THE RUNGE–KUTTA DISCONTINUOUS GALERKIN METHOD WITH COMPACT STENCILS FOR HYPERBOLIC CONSERVATION LAWS

QIFAN CHEN*, ZHENG SUN†, AND YULONG XING‡

Abstract. In this paper, we develop a new type of Runge–Kutta (RK) discontinuous Galerkin (DG) method for solving hyperbolic conservation laws. Compared with the original RKDG method, the new method features improved compactness and allows simple boundary treatment. The key idea is to hybridize two different spatial operators in an explicit RK scheme, utilizing local projected derivatives for inner RK stages and the usual DG spatial discretization for the final stage only. Limiters are applied only at the final stage for the control of spurious oscillations. We also explore the connections between our method and Lax–Wendroff DG schemes and ADER-DG schemes. Numerical examples are given to confirm that the new RKDG method is as accurate as the original RKDG method, while being more compact, for problems including two-dimensional Euler equations for compressible gas dynamics.

Key words. Discontinuous Galerkin method, Runge–Kutta method, stencil size, high-order numerical method, convergence, hyperbolic conservation laws

AMS subject classifications. 65L06, 65M12, 65M20, 65M60

1. Introduction.

In this paper, we present a novel class of high-order Runge–Kutta (RK) discontinuous Galerkin (DG) methods for solving hyperbolic conservation laws. Compared with the original RKDG method proposed in [6–10], the new method features more compact stencil sizes and will be hence referred to as the compact RKDG (cRKDG) method throughout the paper.

The RKDG method for conservation laws was originally proposed by Cockburn et al. in a series of papers [6–10]. The method combines the DG finite element spatial discretization [25] with the strong-stability-preserving (SSP) RK time discretization [16, 17]. A limiting procedure is employed to control oscillations near physical discontinuities. The method naturally preserves the local conservation, features good *hp* adaptivity, and can be fitted into complex geometries. Due to its various advantages, the RKDG method has become one of the primary numerical methods for the simulation of hyperbolic conservation laws.

This paper aims to further improve the RKDG method by reducing its stencil size within each time step, which can reduce its data communication and lead to potential advantages in parallel computing or implicit time marching. The DG spatial discretization is typically more compact when compared to the finite difference method of the same order. For example, when approximating the first-order spatial derivatives of a function on a given cell, the DG method only utilizes data from immediate neighbors, while the finite difference method may require data from farther nodes for a high-order approximation. However, this spatial discretization advantage is not preserved by the one-step multi-stage RK time stepping. In each temporal stage, the spatial operator calls for information from neighboring cells, hence the stencil of the scheme will be expanded after each stage. For example, for one-dimensional scalar

*Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA. (chen.11010@osu.edu)

†Department of Mathematics, The University of Alabama, Tuscaloosa, AL 35487, USA. (zsun30@ua.edu)

‡Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA. (xing.205@osu.edu)

conservation laws with Lax–Friedrichs flux, the stencil size is 3 for the first-order forward-Euler time stepping, but becomes $2s+1$ for the high-order RK time stepping with s stages. See Figure 3.1 for an illustration. This issue could be more evident with a very high-order RK method in multidimensions.

There are some existing techniques in the literature circumventing the aforementioned issue. The main idea is to construct a one-step one-stage method for time marching. For instance, the Lax–Wendroff temporal discretization can be used as an alternative, which leads to the so-called Lax–Wendroff DG (LWDG) method [22, 23]. This method utilizes only information from immediate neighbors regardless of its temporal order. However, implementing this method can be very tedious, especially for high-order schemes for multidimensional systems, as one needs to compute the high-order derivatives of the flux function in the Cauchy–Kowalewski procedure. Another avenue is to employ the Arbitrary DERivative (ADER) time stepping [29,30], resulting in the so-called ADER-DG method, which is presented in the spacetime integral form of the conservation laws and is known to be closely connected to the LWDG scheme. See [12, 14, 15, 24] and references therein. Besides LWDG and ADER-DG methods, there is also a stream of research addressing the compactness of DG methods for discretizing the second or higher-order spatial derivatives [1–3, 5, 20, 21, 31, 32, 34]. These studies focus on reducing the stencils of spatial operators but are less related with the issue arising from the multi-stage RK time stepping. If RK methods are used to discretize the corresponding semi-discrete DG schemes for a time-dependent problem, the stencil size of the fully discrete schemes still grows with the number of RK stages.

In this paper, we propose a very different novel approach to tackle this issue. Our method is still based on the RK methods for temporal discretization, specifically using the Butcher form instead of the Shu–Osher form [16]. The key idea is to hybridize two different types of spatial operators within each time step. For the inner stage(s) of the RK method, we employ the local derivative operator, which returns the L^2 projection of the spatial derivative of the flux function. While for the very last stage, we use the DG operator as in the original DG scheme. A limiter will be applied only once at the end of each time step, if necessary. The proposed cRKDG method has the following desirable properties.

- Stencil size: In each time step, the stencil of the cRKDG method only contains the current cell and its immediate neighbors, resulting in a compact stencil.
- Convergence: We prove that if the cRKDG method converges boundedly, then its limit is a weak solution of the conservation laws.
- Accuracy: We numerically observe that the cRKDG method attains $(k+1)$th order convergence rate when we couple a $(k+1)$th order RK method with a DG method using kth order spatial polynomials. This matches the optimal convergence of the original RKDG method.
- Boundary error: We numerically observe that the cRKDG method attains the same optimal convergence rate when nonhomogeneous Dirichlet boundary conditions are imposed. In contrast, the original RKDG method will suffer the accuracy degeneracy under this setting [35].

In addition to the aforementioned analysis and numerical observations, we conduct a linear stability analysis for the cRKDG method and identify its maximum CFL numbers for both second- and third-order cases. We employ the standard von Neumann analysis and investigate the eigen-structures of the amplification matrices. This approach aligns with the methodologies employed in previous works such as [11] for the RKDG method and [23] for the LWDG method. The details are omitted due to the space constraint. The maximum CFL numbers are documented in Table 1.1.

We can see from the table that the CFL number of the cRKDG method is the same as that of the RKDG method for the second-order case, and is slightly smaller for the third-order case. While comparing to the LWDG method, the CFL numbers of the cRKDG method are larger.

Maximum CFL	cRKDG	RKDG [11]	LWDG [23]
Second-order	0.333	0.333	0.223
Third-order	0.178	0.209	0.127

Table 1.1: Maximum CFL numbers of the cRKDG method, the RKDG method, and the LWDG method. Here the upwind flux is used for cRKDG method and the RKDG method, and the Lax–Friedrichs flux (3.15) is used for the LWDG method [23].

For general nonlinear conservation laws, the cRKDG method proposed in this paper is different from the LWDG and ADER-DG methods. However these methods are closely connected. Indeed, for linear conservation laws with constant coefficients, the cRKDG method and the LWDG method are actually equivalent under certain choice of the numerical fluxes,[1] although the two methods are designed from totally different perspectives. Furthermore, the cRKDG method with a special implicit RK method is equivalent to the ADER-DG method with a special local predictor. As a result, one can expect that the cRKDG method may share some similar properties with the LWDG and ADER-DG methods. These connections may bring in new perspectives that could contribute to the further development and understanding of existing methods.

The rest of the paper is organized as follows. In Section 2, we review the original RKDG method and explain the formulation of the novel cRKDG method. In Section 3, we summarize the theoretical properties of the cRKDG schemes and postpone some of the technical proofs to the appendix. Numerical tests are provided in Section 4 and the conclusions are given in Section 5.

2. Numerical schemes.

In this section, we start by briefly reviewing the RKDG method and then describe in detail the construction of the cRKDG method. For ease of notation, we focus on scalar conservation laws, but the method can be extended to systems of conservation laws straightforwardly.

2.1. RKDG schemes.

Consider the hyperbolic conservation laws

$$\partial_t u + \nabla \cdot f(u) = 0, \qquad u(x,0) = u_0(x). \tag{2.1}$$

Let $\mathcal{T}_h = \{K\}$ be a partition of the spatial domain in d dimension. We denote by h_K the diameter of K and $h = \max_{K\in\mathcal{T}_h} h_K$. Let ∂K be the boundary of K. For each edge $e \in \partial K$, $\nu_{e,K}$ is the unit outward normal vector along e with respect to K. The finite element space of the DG approximation is defined as $V_h = \{v : v|_K \in \mathcal{P}^k(K), \forall K \in \mathcal{T}_h\}$, where $\mathcal{P}^k(K)$ denotes the set of polynomials of degree up to k on the cell K. The standard semi-discrete DG method for solving (2.1) is defined as follows: find $u_h \in V_h$ such that on each $K \in \mathcal{T}_h$,

$$\int_K (u_h)_t \, v_h \mathrm{d}x - \int_K f(u_h) \cdot \nabla v_h \mathrm{d}x + \sum_{e\in\partial K} \int_e \widehat{f \cdot \nu_{e,K}} v_h \mathrm{d}l = 0, \quad \forall v_h \in V_h. \tag{2.2}$$

[1] While with the numerical flux in [23], the LWDG method is not equivalent to the cRKDG method even in the linear case. See Remark 3.4. This fact explains the different CFL numbers in Table 1.1.

Here $\widehat{f \cdot \nu_{e,K}}$ is the numerical flux, which can be computed from the exact or approximate Riemann solver defined at the cell interface. For example, we can choose the Lax–Friedrichs flux of the form

$$\widehat{f \cdot \nu_{e,K}} = \frac{1}{2}\left(f(u_h^{\text{int}}) \cdot \nu_{e,K} + f(u_h^{\text{ext}}) \cdot \nu_{e,K} - \alpha_{e,K}\left(u_h^{\text{ext}} - u_h^{\text{int}}\right)\right),$$

with $\alpha_{e,K} = \max|\partial_u f \cdot \nu_{e,K}|$. Here u_h^{int} and u_h^{ext} are limits of u_h along e from the interior and exterior of the cell K.

We introduce the discrete operator $\nabla^{\text{DG}} \cdot f : V_h \to V_h$, defined by

$$\int_K \nabla^{\text{DG}} \cdot f(u_h) v_h \mathrm{d}x = -\int_K f(u_h) \cdot \nabla v_h \mathrm{d}x + \sum_{e \in \partial K} \int_e \widehat{f \cdot \nu_{e,K}} v_h \mathrm{d}l, \quad \forall v_h \in V_h. \tag{2.3}$$

Therefore the semi-discrete DG scheme (2.2) can be rewritten in the strong form

$$\partial_t u_h + \nabla^{\text{DG}} \cdot f(u_h) = 0. \tag{2.4}$$

Then we apply an explicit RK method to discretize (2.4) in time. Consider an explicit RK method associated with the Butcher Tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}, \quad A = (a_{ij})_{s\times s}, \quad b = (b_1, \cdots, b_s), \tag{2.5}$$

where A is a lower triangular matrix in (2.5), namely, $a_{ij} = 0$ if $i > j$. The corresponding RKDG scheme is given by

$$u_h^{(i)} = u_h^n - \Delta t \sum_{j=1}^{i-1} a_{ij} \nabla^{\text{DG}} \cdot f\left(u_h^{(j)}\right), \quad i = 1, 2, \cdots, s, \tag{2.6a}$$

$$u_h^{n+1} = u_h^n - \Delta t \sum_{i=1}^{s} b_i \nabla^{\text{DG}} \cdot f\left(u_h^{(i)}\right). \tag{2.6b}$$

Note we have $u_h^{(1)} = u_h^n$ for the explicit RK method. In the case that the problem is nonautonomous, for example, when a time-dependent source term $q(t)$ is included, $q(t + c_j\Delta t)$ should be included at appropriate places of the RK stages.

