Runge-Kutta Central Discontinuous Galerkin Methods for the Special Relativistic Hydrodynamics

Jian Zhao\textsuperscript{1} and Huazhong Tang\textsuperscript{2,\textsuperscript{*}}

\textsuperscript{1} HEDPS, CAPT & LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China.
\textsuperscript{2} HEDPS, CAPT & LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China; School of Mathematics and Computational Science, Xiangtan University, Hunan Province, Xiangtan 411105, P.R. China.

Received 2 November 2016; Accepted (in revised version) 27 January 2017

Abstract. This paper develops Runge-Kutta $P^k$-based central discontinuous Galerkin (CDG) methods with WENO limiter to the one- and two-dimensional special relativistic hydrodynamical (RHD) equations, $K = 1, 2, 3$. Different from the non-central DG methods, the Runge-Kutta CDG methods have to find two approximate solutions defined on mutually dual meshes. For each mesh, the CDG approximate solutions on its dual mesh are used to calculate the flux values in the cell and on the cell boundary so that the approximate solutions on mutually dual meshes are coupled with each other, and the use of numerical flux will be avoided. The WENO limiter is adaptively implemented via two steps: the “troubled” cells are first identified by using a modified TVB minmod function, and then the WENO technique is used to locally reconstruct new polynomials of degree $(2K+1)$ replacing the CDG solutions inside the “troubled” cells by the cell average values of the CDG solutions in the neighboring cells as well as the original cell averages of the “troubled” cells. Because the WENO limiter is only employed for finite “troubled” cells, the computational cost can be as little as possible. The accuracy of the CDG without the numerical dissipation is analyzed and calculation of the flux integrals over the cells is also addressed. Several test problems in one and two dimensions are solved by using our Runge-Kutta CDG methods with WENO limiter. The computations demonstrate that our methods are stable, accurate, and robust in solving complex RHD problems.

AMS subject classifications: 76M10, 76M25, 76Y05, 76N15

Key words: Central discontinuous Galerkin method, WENO limiter, Runge-Kutta time discretization, relativistic hydrodynamics.

\textsuperscript{*}Corresponding author. Email addresses: everease@163.com (J. Zhao); hztang@math.pku.edu.cn (H. Z. Tang)
1 Introduction

Relativistic fluid widely appears in nuclear physics, astrophysics, plasma physics, and other fields. For example, in the physical phenomena such as the formation of neutron stars and black holes and the high-speed jet, the local fluid velocity may be close to the speed of light, at this time the relativistic effect can not be neglected and the relativistic fluid dynamics (RHD) is needed. Because the RHD equations are more complicated, their theoretical analysis is impractical so that conversely numerical simulation has become a primary and powerful way to study and understand the physical mechanisms in the RHDs.

The pioneering numerical work may date back to the finite difference code via artificial viscosity for the spherically symmetric general RHD equations in the Lagrangian coordinate [32, 33]. Wilson first attempted to solve multi-dimensional RHD equations in the Eulerian coordinate by using the finite difference method with the artificial viscosity technique [47]. Since 1990s, the numerical study of the RHDs began to attract considerable attention, and various modern shock-capturing methods with an exact or approximate Riemann solver have been developed for the RHD equations, the readers are referred to the early review articles [31, 46]. Some examples on existing methods, which are extensions of Godunov type shock capturing methods, are the upwind schemes based on local linearization [16, 17], the two shock approximation solvers [1, 12, 35], flux-vector splitting scheme [14], HLL (Harten-Lax-van Leer) schemes [15, 41], HLLC (Harten-Lax-van Leer-Contact) scheme [34], non-oscillatory essentially (ENO) schemes [13, 58], and kinetic schemes [21, 55] and so on. Recently the second author and his co-workers developed adaptive moving mesh method [18], derived the second-order accurate generalized Riemann problem (GRP) methods for the one- and two-dimensional RHD equations [56, 57], and the finite volume local evolution Galerkin scheme for two-dimensional RHD equations [48]. Later, the third-order accurate GRP scheme in [54] was extended to the one-dimensional RHD equations [53], and the direct Eulerian GRP scheme was developed for the spherically symmetric general relativistic hydrodynamics [49]. The physical-constraints-preserving (PCP) schemes were also studied for the special RHD equations recently. The high-order accurate PCP finite difference weighted essentially non-oscillatory (WENO) schemes and discontinuous Galerkin (DG) methods were proposed in [36, 50, 52]. Moreover, the set of admissible states and the PCP schemes of the ideal relativistic magnetohydrodynamics was studied for the first time in [51], where the importance of divergence-free fields was revealed in achieving PCP methods especially.

