numerical test. Table 20 shows that when 7 cells are used as constraints, the 2D fourth-order accurate constrained RKDG method (2D Constrained RKDG4-7Cell) is allowed to use a CFL number = 0.35 for the nonlinear equation test case. The L^1 and L^∞ errors of the numerical solutions to Eq. 3.7 computed by the 2D Constrained RKDG4-7Cell method is about $2 \sim 4$ times smaller than the ones computed by the 2D Constrained RKDG4-13Cell method.

To summarize, we found that the 2D constrained RKDG3-7Cell method is stable when CFL ≤ 0.8; this improves CFL number about $3 - 4$ times compared with the one that the 2D third-order accurate RKDG scheme can take. The 2D constrained RKDG4-13Cell method improves the CFL number over the 2D fourth-order accurate RKDG method by 5

Table 16: Accuracy test results of solving 2D linear advection equation (3.6) by the fourth-order accurate constrained RKDG method (2D Constrained RKDG4) with different numbers of cells used for conservation penalty. L^1 and L^∞ errors. $\mu = 1$. $T = 2.0$.

h	2D Constrained RKDG4-13Cell, CFL = 1.3				2D Constrained RKDG4-7Cell, CFL = 0.35			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{160}$	1.93E-8	-	1.79E-8	-	1.92E-8	-	1.04E-8	-
$\frac{1}{320}$	1.29E-9	-	1.38E-9	-	1.20E-9	-	6.94E-10	-
$\frac{1}{640}$	8.02E-11	-	8.76E-11	-	7.62E-11	-	4.56E-11	-

Table 17: Accuracy test results of solving 2D Burgers' equation (3.7) by the third-order accurate schemes. L^1 and L^∞ errors. 7 cells are used for conservation penalty. $T = 0.5/\pi$.

h	2D Constrained RKDG3-7Cell, $\mu = 1$, CFL = 0.8				2D RKDG3, CFL = 0.22			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{80}$	3.53E-5	-	6.47E-5	-	3.06E-6	-	1.26E-5	-
$\frac{1}{160}$	4.40E-6	3.00	9.56E-6	2.76	3.85E-7	2.99	1.67E-6	2.92
$\frac{1}{320}$	5.49E-7	3.00	1.29E-6	2.89	4.85E-8	2.99	2.27E-7	2.88
$\frac{1}{640}$	6.93E-8	2.99	1.73E-7	2.90	6.08E-9	3.00	3.03E-8	2.91
$\frac{1}{1280}$	9.44E-9	2.88	2.99E-8	2.53	7.62E-10	3.00	4.04E-9	2.91
$\frac{1}{2560}$	1.08E-9	3.13	3.31E-9	3.18	9.58E-11	2.99	5.49E-10	2.88

Table 18: Accuracy test results of solving 2D Burgers' equation (3.7) by the third-order accurate constrained RKDG method (2D Constrained RKDG3) with different numbers of cells used for conservation penalty. L^1 and L^∞ errors. $\alpha = 1$. $T = 0.5/\pi$.

h	2D Constrained RKDG3-7Cell, CFL = 0.8				2D Constrained RKDG3-4Cell, CFL = 0.3			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{160}$	4.40E-6	-	9.56E-6	-	1.12E-6	-	3.46E-6	-
$\frac{1}{320}$	5.49E-7	3.00	1.29E-6	2.89	1.42E-7	2.98	4.89E-7	2.82
$\frac{1}{640}$	6.93E-8	2.99	1.73E-7	2.90	1.80E-8	2.98	6.50E-8	2.91
$\frac{1}{1280}$	9.44E-9	2.88	2.99E-8	2.53	2.27E-9	2.99	9.23E-9	2.82
$\frac{1}{2560}$	1.08E-9	3.13	3.31E-9	3.18	2.85E-10	2.99	1.35E-9	2.77

Table 19: Accuracy test results of solving 2D Burgers' equation (3.7) by the fourth-order accurate schemes. L^1 and L^∞ errors. 13 cells are used for conservation penalty. $T = 0.5/\pi$.

h	2D Constrained RKDG4-13Cell, $\mu = 1$, CFL = 1.4				2D RKDG4, CFL = 0.25			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{80}$	2.39E-6	-	8.22E-6	-	5.18E-8	-	1.88E-7	-
$\frac{1}{160}$	1.55E-7	3.95	6.38E-7	3.69	3.29E-9	3.98	1.25E-8	3.91
$\frac{1}{320}$	9.36E-9	4.05	5.19E-8	3.62	2.10E-10	3.97	9.51E-10	3.72
$\frac{1}{640}$	5.89E-10	3.99	3.29E-9	3.98	1.35E-11	3.96	6.61E-11	3.85
$\frac{1}{1280}$	3.73E-11	3.98	2.39E-10	3.78	-	-	-	-

~ 6 times.