2.2. cRKDG schemes. Similarly to (2.3), we define a local discrete spatial operator $\nabla^{\text{loc}} \cdot f : V_h \to V_h$ such that

$$\int_K \nabla^{\text{loc}} \cdot f\left(u_h\right) v_h \mathrm{d}x = -\int_K f\left(u_h\right) \cdot \nabla v_h \mathrm{d}x + \sum_{e \in \partial K} \int_e f(u_h) \cdot \nu_{e,K} v_h \mathrm{d}l, \quad \forall v_h \in V_h. \tag{2.7}$$

In other words, instead of using a numerical flux involving u_h on both sides of the cell interfaces, the values of u_h along ∂K are taken from the interior of the cell K. Therefore it is a local operation defined within K without couplings with the neighboring cells. It is easy to see that when all integrals in (2.7) are computed exactly, $\nabla^{\text{loc}} \cdot f$ indeed returns the projected local derivative

$$\nabla^{\text{loc}} \cdot f\left(u_h\right) = \Pi \nabla \cdot f\left(u_h\right),$$

where Π is the L^2 projection to V_h.

With $\nabla^{\mathrm{loc}} \cdot f$ defined above, we propose our new cRKDG scheme in the following Butcher Tableau form:

$$u_h^{(i)} = u_h^n - \Delta t \sum_{j=1}^{i-1} a_{ij} \nabla^{\mathrm{loc}} \cdot f\left(u_h^{(j)}\right), \quad i = 1, 2, \cdots, s, \tag{2.8a}$$

$$u_h^{n+1} = u_h^n - \Delta t \sum_{i=1}^{s} b_i \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(i)}\right). \tag{2.8b}$$

The main difference with the original RKDG scheme (2.6) is to use the local operator $\nabla^{\mathrm{loc}} \cdot f$ instead of $\nabla^{\mathrm{DG}} \cdot f$ when evaluating the inner stage values $u_h^{(i)}$ in (2.8a).

For optimal convergence, we will couple $(k+1)$th order RK method with $\mathcal{P}^k$ spatial elements. The resulted fully discrete scheme is $(k+1)$th order accurate, which is the same optimal rate as that of the original RKDG method. For clarity, we list second- and third-order cRKDG schemes below as examples.

Second-order scheme ($k = 1$).

$$u_h^{(2)} = u_h^n - \frac{\Delta t}{2} \nabla^{\mathrm{loc}} \cdot f\left(u_h^n\right), \tag{2.9a}$$

$$u_h^{n+1} = u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(2)}\right). \tag{2.9b}$$

Third-order scheme ($k = 2$).

$$u_h^{(2)} = u_h^n - \frac{1}{3}\Delta t \nabla^{\mathrm{loc}} \cdot f\left(u_h^n\right), \qquad u_h^{(3)} = u_h^n - \frac{2}{3}\Delta t \nabla^{\mathrm{loc}} \cdot f\left(u_h^{(2)}\right), \tag{2.10a}$$

$$u_h^{n+1} = u_h^n - \Delta t \left(\frac{1}{4}\nabla^{\mathrm{DG}} \cdot f\left(u_h^n\right) + \frac{3}{4}\nabla^{\mathrm{DG}} \cdot f\left(u_h^{(3)}\right)\right). \tag{2.10b}$$

Remark 2.1 (RK methods in Butcher form). For the original RKDG method, strong-stability-preserving Runge–Kutta (SSP-RK) methods are often adopted as time stepping methods. However, we address that the cRKDG method (2.8) has to be written based on the Butcher form of RK methods, in order to preserve local conservation and achieve optimal accuracy. We refer to Section 3.2.1 and Example 4.2 for further details.

Since the cRKDG method is not based on SSP-RK time stepping, one can choose from a larger class of RK schemes without worrying about order barriers brought by the SSP property [16]. For example, the classical four-stage fourth-order RK method can be used for the fourth-order scheme.

Remark 2.2 (Limiters). To suppress spurious oscillation near discontinuities, a minmod or WENO type limiter is often applied after the update of each inner stage value $u_h^{(i)}$ in the original RKDG method (2.6). For the cRKDG method, the limiter can be applied at the end of each time step. This is similar to the limiting strategy for LWDG method in [23] and will not change the stencil size of the cRKDG scheme. But at the same time, it's worth noting that the total-variation boundedness property, guaranteed for the original RKDG method [8], may not hold in this case.

3. Properties of the cRKDG method.

3.1. Stencil size. In the cRKDG method, all inner stages are discretized with a local operator only using the information on the cell K. As a result, the stencil of the cRKDG scheme is determined by that of the last stage (2.8b) only, and its size is

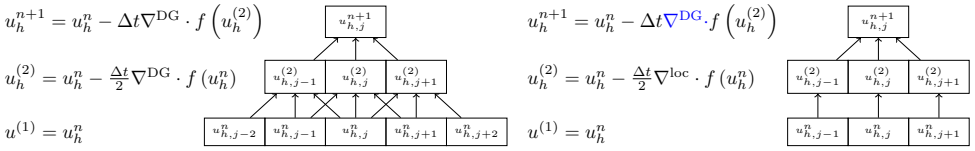


Fig. 3.1: Stencils of RKDG and cRKDG methods with a second-order RK method.

the same as the forward-Euler–DG scheme. For example, in the one-dimensional case with Lax–Friedrichs fluxes, regardless of the number of RK stages, the stencil size of the cRKDG scheme is identically 3.

In contrast, the stencil size of the RKDG scheme grows with the number of RK inner stages. In the one-dimensional case, the DG operator with Lax–Friedrichs flux has the stencil size 3. While after the temporal discretization with an s-stage RK method, the stencil size becomes $2s + 1$. See Figure 3.1 for an example with the second-order RK method. The difference on the stencil sizes between the two methods could be more significant in multidimensions.

Proposition 3.1. The stencil of a cRKDG method of any temporal order only involves the current mesh cell and its immediate neighbors.

3.2. Convergence.

3.2.1. RK methods in Butcher form. SSP-RK methods are usually used for time discretization in the original RKDG method. For example, the widely used second-order SSP-RKDG method is given by

$$\begin{aligned} u_h^{(1)} =& u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot f(u_h^n), \\ u_h^{n+1} =& \frac{1}{2} u_h^n + \frac{1}{2}\left(u_h^{(1)} - \Delta t \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(1)}\right)\right). \end{aligned}$$

As above, an SSP-RK method can be written as convex combinations of forward-Euler steps, which is also referred to as the *Shu–Osher form* in the literature. The method can also be written in the equivalent *Butcher form* as those in (2.6).

In the design of the cRKDG method, one cannot directly replace $\nabla^{\mathrm{DG}} \cdot f$ with $\nabla^{\mathrm{loc}} \cdot f$ in the inner stages of an SSP-RKDG method in its Shu–Osher form, as this will cause accuracy reduction and violation of local conservation. For example, the following scheme is suboptimal and nonconservative:

$$u_h^{(1)} = u_h^n - \Delta t \nabla^{\mathrm{loc}} \cdot f(u_h^n), \tag{3.1a}$$

$$u_h^{n+1} = \frac{1}{2} u_h^n + \frac{1}{2}\left(u_h^{(1)} - \Delta t \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(1)}\right)\right). \tag{3.1b}$$

This can be made clear by substituting (3.1a) into (3.1b):

$$u_h^{n+1} = u_h^n - \frac{\Delta t}{2}\left(\nabla^{\mathrm{loc}} \cdot f\left(u_h^n\right) + \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(1)}\right)\right). \tag{3.2}$$

One can see that a low-order and nonconservative spatial operator $\nabla^{\mathrm{loc}} \cdot f$ is used to update u_h^{n+1}, which will cause trouble. Indeed, if we consider its Butcher form, then the resulting cRKDG method can be written as

$$u_h^{(1)} = u_h^n - \Delta t \nabla^{\mathrm{loc}} \cdot f(u_h^n), \tag{3.3a}$$

$$u_h^{n+1} = u_h^n - \frac{\Delta t}{2}\left(\nabla^{\mathrm{DG}} \cdot f\left(u_h^n\right) + \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(1)}\right)\right), \tag{3.3b}$$

i.e., replacing the last step (3.1b) or (3.2) by (3.3b). The resulting scheme will have optimal convergence and the provable local conservation property. Some numerical tests will be given in Example 4.2 to illustrate the different convergence rates of these methods. Note that (3.3) and (2.9) are both second-order cRKDG schemes, although they are different as the associated RK methods are different.

In general, for the RKDG method in Butcher form, we can prove that replacing inner stages with $\nabla^{\mathrm{loc}} \cdot f$ will retain local conservation property (see Theorem 3.1), and the optimal convergence rate can still be observed numerically (see Example 4.2).

3.2.2. Convergence. Although all inner stages of the cRKDG method (2.8a) do not preserve local conservation, the fully discrete numerical scheme is still conservative as long as the update of u_h^{n+1} in the last stage is discretized with a conservative method. Indeed, for explicit RK methods, the cRKDG scheme can be formally written as a one-step scheme after recursive substitutions. By taking $v_h = 1_K$, the characteristic function on K, in (2.8b) and combining with (2.3), we obtain

$$\bar{u}_{h,K}^{n+1} = \bar{u}_{h,K}^{n} - \frac{\Delta t}{|K|} \sum_{e \in \partial K} g_{e,K}(u_h^n), \tag{3.4}$$

where the bar notation is used to represent the cell average,

$$g_{e,K}(u_h^n) = \int_e \left(\sum_{i=1}^{s} b_i \widehat{f \cdot \nu_{e,K}} \left(u_h^{(i)} \right) \right) \mathrm{d}l \tag{3.5}$$

and $u_h^{(i)}$ can be explicitly computed from (2.8a). (3.4) is written in the conservative form. Following [26], we obtain a Lax–Wendroff type theorem in Theorem 3.1, with its detailed proof given in Appendix A. Here we consider scalar conservation laws for simplicity, and the results can also be extended to multidimensional systems. We will use C for a generic constant that may depend on the polynomial order k, the stage number s, the bound of u and u_h, etc., but is independent of Δt and h.

THEOREM 3.1. *Consider the cRKDG scheme (2.8) for the hyperbolic conservation law (2.1) on quasi-uniform meshes with the following assumptions:*

1. *f is Lipschitz continuous and f', f'' are uniformly bounded in L^∞.*
2. *The numerical flux $\widehat{f \cdot \nu_{e,K}}(u_h)$ has the following properties:*
   (a) *Consistency: if $u_h = u_h^{\mathrm{int}} = u_h^{\mathrm{ext}}$, then $\widehat{f \cdot \nu_{e,K}}(u_h) = f(u_h) \cdot \nu_{e,K}$.*
   (b) *Lipschitz continuity: $|\widehat{f \cdot \nu_{e,K}}(u_h) - \widehat{f \cdot \nu_{e,K}}(v_h)| \le C \left\| u_h^n - v_h^n \right\|_{L^\infty(B_K)}$, where $B_K = K \cup K^{\mathrm{ext}}$ is the union of K and its neighboring cell K^{ext}.*
3. *The CFL condition $\Delta t / h \le C$ is satisfied.*

If u_h^n converges boundedly almost everywhere to some function u as $\Delta t, h \to 0$, then the limit u is a weak solution to the conservation law (2.1), namely

$$\int_{\mathbb{R}^d} u_0 \phi \mathrm{d}x + \int_{\mathbb{R}^d \times \mathbb{R}^+} u \phi_t \mathrm{d}x\mathrm{d}t + \int_{\mathbb{R}^d \times \mathbb{R}^+} f(u) \cdot \nabla \phi \mathrm{d}x\mathrm{d}t = 0, \quad \forall \phi \in C_0^\infty(\mathbb{R}^d \times \mathbb{R}^+). \tag{3.6}$$

3.3. Accuracy. In the cRKDG scheme, we apply $\nabla^{\mathrm{loc}} \cdot f$ to approximate spatial derivatives of f for all inner stages. However, $\nabla^{\mathrm{loc}} \cdot f$ may not have the same approximation property as $\nabla^{\mathrm{DG}} \cdot f$. (For example, for $\mathcal{P}^0$ elements, $\nabla^{\mathrm{loc}} \cdot f(u_h) = \Pi \nabla \cdot f(u_h) \equiv 0$.) A natural question to ask is whether the cRKDG scheme will still admit the optimal convergence rate in both space and time.

In [18], Grant analyzed perturbed RK schemes with mixed precision. It is pointed out that replacing certain stages in an RK scheme with a low-precision approximation may not affect the overall accuracy of the solver. The order conditions of the methods were systematically studied there. In particular, the work by Grant implies the following results.

THEOREM 3.2 (Grant, 2022. [18]). *Let $F^\varepsilon(u) = F(u) + \mathcal{O}(\varepsilon)$ be a low-precision perturbation of $F(u)$. Consider a pth order RK method for solving the ordinary differential equation $u_t = F(u)$:*

$$u^{(i)} = u^n + \sum_{j=1}^{s} a_{ij} F^\varepsilon\left(u_h^{(j)}\right), \quad i = 1, 2, \cdots, s, \tag{3.7a}$$

$$u^{n+1} = u^n + \sum_{j=1}^{s} b_j F\left(u_h^{(j)}\right). \tag{3.7b}$$

The local truncation error of the scheme (3.7) is $\mathcal{O}\left(\Delta t^{p+1}\right) + \mathcal{O}(\varepsilon \Delta t^2)$.

The numerical error of the cRKDG scheme (2.8) can be analyzed by considering $F = \nabla^{\mathrm{DG}} \cdot f$ and $F^\varepsilon = \nabla^{\mathrm{loc}} \cdot f$ in Theorem 3.2. Although $\nabla^{\mathrm{loc}} \cdot f$ could be low-order accurate, it may yield $\left\|\nabla^{\mathrm{DG}} \cdot f(u_h) - \nabla^{\mathrm{loc}} \cdot f(u_h)\right\| \le Ch^k$. Hence from Theorem 3.2, we expect that the local truncation error after one step is $\mathcal{O}(\Delta t^{p+1} + h^k \Delta t^2 + h^{k+1}\Delta t)$. Under the standard CFL condition for hyperbolic conservation laws, we have $\Delta t \le Ch$, hence the local truncation error is $\mathcal{O}(\Delta t^{\min(p+1,k+2)})$. Hence a heuristic global error estimate is $\|u_h^n(\cdot) - u(\cdot, t^n)\| = \mathcal{O}(\Delta t^{\min(p,k+1)})$, which is the same as the original RKDG scheme.

However, the above argument is far from rigorous error estimates. For example, estimates in Theorem 3.2 rely on the derivatives of F and $F - F^\varepsilon$, which are not defined for $\nabla^{\mathrm{DG}} \cdot f$ and $\nabla^{\mathrm{DG}} \cdot f - \nabla^{\mathrm{loc}} \cdot f$. A detailed fully discrete error analysis using an energy type argument is still needed and is postponed to our future work.

3.4. Boundary Error. When the nonhomogeneous Dirichlet boundary condition is used, it is known that, if one directly uses the exact inflow data for RK inner stages, the RKDG method for hyperbolic conservation laws may suffer order degeneration of the accuracy [35]. This order degeneration is not specific to DG schemes but can also arise with other spatial discretization methods [4].

The reason for such order deductions relates to the fact that some RK stages are designed to be of low stage order, which means that they should be low-order approximations of the true solutions at the corresponding stages. These low-order stages are combined in a subtle way with the coefficients in the Butcher Tableau so that they can build up a high-order accurate solution at t^{n+1}. If we replace the low-order RK stage with the exact boundary data, we also break the subtle cancellation of the error terms, which will lead to a low-order accurate approximation at t^{n+1}.

Compared with the RKDG method, the cRKDG method uses the local operator to approximate the values at the inner stages. By doing so, it does not need any exterior information and will avoid introducing the boundary data in the update of the inner stage values. Therefore, the boundary condition is only needed in the last stage to update u_h^{n+1}, and this will automatically maintain the optimal convergence rate of the cRKDG method. A detailed analysis involving the nonhomogeneous boundary condition will be provided in future work.

3.5. Connections with other DG methods.

3.5.1. Equivalence with Lax–Wendroff DG in special cases. DG methods with Lax–Wendroff type time discretization were studied in [23]. The main idea there was to consider the high-order temporal Taylor expansion

$$u(x,t+\Delta t) = u(x,t) + \Delta t u_t(x,t) + \frac{\Delta t^2}{2!}u_{tt}(x,t) + \frac{\Delta t^3}{3!}u_{ttt}(x,t) + \cdots \tag{3.8}$$

and then apply the Lax–Wendroff procedure (or the so-called Cauchy–Kowalewski procedure) to convert all temporal derivatives to spatial derivatives. For example, for the one-dimensional problem with a third-order expansion, this gives

$$u^{n+1} = u^n - \Delta t F(u)_x,$$

where

$$\begin{aligned} &F(u) = f(u) + \frac{\Delta t}{2} f'(u)u_t + \frac{\Delta t^2}{6}\left(f''(u)(u_t)^2 + f'(u)u_{tt}\right) + \cdots, \\ &u_t = -f(u)_x,\ u_{tt} = -\left(f'(u)u_t\right)_x,\ u_{ttt} = -\left(f''(u)(u_t)^2 + f'(u)u_{tt}\right)_x, \quad \text{etc..} \end{aligned}$$

The LWDG method [23] is then given as

$$u_h^{n+1} = u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot F\left(u_h^n\right)$$

with a suitable choice of the numerical flux $\widehat{F \cdot \nu_{e,K}}$.

Although for a generic nonlinear problem, the RK methods suffer the so-called order barrier and we typically need $s > p$ for high-order RK methods. While for linear autonomous problems, it is possible to construct a p-stage and pth order RK method [17]. The following theorem states that for linear conservation laws with constant coefficients and with a certain choice of the numerical flux, the LWDG method is the same as the cRKDG method with such p-stage and pth order RK method.

THEOREM 3.3. *Consider linear conservation laws with constant coefficients. Suppose $\nabla^{\mathrm{DG}} \cdot f$ is linear in the sense that*

$$\nabla^{\mathrm{DG}} \cdot f\left(\sum_{j=1}^{s} b_j u_h^{(j)}\right) = \sum_{j=1}^{s} b_j \nabla^{\mathrm{DG}} \cdot f\left(u_h^{(j)}\right).$$

Then when a specific numerical flux is chosen, as indicated in (3.14), the LWDG method with pth temporal order is equivalent to a cRKDG method using an explicit RK method of pth order with p stages.

Proof. For ease of notation, we will only consider the scalar conservation law $\partial_t u + \nabla \cdot (\beta u) = 0$, where β is a constant vector. But the argument can be similarly generalized to multidimensional linear systems with constant coefficients. We will also denote by $\nabla^{\mathrm{DG}} \cdot f(v_h) = \nabla^{\mathrm{DG}} \cdot (\beta v_h)$ and $\nabla^{\mathrm{loc}} \cdot f(v_h) = \nabla^{\mathrm{loc}} \cdot (\beta v_h)$ for any $v \in V_h$.

We will show that both schemes can be written in the following form.

$$u_h^{n+1} = u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot \left(\beta \sum_{i=0}^{p-1} \frac{\Delta t^i}{(i+1)!} L_h^i u_h^n\right), \quad \text{with} \quad L_h u_h^n := -\nabla^{\mathrm{loc}} \cdot (\beta u_h^n). \tag{3.9}$$

Define $Lu = -\nabla \cdot (\beta u)$. Using the conservation law, we have

$$u_t = -\nabla \cdot (\beta u) = Lu \quad \text{and} \quad u_{tt} = -\nabla \cdot (\beta u_t) = -\nabla \cdot (\beta L u).$$

Using a recursive argument, we have $\partial_t^j u = -\nabla \cdot \left(\beta L^{j-1} u\right)$, for all $j = 0, 1, \cdots, p$. Substituting it into the Taylor expansion (3.8), we have

$$u^{n+1} = u^n - \Delta t \nabla \cdot F(u^n), \quad F(u^n) = \beta \sum_{i=1}^{p} \frac{\Delta t^{i-1}}{i!} L^{i-1} u^n.$$

In the LWDG method, we take the solution to be from V_h and approximate the outer spatial derivative $\nabla \cdot F$ with the DG operator $\nabla^{\mathrm{DG}} \cdot F$ to obtain

$$u_h^{n+1} = u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot F(u_h^n), \quad F(u_h^n) = \beta \sum_{i=0}^{p-1} \frac{\Delta t^i}{(i+1)!} L^i u_h^n, \tag{3.10}$$

after changing the summation index in the definition of F. Finally, note that when β is constant, we have $L u_h^n = -\nabla \cdot (\beta u_h^n) = \Pi(-\nabla \cdot (\beta u_h^n)) = -\nabla^{\mathrm{loc}} \cdot (\beta u_h^n) := L_h u_h^n$. Hence the LWDG scheme (3.10) can be written in the form of (3.9).

Now consider the cRKDG method with explicit time stepping (2.8). Using the linearity of $\nabla^{\mathrm{DG}} \cdot f$, we have

$$u_h^{n+1} = u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot \left(\beta \sum_{i=1}^{p} b_i u_h^{(i)} \right). \tag{3.11}$$

For explicit methods, we can use forward substitution to obtain

$$u_h^{(i)} = \sum_{l=0}^{i-1} d_{il} \left(\Delta t L_h\right)^l u_h^n \tag{3.12}$$

from (2.8a), where d_{il} is defined recursively as $d_{i0} = 1$ and $d_{il} = \sum_{m=l}^{i-1} a_{im} d_{m,l-1}$. Substituting (3.12) into (3.11) and exchanging the summation indices, we have

$$\begin{aligned} u_h^{n+1} &= u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot \left(\beta \sum_{i=1}^{p} \sum_{l=0}^{i-1} b_i d_{il} \left(\Delta t L_h\right)^l u_h^n \right) \\ &= u_h^n - \Delta t \nabla^{\mathrm{DG}} \cdot \left(\beta \sum_{l=0}^{p-1} \left(\sum_{i=l+1}^{p} b_i d_{il} \right) \left(\Delta t L_h\right)^l u_h^n \right). \end{aligned} \tag{3.13}$$

Note that both $-\nabla^{\mathrm{DG}} \cdot (\beta v_h)$ and $L_h v_h$ are approximations to $\partial_t v_h$. To achieve pth order temporal accuracy, we need (3.13) to match the Taylor series after replacing both spatial operators to ∂_t. Hence $\sum_{i=l+1}^{p} b_i d_{il} = 1/(l+1)!$. Substituting it into (3.13) gives (3.9).

Finally, we consider the numerical fluxes of the LWDG method and the cRKDG method. The numerical flux for the LWDG method is $\widehat{F \cdot \nu_{e,K}}(u_h^n)$. For the cRKDG method, from (2.6b), one can see that the numerical flux in the update of u_h^{n+1} is $\sum_{j=1}^{s} b_j \widehat{f \cdot \nu_{e,K}}(u_h^{(j)})$. Therefore, the LWDG method and the cRKDG method are equivalent if we have

$$\widehat{F \cdot \nu_{e,K}}\left(u_h^n\right) = \sum_{j=1}^{s} b_j \widehat{f \cdot \nu_{e,K}}\left(u_h^{(j)}\right). \tag{3.14}$$

∎

Remark 3.4. For the LWDG method studied in [23], the authors adopted the Lax–Friedrichs flux

$$\widehat{F\cdot\nu_{e,K}} = \frac{1}{2}\left(F(u_h^{\text{int}})\cdot\nu_{e,K} + F(u_h^{\text{ext}})\cdot\nu_{e,K} - \alpha_{e,K}\left(u_h^{\text{ext}} - u_h^{\text{int}}\right)\right), \tag{3.15}$$

where $\alpha_{e,K} = \max|\partial_u f\cdot\nu_{e,K}|$. Note that the resulted scheme will be different from the cRKDG method with the Lax–Friedrichs flux. This is because the cRKDG method will include the jump terms of the inner stages $u_h^{(j)}$, $j = 1,\cdots,s$, but the LWDG method will only include the jump terms of u_h^n, which leads to different numerical viscosity used in these numerical fluxes (hence, (3.14) is not satisfied). For the linear scalar case, the Lax–Friedrichs flux for the cRKDG method retrieves the upwind flux, whereas the Lax–Friedrichs flux for the LWDG method (3.15) does not retrieve the upwind flux. However, if we also choose the upwind flux for $\widehat{F\cdot\nu_{e,K}}$ in the LWDG method, the resulting scheme will be identical to the cRKDG method with the upwind flux (or, equivalently, the Lax–Friedrichs flux). From Theorem 3.3 and the discussion here, one can see that the cRKDG method is closely related to the LWDG method in the linear case, which has been carefully investigated in [23].