3.5 Test cases using 1D Euler equations with discontinuous solutions

We now assess the resolution and the non-oscillatory property of numerical solutions computed by the constrained RKDG methods and limited by HR. We compute solutions of the 1D Euler equations

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0$$

with $\mathbf{u} = (\rho, \rho v, E)^T$, $\mathbf{f}(\mathbf{u}) = (\rho v, \rho v^2 + p, v(E+p))^T$, $p = (\gamma - 1)(E - \frac{1}{2}\rho v^2)$ and $\gamma = 1.4$.

3.5.1 1D Shu-Osher problem

The 1D Shu-Osher problem [22] is the Euler equations with an initial data

$$\begin{aligned} (\rho, v, p) &= (3.857143,\ 2.629369,\ 10.333333), \quad \text{for } x < -4, \\ (\rho, v, p) &= (1 + 0.2\sin(5x),\ 0,\ 1), \quad \text{for } x \geq -4. \end{aligned}$$

We computed the numerical solutions using 300 equal size cells. Density profiles of the solutions computed by the third-order and fourth-order accurate constrained RKDG and RKDG schemes respectively at the time $t = 1.8$ are shown in Fig. 2. We can clearly see that the fourth-order constrained RKDG solution and the fourth-order RKDG solution have almost identical resolution for this test problem. The resolution of the third-order accurate constrained RKDG solution is inferior to that of the third-order accurate RKDG solution for this Shu-Osher test case.

3.5.2 1D Woodward-Colella blast wave problem

The 1D Woodward-Colella blast wave problem [23] is the Euler equations with an initial data

$$\begin{aligned} (\rho, \rho v, E) &= (1,\ 0,\ 2500), \quad \text{for } 0 < x < 0.1, \\ (\rho, \rho v, E) &= (1,\ 0,\ 0.025), \quad \text{for } 0.1 < x < 0.9, \\ (\rho, \rho v, E) &= (1,\ 0,\ 250), \quad \text{for } 0.9 < x < 1. \end{aligned}$$

Table 20: Accuracy test results of solving 2D Burgers' equation (3.7) by the fourth-order accurate constrained RKDG method (2D Constrained RKDG4) with different numbers of cells used for conservation penalty. L^1 and L^∞ errors. $\mu = 1$. $T = 0.5/\pi$.

h	2D Constrained RKDG4-13Cell, CFL = 1.4				2D Constrained RKDG4-7Cell, CFL = 0.35			
	L^1 error	L^1 order	L^∞ error	L^∞ order	L^1 error	L^1 order	L^∞ error	L^∞ order
$\frac{1}{160}$	1.55E-7	-	6.38E-7	-	5.22E-8	-	2.56E-7	-
$\frac{1}{320}$	9.36E-9	4.05	5.19E-8	3.62	3.41E-9	3.94	1.73E-8	3.89
$\frac{1}{640}$	5.89E-10	3.99	3.29E-9	3.98	2.21E-10	3.95	1.18E-9	3.87
$\frac{1}{1280}$	3.73E-11	3.98	2.39E-10	3.78	1.44E-11	3.94	8.41E-11	3.81

We compute the numerical solutions using 400 equal size cells. The density profiles of the solutions are plotted at the time $T = 0.038$ in Fig. 3. We can clearly see that the fourth-order accurate constrained RKDG solution and the fourth-order accurate RKDG solution have almost identical resolution for the 1D Woodward-Colella blast wave problem; while the resolution of the third-order accurate constrained RKDG solution is comparable with that of the third-order accurate RKDG solution for this test case.

3.5.3 1D Lax problem

The 1D Lax problem [8] is the Euler equations with the Lax's initial data: the density ρ, momentum ρv and total energy E are 0.445, 0.311 and 8.928 in $(-1, 0)$; and are 0.5, 0 and 1.4275 in $(0, 1)$. We compute the numerical solutions using 200 equal size cells. The density profiles of the solutions are plotted at the time $T = 0.26$ in Fig. 4.

From these 1D compressible gas flow test problems, we conclude that the constraint RKDG method combined with HR limiter, gives good quality results for problems containing strong shock waves in the solution.

3.6 Test case using 2D Euler equations with discontinuous solutions

We test 2D problems with discontinuities in solutions to assess the non-oscillatory property of numerical solutions computed by the 2D constrained RKDG method together with HR limiter, again using the Euler equations for gas dynamics.