Remark 3.5. The maximum CFL numbers of the cRKDG method and the LWDG method in [23], in comparison to those of the RKDG method, are given in Table 1.1. The CFL numbers of the LWDG method (with Lax–Friedrichs flux) are about 60% to 67% of those of the corresponding RKDG method. While the cRKDG method achieves the same CFL number as the RKDG method for the second-order case, its CFL number for the third-order case is about 85% of that of the RKDG method. The difference in the CFL numbers of the cRKDG method and the LWDG method is indeed caused by the difference in the numerical fluxes.

Remark 3.6. For nonlinear problems, the LWDG and the cRKDG methods are completely different. The LWDG method requires to precompute the Jacobian and the high-order derivatives of the flux function, which can be very cumbersome in the implementation of the multidimensional systems. However, the cRKDG method does not involve such complications and can be programmed as that of the original RKDG method.

Remark 3.7. A "new" LWDG method was proposed in [19] and then analyzed in [28]. Note this method is different from the LWDG method in [23] in the linear case and is also different from the cRKDG method.

3.5.2. Connections with the ADER-DG schemes with a local predictor. The ADER approach was proposed in [29,30] as a high-order extension of the classical Godunov scheme. Then the methods have been extended in different ways and under both the finite volume and the DG framework. Here we consider one version of the ADER-DG scheme with a local predictor in [12,13].

Multiply the equation (2.1) with a test function $v_h \in V_h$ over a spacetime element $K\times[t^n,t^{n+1}]$. Replacing u with $u_h \in V_h$ yields

$$\int_{t^n}^{t^{n+1}}\int_K \partial_t u_h v_h \mathrm{d}x\mathrm{d}t + \int_{t^n}^{t^{n+1}}\int_K \nabla\cdot f(u_h)v_h\mathrm{d}x\mathrm{d}t = 0.$$

Note $v_h = v_h(x)$ is independent of t. Then we apply Newton-Lebniz rule to ∂_t and

perform integration by parts for $\nabla\cdot$, which yields

$$\int_K \left(u_h^{n+1}-u_h^n\right) v_h \mathrm{d}x - \int_{t^n}^{t^{n+1}} \int_K f(u_h)\cdot\nabla v_h \mathrm{d}x\mathrm{d}t + \int_{t^n}^{t^{n+1}} \int_{\partial K} f(u_h)\cdot\nu_{e,K} v_h \mathrm{d}l\mathrm{d}t = 0.$$

The ADER-DG method with a local predictor can be obtained by replacing u_h in the spatial flux with a local prediction q_h and then replacing the cell interface term with the numerical flux. To be more specific, letting $\mathcal{P}^{k,s}_{K,n} := \mathcal{P}^k(K)\times\mathcal{P}^s\left([t^n,t^{n+1}]\right)$ be the polynomial space on the spacetime element $K\times[t^n,t^{n+1}]$, the ADER-DG method is written as the following: find $u_h\in V_h$ such that

(3.16)
$$\int_K \left(u_h^{n+1}-u_h^n\right) v_h \mathrm{d}x - \int_{t^n}^{t^{n+1}} \int_K f(q_h)\cdot\nabla v_h \mathrm{d}x\mathrm{d}t + \int_{t^n}^{t^{n+1}} \sum_{e\in\partial K}\int_e \widehat{f\cdot\nu_{e,K}}(q_h) v_h \mathrm{d}l\mathrm{d}t = 0$$

holds for all $v_h\in V_h$, where on each spacetime element $K\times[t^n,t^{n+1}]$, $q_h=q_h(x,t)\in\mathcal{P}^{k,s}_{K,n}$ is a local predictor obtained through a local spacetime Galerkin method

$$\int_{t^n}^{t^{n+1}} \int_K \left(\partial_t q_h + \nabla\cdot f(q_h)\right) w_h \mathrm{d}x\mathrm{d}t = 0, \quad \forall w_h = w_h(x,t)\in\mathcal{P}^{k,s}_{K,n}, \tag{3.17}$$

with $q_h(x,t^n)=u_h^n(x)$. Note (3.17) can then be converted into a system of nonlinear equations on each element K and solved locally. By adopting the notations in Section 2, we can write the ADER-DG scheme (3.16) with the local predictor (3.17) into the following strong form:

$$u_h^{n+1} = u_h^n - \int_{t^n}^{t^{n+1}} \nabla^{\mathrm{DG}}\cdot f(q_h)\mathrm{d}t, \tag{3.18a}$$

$$\partial_t q_h = -\,\Pi_{\mathrm{st}}\nabla\cdot f\left(q_h\right), \tag{3.18b}$$

where $u_h^n, u_h^{n+1}\in V_h$, $q_h\in\mathcal{P}^{k,s}_{K,n}$, and Π_{st} is the local L^2 projection to the spacetime polynomial space $\mathcal{P}^{k,s}_{K,n}$.

To see the connection between (3.18) and the cRKDG scheme (2.8), we rewrite both equations in (3.18) in the integral form, which yield

$$u_h^{n+1} = u_h^n - \int_{t^n}^{t^{n+1}} \nabla^{\mathrm{DG}}\cdot f(q_h)\mathrm{d}t \quad\text{and}\quad q_h^{n+1} = q_h^n - \int_{t^n}^{t^{n+1}} \Pi_{\mathrm{st}}\nabla\cdot f\left(q_h\right)\mathrm{d}t. \tag{3.19}$$

Next, we apply a collocation method with s points to integrate (3.19). Suppose this method is associated with the Butcher Tableau (2.5). Then with $q_h^n = u_h^n$, we get (the equations for q_h^{n+1} and $u_h^{(i)}$ are omitted)

$$q_h^{(i)} = u_h^n - \Delta t\sum_{j=1}^s a_{ij}\left(\Pi_{\mathrm{st}}\nabla\cdot f\right)^{(j)}, \quad i=1,2,\cdots,s, \tag{3.20a}$$

$$u_h^{n+1} = u_h^n - \Delta t\sum_{i=1}^s b_i \nabla^{\mathrm{DG}}\cdot f\left(q_h^{(i)}\right), \tag{3.20b}$$

where

$$\left(\Pi_{\mathrm{st}}\nabla\cdot f\right)^{(j)} = \left(\Pi_{\mathrm{st}}\nabla\cdot f\left(\sum_{l=1}^s q_h^{(l)}(x) b_l(t)\right)\right)\Big|_{t=t+c_j\Delta t},$$

and $\{b_l(t)\}_{l=1}^s$ are the Lagrange basis functions associated with the collocation points.

Comparing (3.20) with (2.8), we can see that after applying a collocation type approximation, the ADER-DG method with a local predictor can be written in a similar form as that of the cRKDG method. The main difference between the two methods comes from the definition of the local operator. In the ADER-DG method, the local operator involves the spacetime projection $K \times [t^n, t^{n+1}]$, which couples all s stages in each of the terms of $(\Pi_{\text{st}} \nabla \cdot f)^{(j)}$, hence an implicit approach is needed to solve for $q_h^{(i)}$. While in the cRKDG method, the local operator only involves projection in space and hence is decoupled among different temporal stages – only the term $q_h^{(j)}$ appears in $\nabla^{\text{loc}} \cdot f\left(q_h^{(j)}\right)$.

In light of the above analysis, we have the following theorem.

THEOREM 3.8. *Suppose we replace* Π_{st} *with the spatial only projection* Π *and apply a collocation method with* s *points to integrate* (3.19)*. Then the ADER-DG method with a* $\mathcal{P}_{K,n}^{k,s}$ *local predictor* (3.18) *becomes a cRKDG method associated with a collocation type RK method with* s *stages.*

Moreover, for linear conservation laws with constant coefficients, we have the following equivalence theorem.

THEOREM 3.9. *For linear hyperbolic conservation laws with constant coefficients, the ADER-DG method* (3.16) *with a* $\mathcal{P}_{K,n}^{k,s}$ *local predictor* (3.17) *is equivalent to the cRKDG method with the corresponding RK method being the (implicit) Gauss method.*

Proof. For ease of notation, we will only consider the scalar case $f(u) = \beta u$ with β being a constant vector. But the system case can be proved similarly. Note that

$$\Pi_{\text{st}} \nabla \cdot f(q_h) = \Pi_{\text{st}} \nabla \cdot (\beta q_h) = \nabla \cdot (\beta q_h) = \Pi \nabla \cdot (\beta q_h) = \nabla^{\text{loc}} \cdot f(q_h) \in \mathcal{P}_{K,n}^{k,s}.$$

Therefore, only considering its dependence on t, $\Pi_{\text{st}} \nabla \cdot f(q_h)$ is a temporal polynomial of degree at most s. Using the exactness of Gauss quadrature, the temporal integration in (3.19) can be replaced by an s-stage Gauss collocation method. Hence (3.19) is equivalent to (3.20) with the RK method chosen as the Gauss method. Moreover, note that

$$\begin{aligned}(\Pi_{\text{st}} \nabla \cdot f)^{(j)} &= \left(\nabla \cdot \left(\beta \sum_{l=1}^{s} q_h^{(l)}(x) b_l(t) \right) \right) \Big|_{t=t+c_j \Delta t} \\ &= \sum_{l=1}^{s} \nabla \cdot \left(\beta q_h^{(l)}(x) \right) b_l(t + c_j \Delta t) = \nabla \cdot \left(\beta q_h^{(j)}(x) \right) = \nabla^{\text{loc}} \cdot f\left(q_h^{(j)} \right).\end{aligned}$$

We can replace $(\Pi_{\text{st}} \nabla \cdot f)^{(j)}$ by $\nabla^{\text{loc}} \cdot f\left(q_h^{(j)}\right)$ in (3.20). Then (3.20) becomes the cRKDG method, where the RK method is the s-stage Gauss method. ∎

4. Numerical results. In this section, we present the numerical results of our cRKDG schemes and compare them with those of the original RKDG schemes. We always couple a $\mathcal{P}^k$-DG method with a $(k+1)$th-order RK method. Unless otherwise stated, for the second- and the third-order schemes, we use the SSP-RK time discretization for the original RKDG method, and (2.9) and (2.10) for the cRKDG method, respectively. For the fourth- and the fifth-order schemes, we use the classical forth-order RK and the fifth-order Runge–Kutta–Fehlberg methods for both the original and cRKDG methods, respectively.

4.1. Accuracy tests. In this section, we test the accuracy of cRKDG schemes in different settings and demonstrate the effectiveness of cRKDG schemes for handling nonhomogenous Dirichlet boundary conditions.

4.1.1. One-dimensional tests.

Example 4.1 (Burgers equations). We solve the nonlinear Burgers equation in one dimension with periodic boundary conditions: $\partial_t u + \partial_x \left(u^2/2\right) = 0$, $x \in (-\pi, \pi)$, $u(x,0) = \sin(x)$. We use the Godunov flux and compute to $t = 0.2$ on both uniform and nonuniform meshes with spatial polynomial degrees $k = 1, 2, 3, 4$. We compare the numerical results of RKDG and cRKDG methods. Their L^2 errors and convergence rates are given in Table 4.2. It can be observed that both schemes could achieve the designed optimal order of accuracy with comparable errors on the same meshes. We have also tested our schemes on randomly perturbed meshes, whose numerical results are omitted due to the space limit, and similar convergence rates are observed.

			$k=1$		$k=2$		$k=3$		$k=4$	
		N	L^2 error	order	L^2 error	order	L^2 error	order	L^2 error	order
uniform	RKDG	40	2.7386e-03	-	3.8131e-05	-	6.3822e-07	-	1.0505e-08	-
		80	6.9998e-04	1.97	4.9991e-06	2.95	4.1961e-08	3.93	3.5188e-10	4.90
		160	1.7637e-04	1.99	6.4554e-07	2.95	2.7101e-09	3.95	1.1821e-11	4.90
		320	4.4366e-05	1.99	8.2632e-08	2.97	1.7286e-10	3.97	3.8814e-13	4.93
	cRKDG	40	2.3502e-03	-	3.4537e-05	-	5.9497e-07	-	1.0241e-08	-
		80	5.9868e-04	1.97	4.5379e-06	2.93	3.8796e-08	3.94	3.3912e-10	4.92
		160	1.5073e-04	1.99	5.8341e-07	2.96	2.4857e-09	3.96	1.1335e-11	4.90
		320	3.7882e-05	1.99	7.4902e-08	2.96	1.5801e-10	3.98	3.7040e-13	4.94
nonuniform	RKDG	40	4.2044e-03	-	7.2335e-05	-	1.6005e-06	-	3.5190e-08	-
		80	1.0118e-03	2.06	9.6082e-06	2.91	1.0456e-07	3.94	1.1728e-09	4.91
		160	2.5507e-04	1.99	1.2302e-06	2.97	6.8121e-09	3.94	3.9468e-11	4.89
		320	6.4143e-05	1.99	1.5724e-07	2.97	4.3541e-10	3.97	1.2971e-12	4.93
	cRKDG	40	3.7976e-03	-	6.8122e-05	-	1.5490e-06	-	3.4695e-08	-
		80	9.0218e-04	2.07	8.9388e-06	2.93	9.8699e-08	3.97	1.1449e-09	4.92
		160	2.2598e-04	2.00	1.1464e-06	2.96	6.4244e-09	3.94	3.8321e-11	4.90
		320	5.6822e-05	1.99	1.4645e-07	2.97	4.0891e-10	3.97	1.2563e-12	4.93

Table 4.2: L^2 error of RKDG and cRKDG methods for the one-dimensional Burgers equation on uniform and nonuniform meshes in Example 4.1. The nonuniform meshes are generated by perturbing every other node by $h/3$. $\Delta t = 0.1h$ for $k = 1, 2$ and $\Delta t = 0.05h$ for $k = 3, 4$.

Example 4.2 (Euler equations and validation for Section 3.2.1). In this test, we solve the following nonlinear system of one-dimensional Euler equations $\partial_t u + \partial_x f(u) = 0$ on $(0, 2)$, where $u = (\rho, \rho w, E)^{\mathrm{T}}$, $f(u) = \left(\rho w, \rho w^2 + p, w(E+p)\right)^{\mathrm{T}}$, $E = p/(\gamma - 1) + \rho w^2/2$ with $\gamma = 1.4$. The initial condition is set as $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $w(x,0) = 1$, $p(x,0) = 1$, and the periodic boundary condition is imposed. The exact solution is $\rho(x,t) = 1 + 0.2\sin(\pi(x-t))$, $w(x,t) = 1$, $p(x,t) = 1$. We use the local Lax–Friedrichs flux to compute to $t = 2$.

First, in Table 4.3, we show the numerical error of the cRKDG method with the CFL numbers close to those obtained from the linear stability analysis, which is 0.3 for the second-order case and 0.16 for the third-order case. As anticipated, the cRKDG method can achieve their designed orders of accuracy and remains stable for this nonlinear system when refining the mesh.

Next, we clarify the comments in Section 3.2.1 and explain why the alternations in the cRKDG scheme should be made based on the Butcher form but not the Shu–

Osher form. We implement the schemes (3.2) and (3.3) and observe suboptimal convergence for scheme (3.2) and optimal convergence for scheme (3.3), as shown in Table 4.4. We have also done a similar test to examine the schemes based on the third-order SSP-RK (SSP-RK3) time discretization coupled with $\mathcal{P}^2$ spatial polynomials. A suboptimal convergence rate is observed when we modify the scheme in its Shu–Osher form (similar to (3.1), or equivalently, (3.2)), and the optimal convergence rate is observed when we modify the scheme in its Butcher form (similar to (3.3)).

	(2.9), $k=1$				(2.10), $k=2$			
N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
20	8.6401e-04	-	1.5693e-03	-	4.8592e-05	-	9.6982e-05	-
40	2.1391e-04	2.01	4.3033e-04	1.87	6.3337e-06	2.94	1.3105e-05	2.89
80	5.3413e-05	2.00	1.1235e-04	1.94	7.9905e-07	2.99	1.6569e-06	2.98
160	1.3096e-05	2.03	2.6785e-05	2.07	9.9311e-08	3.01	2.0228e-07	3.03
320	3.3054e-06	1.99	7.0220e-06	1.93	1.2477e-08	2.99	2.5741e-08	2.97
640	8.3321e-07	1.99	1.8091e-06	1.96	1.5656e-09	2.99	3.2598e-09	2.98
1280	2.0304e-07	2.04	4.1002e-07	2.14	1.9242e-10	3.02	3.8229e-10	3.09
2560	5.1018e-08	1.99	1.0552e-07	1.96	2.4061e-11	3.00	4.7861e-11	3.00

Table 4.3: L^2 and L^∞ error for one-dimensional Euler equations in Example 4.2. CFL is 0.3 for $k=1$ and 0.16 for $k=2$.

	Wrong implementation with Shu–Osher form				Correct implementation with Butcher form			
	Scheme (3.2), $k=1$		SSP-RK3, $k=2$		Scheme (3.3), $k=1$		SSP-RK3, $k=2$	
N	L^2 error	order	L^2 error	order	L^2 error	order	L^2 error	order
20	2.2503e-002	-	8.42142e-04	-	8.3248e-04	-	4.7661e-05	-
40	1.1013e-002	1.03	2.4793e-04	1.93	1.9946e-04	2.06	6.1420e-06	2.96
80	5.4484e-003	1.03	6.3363e-05	1.97	4.9608e-05	2.01	7.7938e-07	2.98

Table 4.4: L^2 error for one-dimensional Euler equations in Example 4.2. CFL is 0.1.

Example 4.3 (Boundary error). In this example, we test the problem [35, Section 4] to examine the possible accuracy degeneration due to the nonhomogeneous boundary condition. We use the $\mathcal{P}^2$-DG method with upwind flux and the third-order RK scheme to solve $\partial_t u + \partial_x u = 0$ on domain $(0, 4\pi)$. The initial condition is set as $u(x,0) = \sin(x)$ and the exact solution is given by $u(x,t) = \sin(x-t)$. Both the periodic and the inflow boundary conditions are considered in our test.

We set $\Delta t = 0.16h$ and compute to $t = 20$. Numerical errors and convergence rates are listed in Table 4.5. It could be seen that the original RKDG method achieves the optimal convergence rate for the periodic boundary condition but a degenerated rate for the inflow boundary condition. While in contrast, the cRKDG method is able to achieve optimal convergence rates for both types of boundary conditions.

4.1.2. Two-dimensional tests. The triangular meshes in this section are generated by taking a cross in each cell of the $N \times N$ uniform square meshes.

Example 4.4 (Euler equations in two dimensions). We solve the nonlinear Euler equations in two dimensions with the periodic boundary condition: $\partial_t u + \partial_x f(u) + \partial_y g(u) = 0$, where $u = (\rho, \rho w, \rho v, E)^T$, $f(u) = (\rho w, \rho w^2 + p, \rho wv, w(E+p))^T$, $g(u) = (\rho v, \rho wv, \rho v^2 + p, v(E+p))^T$, $E = p/(\gamma-1) + \rho(w^2+v^2)/2$ with $\gamma = 1.4$. The initial condition is set as $\rho(x,y,0) = 1 + 0.2\sin(\pi(x+y))$, $w(x,y,0) = 0.7$, $v(x,y,0) = 0.3$, $p(x,y,0) = 1$. The exact solution is $\rho(x,y,t) = 1+0.2\sin(\pi(x+y-(w+v)t))$, $w = 0.7$,

		periodic boundary				inflow boundary			
	N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
RKDG	40	4.5605E-04	-	4.0696E-04	-	3.8572E-04	-	3.8889E-04	-
	80	5.5726E-05	3.03	5.1832E-05	2.97	4.8763E-05	2.98	4.8941E-05	2.99
	160	6.9243E-06	3.01	6.5389E-06	2.99	6.3065E-06	2.95	7.2957E-06	2.75
	320	8.6412E-07	3.00	8.2105E-07	2.99	8.4142E-07	2.91	1.7155E-06	2.09
	640	1.0796E-07	3.00	1.0286E-07	3.00	1.1738E-07	2.84	4.1662E-07	2.04
	1280	1.3493E-08	3.00	1.2873E-08	3.00	1.7331E-08	2.76	1.0270E-07	2.02
cRKDG	40	1.7656E-03	-	7.3382E-04	-	7.3651E-04	-	4.6481E-04	-
	80	2.2030E-04	3.00	9.0392E-05	3.02	9.0921E-05	3.02	5.9025E-05	2.98
	160	2.7536E-05	3.00	1.1209E-05	3.01	1.1296E-05	3.01	7.4400E-06	2.99
	320	3.4428E-06	3.00	1.3953E-06	3.01	1.4079E-06	3.00	9.3401E-07	2.99
	640	4.3036E-07	3.00	1.7415E-07	3.00	1.7576E-07	3.00	1.1701E-07	3.00
	1280	5.3797E-08	3.00	2.1783E-08	3.00	2.1957E-08	3.00	1.4643E-08	3.00

Table 4.5: Error table for the one-dimensional linear advection equation with periodic and inflow boundary conditions in Example 4.3. For the periodic boundary, $u(0,t) = u(4\pi,t)$; for the inflow boundary, $u(0,t) = \sin(-t)$. $k = 2$ and $\Delta t = 0.16h$.

$v = 0.3$, $p = 1$. We use the local Lax–Friedrichs flux and compute the solution up to $t = 2$. Here the CFL numbers are taken as 0.2 and 0.12 for $\mathcal{P}^1$ and $\mathcal{P}^2$ cRKDG methods, and as 0.3 and 0.18 for $\mathcal{P}^1$ and $\mathcal{P}^2$ RKDG methods. We list numerical results in Table 4.6. We can observe that both RKDG and cRKDG schemes achieve their expected order of optimal accuracy with comparable numerical errors on both triangular and rectangular meshes.

		RKDG				cRKDG			
		$k=1$		$k=2$		$k=1$		$k=2$	
	N	L^2 error	order	L^2 error	order	L^2 error	order	L^2 error	order
triangular	20	4.5685e-04	-	5.1570e-05	-	4.4847e-04	-	4.9510e-05	-
	40	1.1073e-04	2.04	6.1085e-06	3.08	1.0842e-04	2.05	5.8449e-06	3.08
	80	2.7508e-05	2.01	7.7595e-07	2.98	2.6859e-05	2.01	7.4251e-07	2.98
	160	6.8934e-06	2.00	9.6981e-08	3.00	6.6652e-06	2.01	9.2728e-08	3.00
rectangular	20	2.4343e-03	-	1.1101e-04	-	2.4662e-03	-	1.1300e-04	-
	40	4.2736e-04	2.51	1.3885e-05	3.00	4.2767e-04	2.53	1.4213e-05	2.99
	80	9.0669e-05	2.24	1.7297e-06	3.00	8.9727e-05	2.25	1.7737e-06	3.00
	160	2.1445e-05	2.08	2.1586e-07	3.00	2.1140e-05	2.09	2.2173e-07	3.00

Table 4.6: L^2 error for two-dimensional Euler equations with the periodic boundary condition on triangular and rectangular meshes in Example 4.4.

Example 4.5 (Boundary error). Consider the linear advection equation in two dimensions $\partial_t u + \partial_x u + \partial_y u = 0$, $(x,y) \in [-1,1]\times[-1,1]$, $u(x,y,0) = \sin(\pi x)\sin(\pi y)$, The exact solution is $u(x,y,t) = \sin(\pi(x-t))\sin(\pi(y-t))$. We use the upwind flux and compute the solution up to $t = 0.4$ with $\mathcal{P}^3$ elements. The numerical results with both periodic and inflow boundary conditions are given in Table 4.7. It can be observed that, on the same set of triangular meshes, both schemes achieve their designed order of accuracy with comparable numerical error under the periodic condition. For the inflow boundary condition, the RKDG method becomes suboptimal while the cRKDG method remains optimal.