Example 3.6.3. Double Mach reflection. The Double Mach reflection problem is taken from [23]. We solve the Euler equations in a rectangular computational domain of $[0, 4] \times [0, 1]$. A reflecting wall lies at the bottom of the domain starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is located at $x = \frac{1}{6}, y = 0$, making a 60^0 angle with the x axis and extends to the top of the computational domain at $y = 1$. The reflective boundary condition is used at the wall.

We test our method on unstructured meshes with the triangle edge length roughly equal to $\frac{1}{400}$. The density contour of the flow in the $[0, 3] \times [0, 1]$ region at the time $t = 0.2$ is shown with 30 equally spaced contour lines. Fig. 5 is the contour plot of the numerical solutions computed by the 3^{rd} order RKDG and constrained RKDG methods respectively. Fig. 6 shows the "blown-up" portion around the double Mach region. We can see that while both of the RKDG and constrained RKDG methods successfully reproduce the vortex sheet roll-up; the solution computed by the constrained RKDG method is better than the one computed by the RKDG method, namely the constrained RKDG method picks up more roll-up and computes smoother contour lines.

Example 3.6.4. Flow past a forward facing step. This flow problem is again taken from [23]. The setup of the problem is the following: a right-going Mach 3 uniform flow enters a wind tunnel of 1 unit wide and 3 units long. The step is 0.2 units high and is located 0.6 units from the left side of the tunnel. The problem is initialized by a uniform, right-going

Mach 3 flow, which has density 1.4, pressure 1.0, and velocity 3.0. The initial state of the gas is also used at the left side boundary. At the right side boundary, the out-flow boundary condition is applied there. Reflective boundary condition is applied along the walls of the tunnel.

The corner of the step is a singularity. Unlike in [23] and in other studies, we do not modify our scheme near the corner, which is known to lead to an erroneous entropy layer at the downstream bottom wall, as well as a spurious Mach stem at the bottom wall. Instead, we use the approach taken in [4], which is to locally refine the mesh near the corner, to decrease these artifacts. The edge length of the triangle away from the corner is roughly equal to $\frac{1}{160}$. Near the corner, the edge length of the triangle is roughly equal to $\frac{1}{320}$. Fig. 7 is the contour plot of the numerical solutions computed by the 3^{rd} order RKDG and constrained RKDG methods respectively. Comparing results in Fig. 7, we can see that the resolution of the solution computed by the constrained RKDG method is better, especially for the contour lines around the triple point. Smoother contour lines are obtained in the constrained RKDG case. The 4^{th} order constrained RKDG solution captures more roll-ups on the vortex sheet. See Fig. 7(c).

3.7 Remark on Computational Cost of Constrained RKDG Method

To estimate the computational cost of the (third order) constrained RKDG method, we use the 2D Burgers' equation with a smooth solution as a test case. See Section 3.4 for the description of this benchmark problem. We employ a mesh with the triangle edge length roughly equal to $\frac{1}{128}$. The code is written in C and is compiled with "g++ -O3". Simulations are performed on a Linux workstation with an Intel i7 2.93 GHz processor. Table 21 shows cpu times spent by the RKDG and constrained RKDG methods for CFL = 0.2 and 0.8 respectively.

Table 21: Cpu time comparison between the third order RKDG and constrained RKDG methods

	CFL = 0.2	CFL = 0.8
RKDG	cpu time = sec.	-
Constrained RKDG	cpu time = sec.	cpu time = sec.

4 Concluding Remarks

In this work, we have developed a conservation constrained RKDG method for solving conservation Laws. The new formulation requires the computed RKDG solution defined on a cell to satisfy additional conservation constraints in adjacent cells (in the least-square sense) and does not increase the complexity or change the compactness of the original RKDG method. This conservation constrained RKDG method improves the CFL number over the RKDG method. The CFL number is improved around $5 \sim 6$ times for the fourth-order

accurate case and around $2 \sim 3$ times for the third-order accurate case on 2D triangular meshes. Moreover, for the multi-dimensional test problems with discontinuous solutions, the constrained RKDG method together with HR limiter also produces comparable resolutions of discontinuous solution compared with the RKDG scheme.

In the future, we will explore the higher order (> 4) constrained DG formulation in multi-dimensions with TVD Runge-Kutta time-stepping as well as other time-stepping methods and develop better compact limiting methods so that we will be able to study shock wave problems with better resolutions.

Acknowledgment

Simulations were performed on the Notre Dame Center for Research Computing (CRC) High Performance Computing (HPC) supercomputer (http://crc.nd.edu).

References

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Comput. Phys.*, 114:45–58, 1994.

[2] R. Biswas, K. D. Devine and J. E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.

[3] A. Christlieb, B. Ong and J.-M. Qiu. Integral Deferred Correction Methods constructed with High Order Runge-Kutta Integrators. *Math. Comp.*, 79:761-783, 2010.