4.2. Tests with discontinuous solutions. We now test the cRKDG method for problems with discontinuous solutions. Only cell averages of the solutions are plotted. For one-dimensional problems, we apply the TVB minmod limiter for systems

		periodic boundary				inflow boundary			
	N	L^2 error	order	L^∞ error	order	L^2 error	order	L^∞ error	order
RKDG	20	1.6470e-06	-	6.1638e-06	-	2.4337e-06	-	2.7118e-05	-
	40	1.0807e-07	3.93	4.1438e-07	3.89	2.8514e-07	3.09	6.6092e-06	2.04
	80	6.7371e-09	4.00	2.5981e-08	4.00	4.5845e-08	2.64	1.6418e-06	2.01
	160	4.2131e-10	4.00	1.6135e-09	4.01	7.9968e-09	2.52	4.0979e-07	2.00
cRKDG	20	1.4623e-06	-	4.4906e-06	-	1.7295e-06	-	4.9173e-06	-
	40	8.9533e-08	4.03	2.8503e-07	3.98	1.0770e-07	4.01	3.0924e-07	3.99
	80	5.6015e-09	4.00	1.7533e-08	4.02	6.7326e-09	4.00	1.9284e-08	4.00
	160	3.5131e-10	3.99	1.0718e-09	4.03	4.2143e-10	4.00	1.2036e-09	4.00

Table 4.7: Error table for the two-dimensional linear advection equation on triangular meshes with periodic and inflow boundary conditions in Example 4.5. $k = 3$. $\Delta t = h/20$ for the RKDG scheme and $\Delta t = h/30$ for the cRKDG scheme.

in [7] to identify troubled cells and the WENO limiter in [23] for reconstruction. For two-dimensional problems, we apply the standard TVB minmod limiters in [10] to identify troubled cells and reconstruct polynomials, with a suitable parameter M to be specified for each problem. For the RKDG method, the limiter is applied in every inner stage. For the cRKDG method, the limiter is only applied once at the final stage for each time step.

Since the cRKDG method is equivalent to the LWDG method for linear problems, they share the same CFL limit. As has been tested in [23] with the von Neumann analysis, this CFL limit will be slightly more restrictive compared with the original RKDG method. In our numerical tests, the CFL numbers are taken as 0.3 and 0.16 for $\mathcal{P}^1$ and $\mathcal{P}^2$ cRKDG methods, and as 0.3 and 0.18 for $\mathcal{P}^1$ and $\mathcal{P}^2$ RKDG methods, respectively, unless otherwise noted.

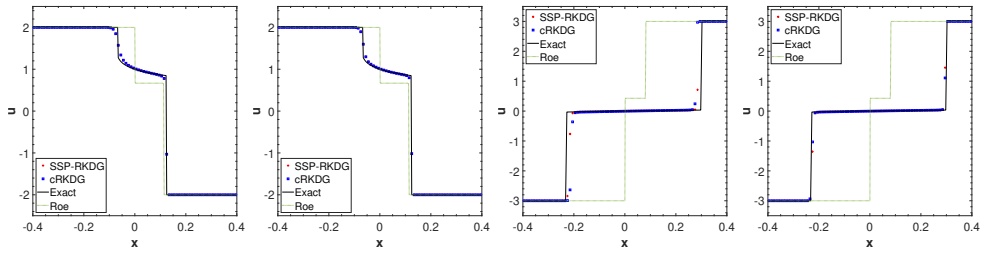
4.2.1. One-dimensional tests.

Example 4.6 (Buckley–Leverett equation). We solve two Riemann problems associated with the Buckley–Leverett equation $\partial_t u + \partial_x \left(4u^2/(4u^2 + (1-u)^2)\right) = 0$. The initial condition is set as $u(x,0) = u_l$ for $x < 0$ and $u(x,0) = u_r$ for $x \geq 0$, where we have $u_l = 2$ and $u_r = -2$ in the first test and $u_l = -3$ and $u_r = 3$ in the second test. We set $k = 1, 2$ and compute to $t = 1$ with 80 mesh cells. The Godunov flux and WENO limiter with TVB constant $M = 1$ are employed in the simulation. We observe that the cRKDG method converges to the correct entropy solutions and its numerical results are in good agreement with the original RKDG method. We have also plotted the solution by the first-order Roe scheme in green dots. In contrast, the solution by the Roe scheme converges to a non-entropy solution.

Example 4.7 (Sod problem). In this test, we solve a Riemann problem for the one-dimensional Euler equations given in Example 4.2. The initial condition is set as

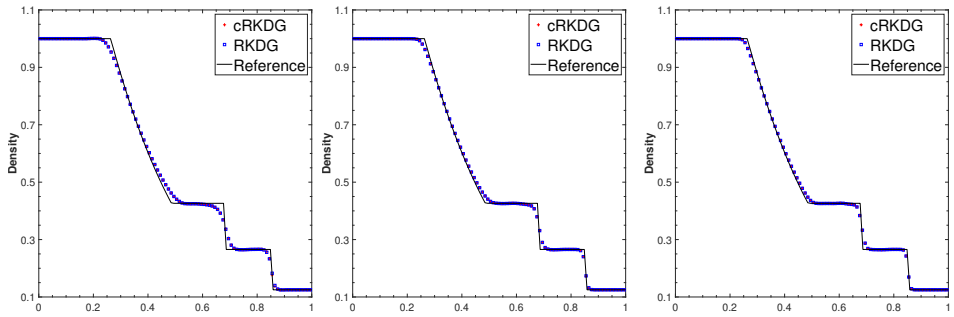
$$\rho(x,0) = \begin{cases} 1.0, & x < 0.5 \\ 0.125, & x \geq 0.5 \end{cases}, \quad \rho u(x,0) = 0, \quad E(x,0) = \frac{1}{\gamma - 1} \begin{cases} 1, & x < 0.5 \\ 0.1, & x \geq 0.5 \end{cases},$$

where $\gamma = 1.4$. We compute to $t = 0.2$ with $N = 100$ elements. We use the local Lax–Friedrichs flux, WENO limiter and TVB constant $M = 1$. The solution profiles are given in Figure 4.3, from which we can observe that the cRKDG method performs well in capturing the shock and contact discontinuity, and its numerical solution matches the RKDG solution and the exact solution.



(a) $k = 1$, test 1 (b) $k = 2$, test 1 (c) $k = 1$, test 2 (d) $k = 2$, test 2

Fig. 4.2: Solution profiles for two Riemann problems of the Buckley–Leverett equation in Example 4.6. $N = 80$ and $M = 1$.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Fig. 4.3: Solution profiles for the Sod problem in Example 4.7. $N = 100$ and $M = 1$.

Example 4.8 (Interacting blast waves). We consider the interacting blast waves with Euler equations using the following initial condition

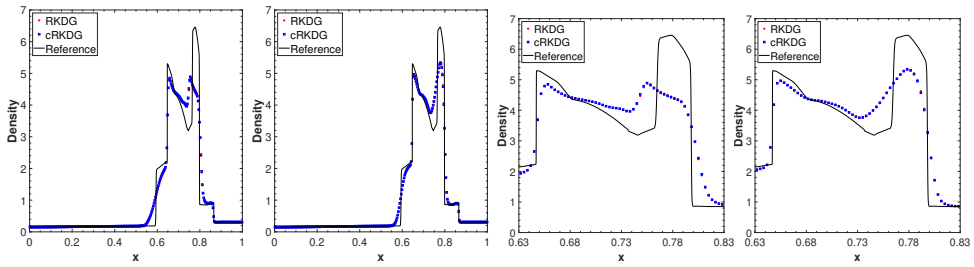
$$(\rho, \mu, p) = \begin{cases} (1, 0, 1000), & x \le 0.1, \\ (1, 0, 0.01), & 0.1 < x \le 0.9, \\ (1, 0, 100), & x > 0.9. \end{cases}$$

Reflective boundaries are imposed both at $x = 0$ and $x = 1$. We use the local Lax–Friedrichs flux, WENO limiter and TVB constant $M = 200$, and compute the solution to $t = 0.038$. Numerical results are shown in Figure 4.4. The numerical density ρ is plotted against the reference solution which is a converged solution computed by the fifth-order finite difference WENO scheme on a much refined mesh. It is observed that numerical solutions obtained from RKDG and cRKDG methods are very close.

Example 4.9 (Shu–Osher problem). We consider the Shu–Osher problem describing a Mach 3 shock interacting with sine waves in density. This is a problem of shock interaction with entropy waves and thus contains both shocks and complex smooth region structures [27]. The initial condition is set as

$$(\rho, \mu, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & x < -4, \\ (1 + 0.2\sin(5x), 0, 1), & x \ge -4. \end{cases}$$

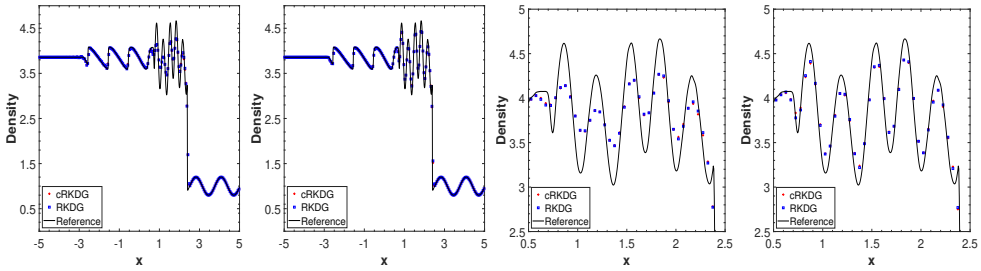
The numerical density ρ is plotted at $t = 1.8$ against the reference solution which is computed by the fifth-order finite difference WENO scheme. In Figure 4.5, we plot the densities by cRKDG and RKDG methods with the local Lax–Friedrichs flux,



(a) $k = 1$ (b) $k = 2$ (c) $k = 1$ (Zoomed-in) (d) $k = 2$ (Zoomed-in)

Fig. 4.4: Solution profiles for the blast wave problem in Example 4.8. $N = 300$ and $M = 200$.

WENO limiter and TVB constant $M = 200$. In addition, we also show a zoomed-in view of the solution at $x \in [0.5, 2.5]$ in Figures 4.5c and 4.5d.



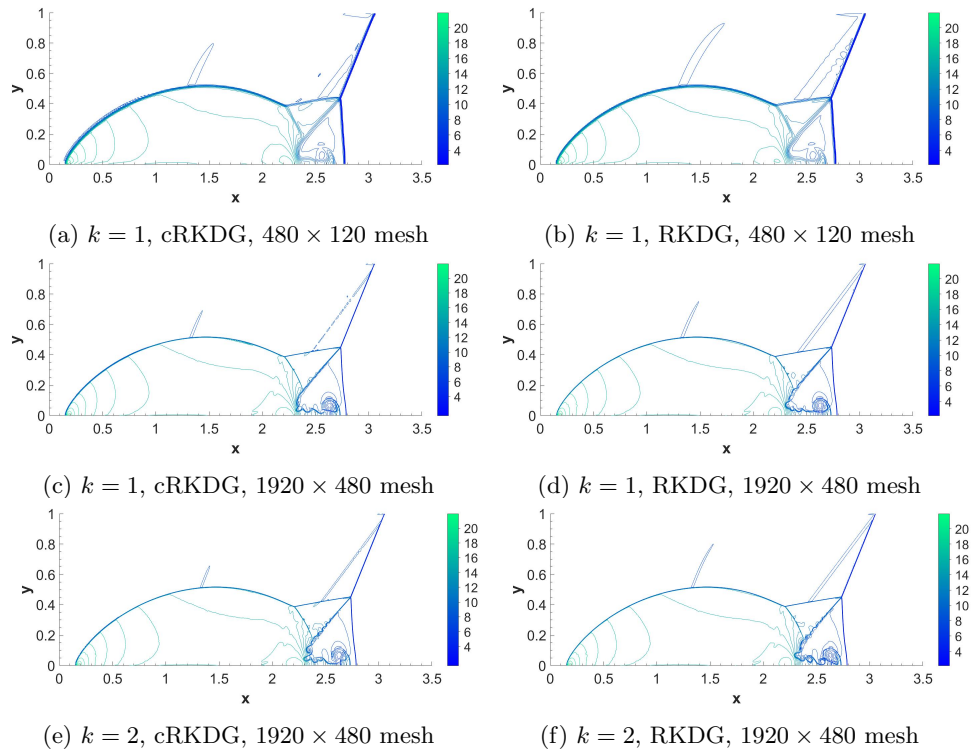
(a) $k = 1$ (b) $k = 2$ (c) $k = 1$ (Zoomed-in) (d) $k = 2$ (Zoomed-in)

Fig. 4.5: Solution profiles for the Shu–Osher problem in Example 4.9 at $t = 1.8$. $M = 200$ and $N = 200$.