[4] B. Cockburn and C.-W. Shu. The TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.

[5] B. Cockburn, S. Hou and C.-W. Shu. The TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[6] B. Cockburn, S.-Y. Lin and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *J. Comput. Phys.*, 52:411–435, 1989.

[7] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comp.*, 52:411–435, 1989.

[8] P. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computations. *Comm. Pure Appl. Math.*, 7:159, 1954.

[9] B. van Leer. Toward the ultimate conservative difference scheme: II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361–370, 1974.

[10] B. van Leer. Towards the ultimate conservative difference scheme: IV. A new approach to numerical convection. *J. Comput. Phys.*, 23:276–299, 1977.

[11] B. van Leer. Towards the ultimate conservative difference scheme: V. A second order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[12] B. van Leer and S. Nomura. Discontinuous Galerkin for diffusion. *AIAA-2005-5108*, 2005.

[13] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang. Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction. *SIAM J. Numer. Anal.*, 45:2442–2467, 2007.

[14] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang. Non-oscillatory hierarchical reconstruction for central and finite volume schemes. *Comm. Comput. Phys.*, 2:933–963, 2007.

[15] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang. L^2 stability analysis of the central discontinuous Galerkin method and a comparison between the central and regular discontinuous Galerkin methods. *ESAIM: Math. Mod. Numer. Anal.*, 42:593–607, 2008.

[16] H. Luo, J.D. Baum and R. Lohner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, *J. Comput. Phys.*, 225:686–713, 2007.

[17] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method. II: Two dimensional case. *Comput. Fluids*, 34:642–663, 2005.

[18] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Tech. report la-ur-73-479, Los Alamos Scientific Laboratory, 1973.

[19] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.

[20] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. *In Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, Berlin. Springer.* , 1697, 1998.

[21] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[22] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.*, 83:32–78, 1989.

[23] P. Woodward and P. Colella. Numerical simulation of two-dimensional fluid flows with strong shocks. *J. Comput. Phys.*, 54:115, 1984.

[24] T. Warburton and T. Hagstrom. Taming the CFL number for discontinuous Galerkin methods on structured meshes. *SIAM J. Numer. Anal.*, 46: 3151–3180, 2008.

[25] Z.-L. Xu and Y.-J. Liu and C.-W. Shu. Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO type linear reconstruction and partial neighboring cells, *J. Comput. Phys.*, 228:2194–2212, 2009.

[26] Z.-L. Xu and G. Lin. Spectral/hp element method with hierarchical reconstruction for solving nonlinear hyperbolic conservation laws, *Acta Mathematica Scientia*, 29(B):1737–1748, 2009.

[27] T. Zhou, Y.-F. Li and C.-W. Shu. Numerical Comparison of WENO Finite Volume and Runge-Kutta Discontinuous Galerkin Methods. *J. Sci. Comput.*, 16(2):145–171, 2001.

[28] J. Zhu, J.-X. Qiu, C.-W. Shu and M. Dumbser. Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. *J. Comput. Phys.*, 227:4330–4353, 2008.

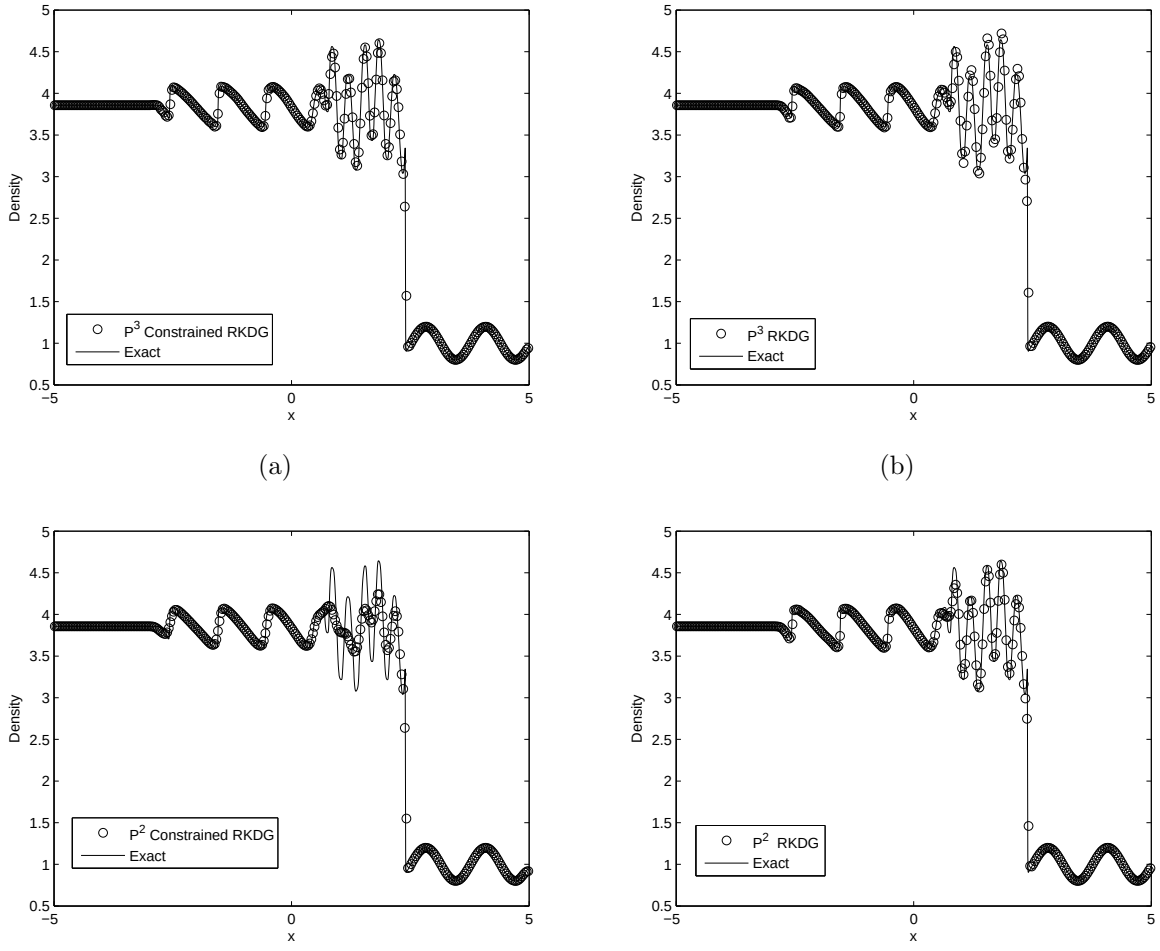


Figure 2: Solutions of the 1D Shu-Osher problem computed on 300 cells. (a) The fourth-order accurate constrained RKDG solution compared with the "exact" solution; (b) the fourth-order accurate RKDG solution compared with the "exact" solution; (c) the third-order accurate constrained RKDG solution compared with the "exact" solution; (b) the third-order accurate RKDG solution compared with the "exact" solution.