4.2.2. Two-dimensional tests.

Example 4.10 (Double Mach reflection). This problem is originally studied in [33] and describes reflections of planar shocks in the air from wedges. The computational domain is $[0, 4] \times [0, 1]$, and the reflecting wall lies at the bottom boundary, starting from $x = 1/6$. Therefore, for the bottom boundary, the exact post-shock condition is imposed for the region from $x = 0$ to $x = 1/6$, while a reflective boundary condition is applied to the rest. At $t = 0$, a right-moving Mach 10 shock is positioned at $x = 1/6$, $y = 0$ and makes a $60°$ angle with the x-axis. At the top boundary, the flow values are set to describe the exact motion of the Mach 10 shock. The boundary conditions at the left and the right are inflow and outflow respectively. We compute the solution up to $t = 0.2$ and use the TVB limiter with $M = 50$. To save space, we only present the simulation results with 480×120 mesh cells for $k = 1$ and 1960×480 mesh cells for $k = 1, 2$ in Figure 4.6. The corresponding zoomed-in figures around the double Mach stem are given in Figure 4.7. For this problem, the resolutions of cRKDG and RKDG methods are comparable for the same order of accuracy and the same meshes.

Example 4.11 (Forward facing step). This is another classical test studied in [33]. In this test, a Mach 3 uniform flow travels to the right and enters a wind tunnel (of 1 length unit wide and 3 length units long), with the step of 0.2 length units high located 0.6 length units from the left-hand end of the tunnel. Reflective boundary conditions are applied along the wall of the tunnel, while inflow/outflow boundary



(a) $k = 1$, cRKDG, 480×120 mesh (b) $k = 1$, RKDG, 480×120 mesh

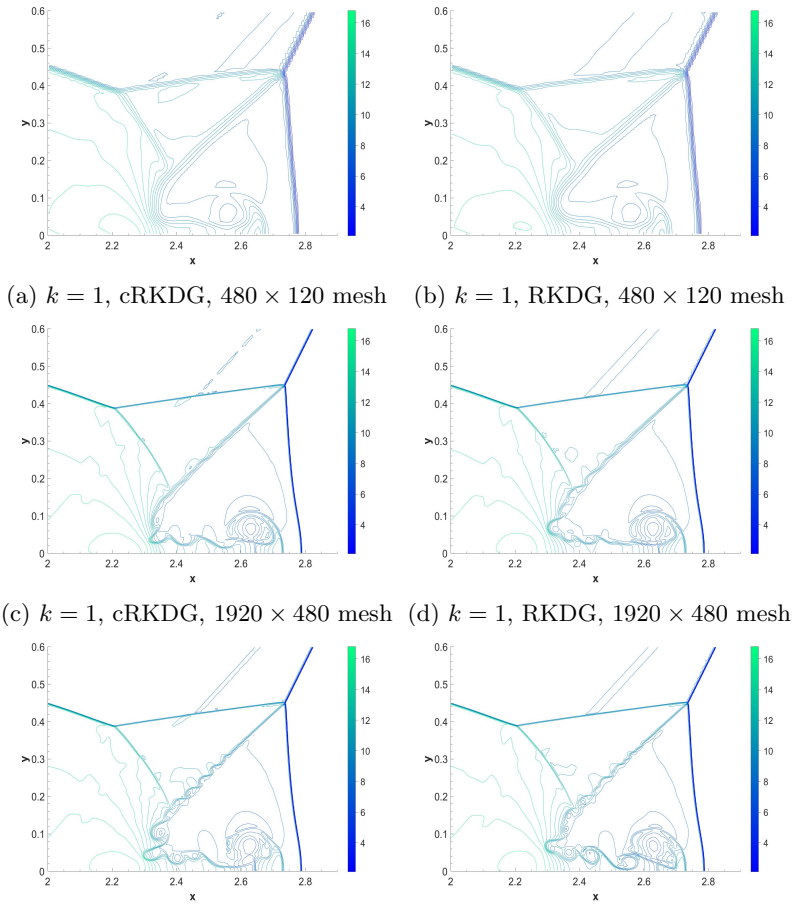
(c) $k = 1$, cRKDG, 1920×480 mesh (d) $k = 1$, RKDG, 1920×480 mesh

(e) $k = 2$, cRKDG, 1920×480 mesh (f) $k = 2$, RKDG, 1920×480 mesh

Fig. 4.6: Solution profiles for the double Mach problem in Example 4.10 at $t = 0.2$ with $M = 50$. 30 equally spaced density contours from 1.3695 to 22.682 are displaced.

conditions are applied at the entrance/exit. At the corner of the step, a singularity is present. Unlike in [33], we do not modify our schemes or refine the mesh near the corner in order to test the performance of our schemes in handling such singularity. We compute the solution up to $t = 4$ and utilize the TVB limiter with a TVB constant $M = 50$. Due to the space limitation, we only present the simulation results with 240×80 mesh cells for $k = 1$ and 960×320 mesh cells for $k = 1, 2$ in Figure 4.8. For this problem, the resolutions of cRKDG and RKDG methods are comparable for the same order of accuracy and mesh.

5. Conclusions and future work. In this paper, we present a novel class of RKDG methods with compact stencils for solving the hyperbolic conservation laws. Our main idea is to replace the DG operator in the inner temporal stages of the fully discrete RKDG scheme by a local derivative operator. We prove a Lax–Wendroff type theorem which guarantees its convergence to the weak solution. Numerically, we observe the new method achieves the optimal convergence rate and does not suffer from the order degeneracy when the Dirichelet type inflow boundary condition is imposed. Moreover, the connections of the new method with the LWDG and ADER-DG methods are established. This is the first paper of a few of our future works, which include the rigorous stability and error analysis with the energy method, extensions to implicit time stepping and convection-dominated problems, and the design of structure-preserving schemes based on the cRKDG methods, etc..



(a) $k = 1$, cRKDG, 480×120 mesh (b) $k = 1$, RKDG, 480×120 mesh

(c) $k = 1$, cRKDG, 1920×480 mesh (d) $k = 1$, RKDG, 1920×480 mesh

(e) $k = 2$, cRKDG, 1920×480 mesh (f) $k = 2$, RKDG, 1920×480 mesh

Fig. 4.7: Zoomed-in solutions for the double Mach problem in Example 4.10 at $t = 0.2$ with $M = 50$. 30 equally spaced density contours from 1.3695 to 22.682 are displaced.

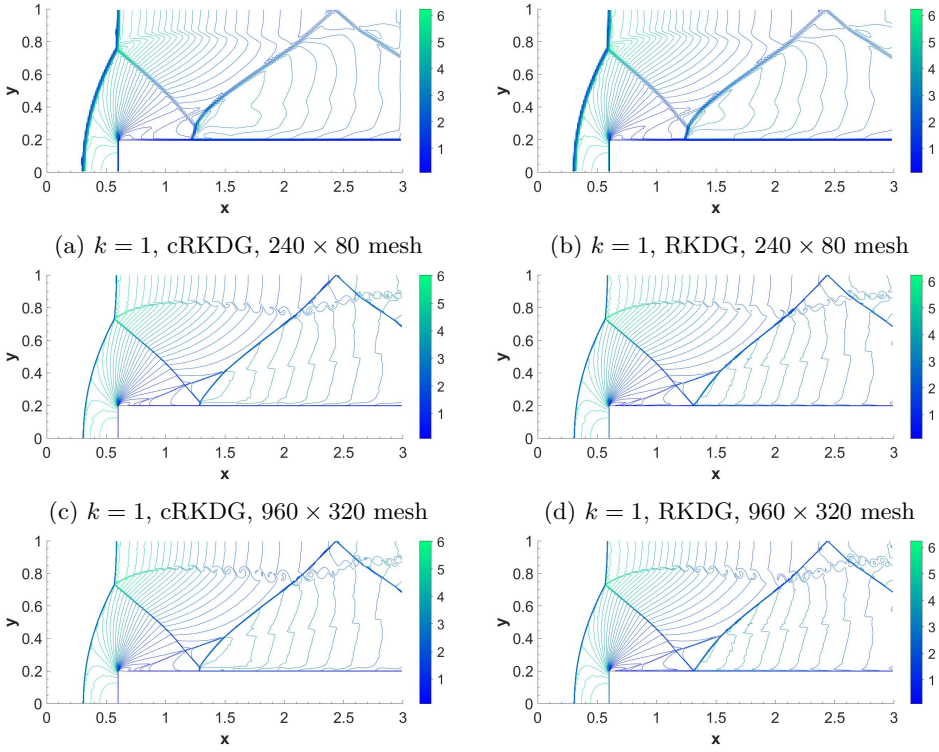
Acknowledgments. Z. Sun was partially supported by the NSF grant DMS-2208391. Y. Xing was partially supported by the NSF grant DMS-1753581 and DMS-2309590. The authors would also like to thank Professor Jianxian Qiu at Xiamen University for sharing the code for 2D Euler tests and providing helpful comments.

Appendix A. Proof of Theorem 3.1. Recall that we denote by $u_h^{(1)} = u_h^n$. To prove a Lax–Wendroff convergence theorem, we will use the following result simplified from [26, Theorems 2.3 and 3.2].

THEOREM A.1 (Shi and Shu, 2018. [26]). *Let f be Lipschitz continuous and f', f'' be uniformly bounded in L^∞. Consider a numerical scheme in d-dimensional space that yields (3.4). For any mesh cell K, its edge $e \in \partial K$, and its neighboring cell K^{ext}, suppose $g_{e,K}$ satisfies the following properties on $B_K = K \cup K^{\text{ext}}$.*

1. *Consistency: if $u_h^n(x) \equiv u$ is a constant, then $g_{e,K}(u_h^n) = |e| f(u) \cdot \nu_{e,K}$.*
2. *Boundedness: $|g_{e,K}(u_h^n) - g_{e,K}(v_h^n)| \le C \|u_h^n - v_h^n\|_{L^\infty(B_K)} \cdot h^{d-1}$.*
3. *Anti-symmetry: $g_{e,K}(u_h^n) + g_{e,K^{\text{ext}}}(u_h^n) = 0$, for $e = K \cap K^{\text{ext}}$.*

If u_h^n converges boundedly almost everywhere to a function u as $\Delta t, h \to 0$, then u is



(a) $k = 1$, cRKDG, 240×80 mesh (b) $k = 1$, RKDG, 240×80 mesh

(c) $k = 1$, cRKDG, 960×320 mesh (d) $k = 1$, RKDG, 960×320 mesh

(e) $k = 2$, cRKDG, 960×320 mesh (f) $k = 2$, RKDG, 960×320 mesh

Fig. 4.8: Solution profiles for the forward step problem in Example 4.11 at $t = 4$ with $M = 50$. 30 equally spaced density contours from 0.090388 to 6.2365 are displaced.

a weak solution to the conservation law saftisfying (3.6).

Hence to analyze the convergence of the cRKDG method, it suffices to verify that $g_{e,K}$ defined in (3.5) does satisfy three properties in Theorem A.1.

Lemma A.1. Under assumptions in Theorem 3.1, the combined flux $g_{e,K}$ defined in (3.5) satisfies the three properties in Theorem A.1.

Once Lemma A.1 is proved, Theorem 3.1 follows as a direct consequence of Theorem A.1. The rest of the section is dedicated to the proof of Lemma A.1, especially on the boundedness of $g_{e,K}$.

Lemma A.2. Let ρ be a L^2 and L^∞ function. Then $\|\Pi\rho\|_{L^\infty(K)} \le C\|\rho\|_{L^\infty(K)}$.