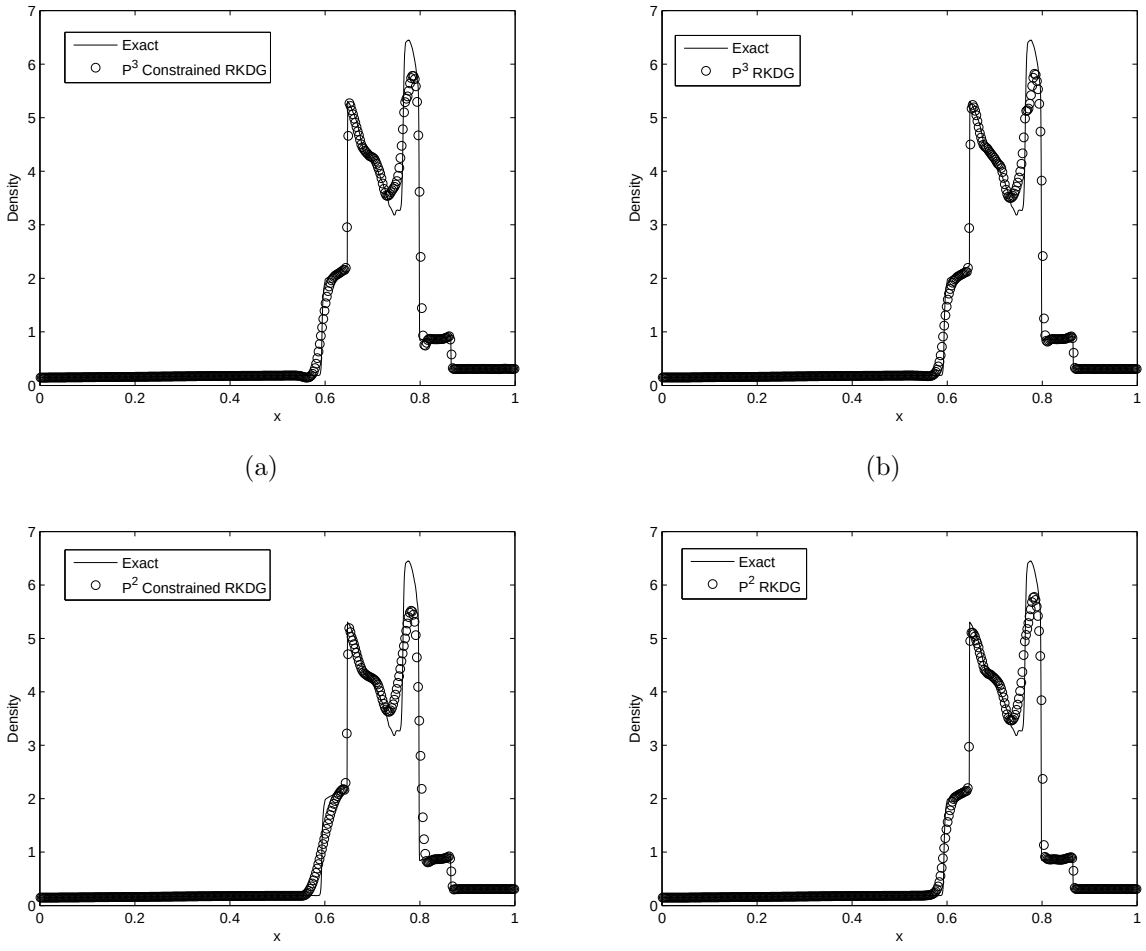


Figure 3: Solutions of the 1D blast wave problem computed on 400 cells. (a) The fourth-order accurate constrained RKDG solution compared with the "exact" solution; (b) the fourth-order accurate RKDG solution compared with the "exact" solution; (c) the third-order accurate constrained RKDG solution compared with the "exact" solution; (b) the third-order accurate RKDG solution compared with the "exact" solution.

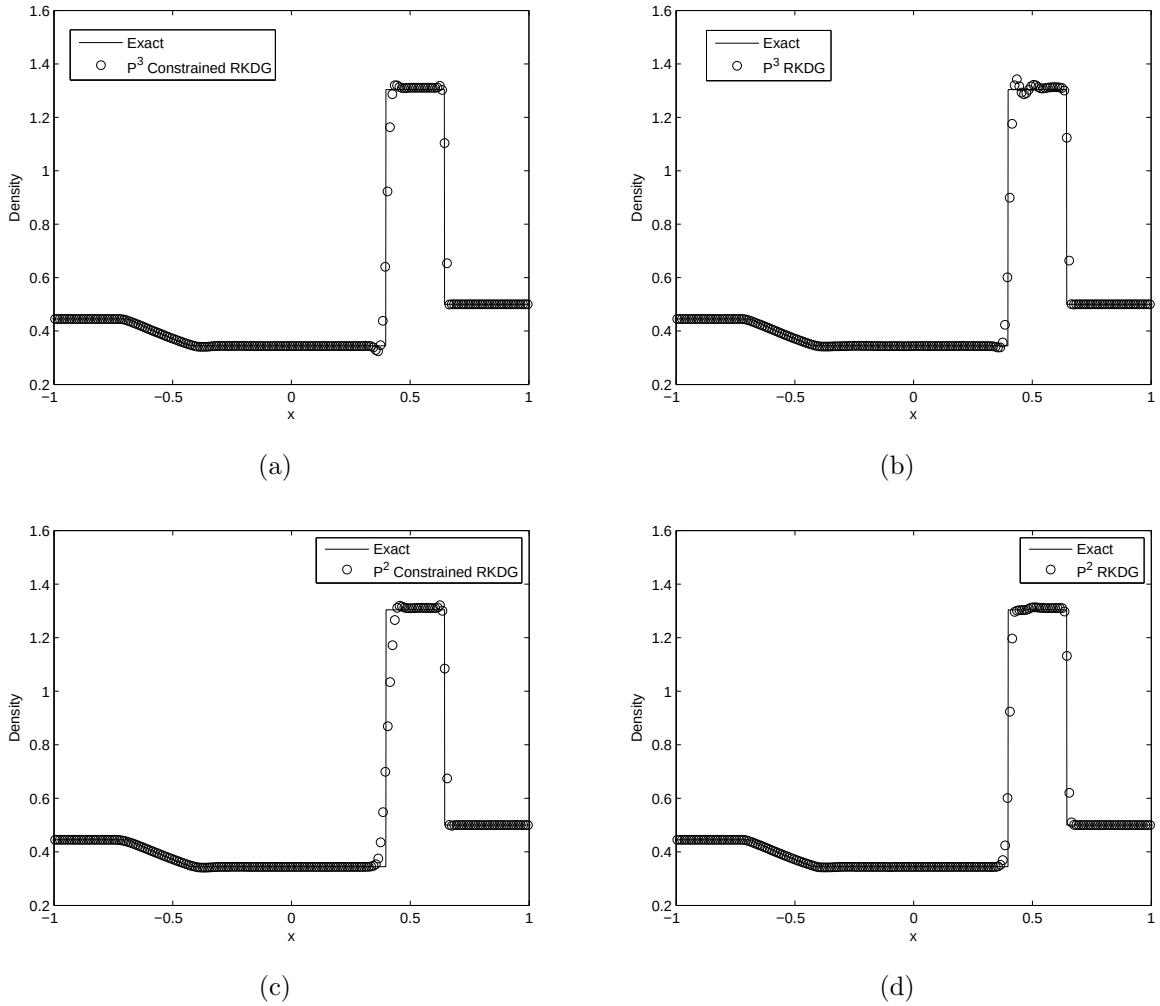
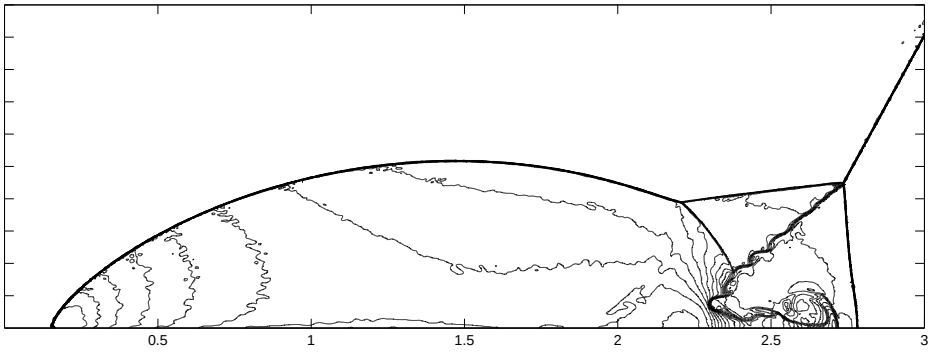
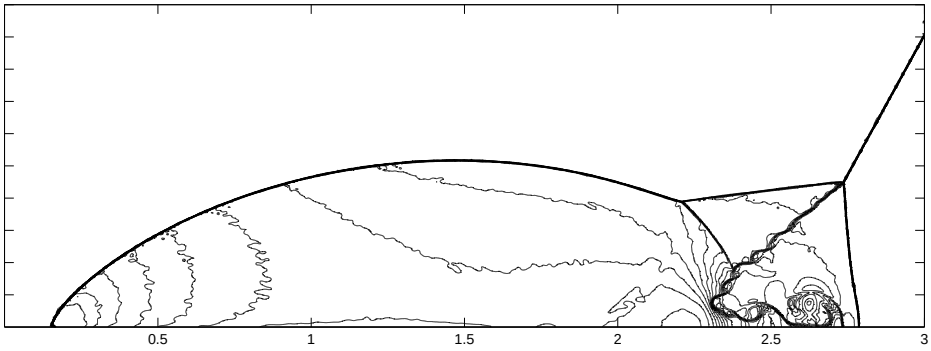


Figure 4: Solutions of the 1D Lax shock tube problem computed on 200 cells. (a) The fourth-order accurate constrained RKDG solution compared with the "exact" solution; (b) the fourth-order accurate RKDG solution compared with the "exact" solution; (c) the third-order accurate constrained RKDG solution compared with the "exact" solution; (b) the third-order accurate RKDG solution compared with the "exact" solution.

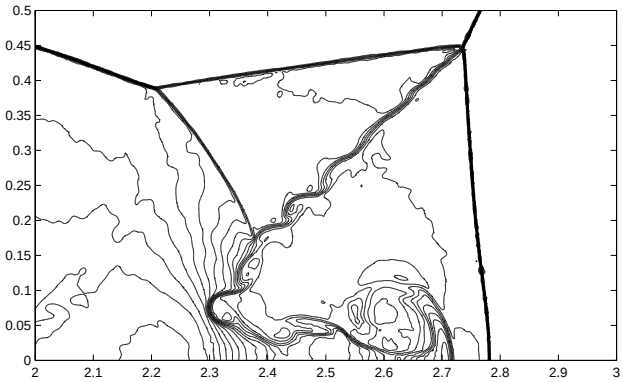


(a)

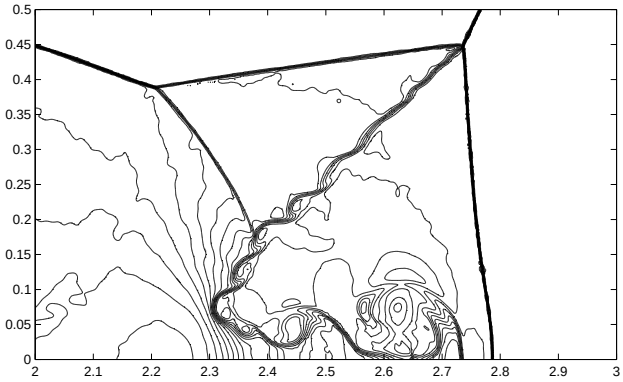


(b)

Figure 5: Double Mach reflection problem. Third-order results. Density ρ. (a) The 3^{rd} order RKDG solution; (b) The 3^{rd} order constrained RKDG solution.