Proof. This lemma can be proved by selecting an orthonormal basis of $\mathcal{P}^k(K)$ and expand $\Pi\rho$ under this basis. Details are omitted. ∎

Lemma A.3. For any $u_h, v_h \in V_h$ with $\|u_h\|_{L^\infty}, \|v_h\|_{L^\infty} \le C$, we have

$$\|\Pi\nabla \cdot (f(u_h) - f(v_h))\|_{L^\infty(B_K)} \le \frac{C}{h}\|u_h - v_h\|_{L^\infty(B_K)}. \tag{A.1}$$

Proof. Applying Lemma A.2 and after some algebraic manipulations, we have

$$
\begin{aligned}
&\|\Pi\nabla\cdot\left(f(u_h)-f(v_h)\right)\|_{L^\infty(B_K)}\\
\leq& C\|\nabla\cdot\left(f(u_h)-f(v_h)\right)\|_{L^\infty(B_K)}\\
=& C\|f'(u_h)\cdot\nabla u_h - f'(v_h)\cdot\nabla v_h\|_{L^\infty(B_K)}\\
=& C\|f'(u_h)\cdot\nabla u_h - f'(v_h)\cdot\nabla u_h + f'(v_h)\cdot\nabla u_h - f'(v_h)\cdot\nabla v_h\|_{L^\infty(B_K)}\\
\leq& C\|\left(f'(u_h)-f'(v_h)\right)\cdot\nabla u_h\|_{L^\infty(B_K)} + C\|f'(v_h)\cdot\nabla\left(u_h-v_h\right)\|_{L^\infty(B_K)}\\
\leq& C\|f''\|_{L^\infty}\|u_h-v_h\|_{L^\infty(B_K)}\|\nabla u_h\|_{L^\infty(B_K)} + C\|f'\|_{L^\infty}\|\nabla(u_h-v_h)\|_{L^\infty(B_K)}.
\end{aligned}
\tag{A.2}
$$

With the inverse estimate, we have $\|\nabla u_h\|_{L^\infty(B_K)} \leq Ch^{-1}\|u_h\|_{L^\infty(B_K)}$ and $\|\nabla(u_h - v_h)\|_{L^\infty(B_K)} \leq Ch^{-1}\|u_h - v_h\|_{L^\infty(B_K)}$. Recall that we assumed f', f'', and u_h are bounded in L^∞. (A.1) can be obtained after substituting these estimates into (A.2).∎

Lemma A.4. $\|u_h^{(i)} - v_h^{(i)}\|_{L^\infty(B_K)} \leq C\|u_h^n - v_h^n\|_{L^\infty(B_K)}$, for all $1 \leq i \leq s$.

Proof. We prove the lemma by induction. For $i = 1$, by definition we have $u_h^{(1)} = u_h^n$ and $v_h^{(1)} = v_h^n$. The inequality is true with $C = 1$ for all $u_h^n, v_h^n \in V_h$.

Now we assume that the inequality is true with $i \leq m$.

First, we want to show that the induction hypothesis implies

$$\|u_h^{(i)}\|_{L^\infty(B_K)} \leq C\|u_h^n\|_{L^\infty(B_K)} \quad \forall 1 \leq i \leq m.$$

Indeed, note that when $v_h^n \equiv 0$ on B_K, $f(v_h^n) \equiv 0$ is a constant. Hence $v_h^{(2)} = 0 - a_{21}\Delta t\nabla^{\text{loc}}\cdot 0 = 0$. Similarly, we have $v_h^{(i)} \equiv 0$ for all $1 \leq i \leq m$. By the induction hypothesis, we can see that for all $1 \leq i \leq m$,

$$\|u_h^{(i)}\|_{L^\infty(B_K)} = \|u_h^{(i)} - v_h^{(i)}\|_{L^\infty(B_K)} \leq C\|u_h^n - v_h^n\|_{L^\infty(B_K)} = C\|u_h^n\|_{L^\infty(B_K)}.$$

Then we prove the lemma is true for $i = m + 1$. It can be seen that

$$
\begin{aligned}
&\|u_h^{(m+1)} - v_h^{(m+1)}\|_{L^\infty(B_K)}\\
=&\left\|\left(u_h^n - \Delta t\sum_{j=1}^m a_{ij}\Pi\nabla\cdot f\left(u_h^{(j)}\right)\right) - \left(v_h^n - \Delta t\sum_{j=1}^m a_{ij}\Pi\nabla\cdot f\left(v_h^{(j)}\right)\right)\right\|_{L^\infty(B_K)}\\
\leq&\|u_h^n - v_h^n\|_{L^\infty(B_K)} + \Delta t\sum_{j=1}^m |a_{ij}|\left\|\Pi\nabla\cdot\left(f\left(u_h^{(j)}\right) - f\left(v_h^{(j)}\right)\right)\right\|_{L^\infty(B_K)}.
\end{aligned}
$$

According to the first part of the proof, $\left\|u_h^{(j)}\right\|_{L^\infty} \leq C\|u_h^n\|_{L^\infty} \leq C$ and $\left\|v_h^{(j)}\right\|_{L^\infty} \leq C\|v_h^n\|_{L^\infty} \leq C$ are bounded. Hence with Lemma A.3, it yields

$$\|u_h^{(m+1)} - v_h^{(m+1)}\|_{L^\infty(B_K)} \leq \|u_h^n - v_h^n\|_{L^\infty(B_K)} + \frac{C\Delta t}{h}\sum_{j=1}^m |a_{ij}|\left\|u_h^{(j)} - v_h^{(j)}\right\|_{L^\infty(B_K)}.$$

One can then prove the lemma after using the CFL condition $\Delta t/h \leq C$ and the induction hypothesis $\|u_h^{(j)} - v_h^{(j)}\|_{L^\infty(B_K)} \leq C\|u_h^n - v_h^n\|_{L^\infty(B_K)}$ for all $1 \leq j \leq m$. ∎

Proof of Lemma A.1. The consistency of $g_{e,K}$ can be obtained from the consistency of $\widehat{f}$ and the consistency of the RK method $\sum_{i=1}^s b_i = 1$. The anti-symmetry

of $g_{e,K}$ can be obtained from the anti-symmetry of $\widehat{f}$. The key is to show the boundedness of $g_{e,K}$ as follows.

$$
\begin{aligned}
|g_{e,K}(u_h) - g_{e,K}(v_h)| =& \left| \int_e \left(\sum_{i=1}^{s} b_i \widehat{f \cdot \nu_{e,K}} \left(u_h^{(i)} \right) \right) \mathrm{d}l - \int_e \left(\sum_{i=1}^{s} b_i \widehat{f \cdot \nu_{e,K}} \left(v_h^{(i)} \right) \right) \mathrm{d}l \right| \\
\leq& \sum_{i=1}^{s-1} \int_e |b_i| \left| \widehat{f \cdot \nu_{e,K}} \left(u_h^{(i)} \right) - \widehat{f \cdot \nu_{e,K}} \left(v_h^{(i)} \right) \right| \mathrm{d}l \\
\leq& C \sum_{i=1}^{s-1} |b_i||e| \left\| u_h^{(i)} - v_h^{(i)} \right\|_{L^\infty} \leq C \left\| u_h^n - v_h^n \right\|_{L^\infty} \cdot h^{d-1}.
\end{aligned}
$$

Here we have used the Lipschitz continuity of $\widehat{f \cdot \nu_{e,K}}$ in the second last inequality and Lemma A.4 in the last inequality. ∎

REFERENCES

[1] F. Bassi, A. Crivellini, S. Rebay, and M. Savini, *Discontinuous Galerkin solution of the reynolds-averaged Navier–Stokes and k–ω turbulence model equations*, Comput. & Fluids, 34 (2005), pp. 507–540.

[2] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini, *A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows*, in Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, 1997, pp. 99–109.

[3] J. L. Bona, H. Chen, O. A. Karakashian, and Y. Xing, *Conservative, discontinuous Galerkin-methods for the Generalized Korteweg-de Vries equation*, Math. Comp., 82 (2013), pp. 1401–1432.

[4] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don, *The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error*, SIAM J. Sci. Comput., 16 (1995), pp. 1241–1252.

[5] Y. Cheng and C.-W. Shu, *A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives*, Math. Comp., 77 (2008), pp. 699–730.

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

[7] 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., 84 (1989), pp. 90–113.

[8] 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 (1989), pp. 411–435.

[9] B. Cockburn and C.-W. Shu, *The Runge–Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws*, ESAIM Math. Model. Numer. Anal., 25 (1991), pp. 337–361.

[10] B. Cockburn and C.-W. Shu, *The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, J. Comput. Phys., 141 (1998), pp. 199–224.

[11] B. Cockburn and C.-W. Shu, *Runge–Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comput., 16 (2001), pp. 173–261.

[12] M. Dumbser, D. S. Balsara, E. F. Toro, and C.-D. Munz, *A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes*, J. Comput. Phys., 227 (2008), pp. 8209–8253.

[13] M. Dumbser, C. Enaux, and E. F. Toro, *Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws*, J. Comput. Phys., 227 (2008), pp. 3971–4001.

[14] M. Dumbser and C.-D. Munz, *Building blocks for arbitrary high order discontinuous Galerkin schemes*, J. Sci. Comput., 27 (2006), pp. 215–230.

[15] E. Gaburro, P. Öffner, M. Ricchiuto, and D. Torlo, *High order entropy preserving ADER-DG schemes*, Appl. Math. Comput., 440 (2023), p. 127644.

[16] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu, *Strong stability preserving Runge-Kutta and multistep time discretizations*, World Scientific, 2011.
[17] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
[18] Z. J. Grant, *Perturbed Runge–Kutta methods for mixed precision applications*, J. Sci. Comput., 92 (2022), p. 6.
[19] W. Guo, J.-M. Qiu, and J. Qiu, *A new Lax–Wendroff discontinuous Galerkin method with superconvergence*, J. Sci. Comput., 65 (2015), pp. 299–326.
[20] H. Luo, L. Luo, R. Nourgaliev, V. A. Mousseau, and N. Dinh, *A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids*, J. Comput. Phys., 229 (2010), pp. 6961–6978.
[21] J. Peraire and P.-O. Persson, *The compact discontinuous Galerkin (CDG) method for elliptic problems*, SIAM J. Sci. Comput., 30 (2008), pp. 1806–1824.
[22] J. Qiu, *A numerical comparison of the Lax–Wendroff discontinuous Galerkin method based on different numerical fluxes*, J. Sci. Comput., 30 (2007), pp. 345–367.
[23] J. Qiu, M. Dumbser, and C.-W. Shu, *The discontinuous Galerkin method with Lax–Wendroff type time discretizations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4528–4543.
[24] L. Rannabauer, M. Dumbser, and M. Bader, *ADER-DG with a-posteriori finite-volume limiting to simulate tsunamis in a parallel adaptive mesh refinement framework*, Comput. & Fluids, 173 (2018), pp. 299–306.
[25] W. H. Reed and T. Hill, *Triangular mesh methods for the neutron transport equation*, tech. report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.
[26] C. Shi and C.-W. Shu, *On local conservation of numerical methods for conservation laws*, Comput. & Fluids, 169 (2018), pp. 3–9.
[27] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, J. Comput. Phys., 83 (1989), pp. 32–78.
[28] Z. Sun and C.-W. Shu, *Stability analysis and error estimates of Lax–Wendroff discontinuous Galerkin methods for linear conservation laws*, ESAIM Math. Model. Numer. Anal., 51 (2017), pp. 1063–1087.
[29] V. A. Titarev and E. F. Toro, *ADER: Arbitrary high order Godunov approach*, J. Sci. Comput., 17 (2002), pp. 609–618.
[30] E. F. Toro, R. Millington, and L. Nejad, *Towards very high order Godunov schemes*, in Godunov Methods: Theory and Applications, Springer, 2001, pp. 907–940.
[31] B. van Leer, M. Lo, and M. van Raalte, *A discontinuous Galerkin method for diffusion based on recovery*, in 18th AIAA Computational Fluid Dynamics Conference, 2007, p. 4083.
[32] B. Van Leer and S. Nomura, *Discontinuous Galerkin for diffusion*, in 17th AIAA Computational Fluid Dynamics Conference, 2005, p. 5108.
[33] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), pp. 115–173.
[34] J. Yan and C.-W. Shu, *A local discontinuous Galerkin method for KdV type equations*, SIAM J. Numer. Anal., 40 (2002), pp. 769–791.
[35] Q. Zhang, *Third order explicit Runge-Kutta discontinuous Galerkin method for linear conservation law with inflow boundary condition*, J. Sci. Comput., 46 (2011), pp. 294–313.