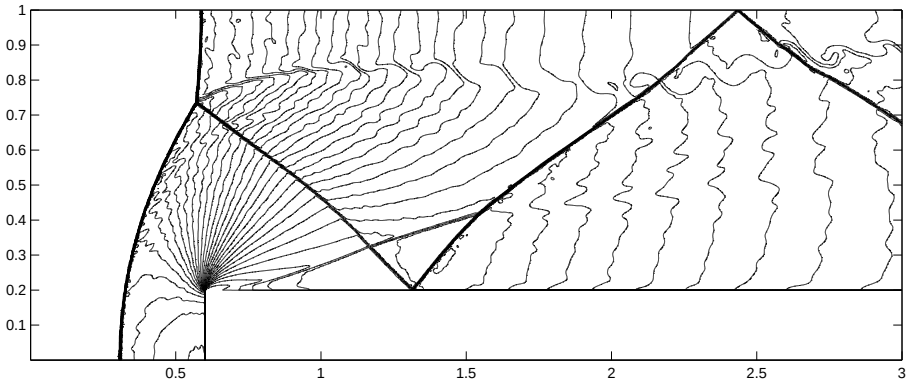


(a)

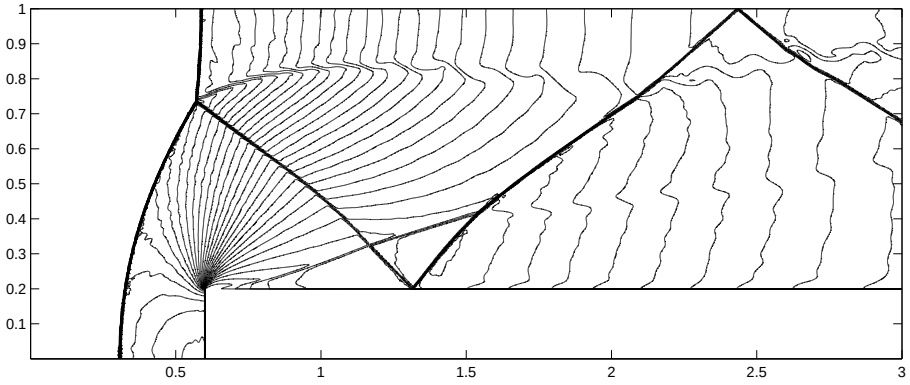


(b)

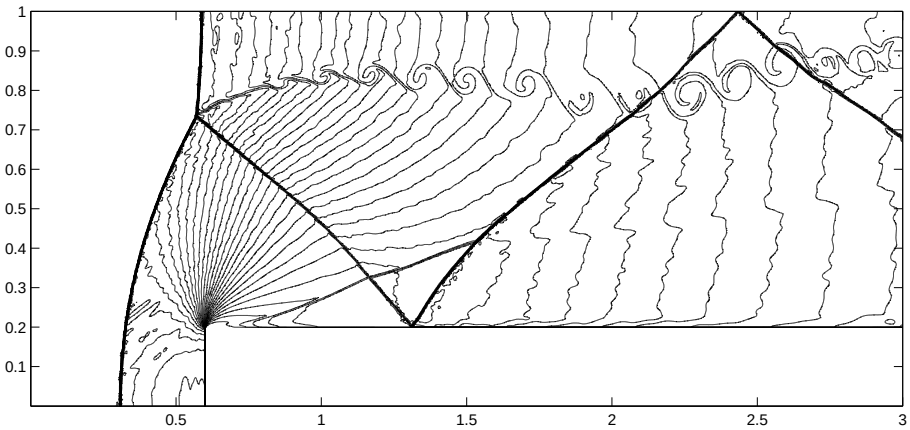
Figure 6: Double Mach reflection problem. Blown-up region around the double Mach stems. Third-order results. Density ρ. (a) The 3^{rd} order RKDG solution; (b) The 3^{rd} order constrained RKDG solution.



(a)



(b)



(c)

Figure 7: Forward-facing step problem. Third-order results. Density ρ. (a) The 3^{rd} order RKDG solution; (b) The 3^{rd} order constrained RKDG solution; (c) The 4^{th} order constrained RKDG solution.