The DG methods have been rapidly developed in recent decades and has become a kind of important methods in computational fluid dynamics. They are easy to achieve high order accuracy, suitable for parallel computing, and adapt to complex domain boundary. The DG method was first developed by Reed and Hill [38] to solve steady-state scalar linear hyperbolic equation but it did not attract people’s attention. A major development of the DG method was carried out in a series of papers [4, 6–8, 10], where the DG spatial approximation was combined with explicit Runge-Kutta time discretization to develop
Runge-Kutta DG (RKDG) methods and a general framework of DG methods was established for the nonlinear equation or system. After that, the RKDG methods began to get a wide range of research and application, such as the Euler equations [3,39,45], Maxwell equations [5], nonlinear Dirac equations [42] etc. Moreover, the DG methods have also been used to solve other partial differential equations, such as convection-diffusion type equation or system [2,9] and Hamilton-Jacobi equation [19,23] etc. The readers are referred to the review article [11]. The Runge-Kutta CDG methods [28] were developed by combing RKDG methods and central scheme [27] and found two approximate solutions defined on mutually dual meshes. Although two approximate solutions are redundant, the numerical flux with any complicated and time-consuming exact or approximate Riemann solver may be avoided due to the use of the solution on the dual mesh to calculate the flux at the cell interface. It is one of the advantages of the central scheme. Because the Runge-Kutta CDG methods can be considered as a variant of RKDG methods, they keep many advantages of RKDG methods, such as compact stencil and parallel implementation etc. Moreover, the Runge-Kutta CDG methods allow a larger CFL number than RKDG methods [40] and reduce numerical oscillations for some problems. Recently, they are also attractive and promising for magnetohydrodynamics, because of their easily achieving globally divergence-free numerical magnetic field [24,25,63]. The Runge-Kutta CDG methods have also been used to solve the Hamilton-Jacobi equation [26] and so on.

A deficiency of the RKDG methods is that when the strong discontinuity appears in the solution, the numerical oscillations should be suppressed after each Runge-Kutta inner stage or after some complete Runge-Kutta steps by using the nonlinear limiter, which is a commonly used technique of the modern shock-capturing methods for hyperbolic conservation laws. The commonly used limiter is the minmod limiter, which limits the slope of solution such that the values of limited solution in the cell falls in the certain interval determined by the cell average values of neighboring cells. The minmod limiter has good robustness but becomes only first-order accurate near extreme points. The modified TVB minmod limiter is given in [7] and applied to the RKDG methods. It does not limit the solution near extreme points by choosing a parameter $M$, thus the accuracy of RKDG methods is not destroyed near the extreme point. In general, for nonlinear equation, the parameter is dependent on the problem, and the accuracy of $P^k$-based RKDG methods for $k \geq 3$ may still be destroyed because more than three of the higher order moments will be set to zero in the modified TVB minmod limiter. Besides those commonly used limiters, some other limiters are proposed, such as the moment based limiters [3] and its improvement [20] etc. Those limiters may suppress numerical oscillations near the discontinuity, however, the accuracy of RKDG methods may be reduced in the some region.

In the modern shock-capturing methods, the ENO and WENO methods are more robust than the slope limiters especially for high order schemes and have been widely used, see the review article [43]. An attempt was made to use them as limiters for the DG methods [37,64,65]. The WENO limiter first identifies the “troubled” cells by using a modified TVB minmod function, and then new polynomials inside the “troubled” cells
are locally reconstructed to replace the DG solutions by using the WENO technique based on the cell average values of the DG solutions in the neighboring cells as well as the original cell averages of the “troubled” cells. It is only employed for less “troubled” cells, so the computational cost can be as little as possible.

This paper proposes the Runge-Kutta $P^K$-based CDG methods with WENO limiter for the one- and two-dimensional special RHD equations, $K=1,2,3$. It is organized as follows. Section 2 introduces the system of special RHD equations. Section 3 proposes Runge-Kutta $P^K$-based CDG methods with WENO limiter. Section 4 gives some discussions of the Runge-Kutta CDG methods. Section 5 gives several numerical examples to verify the accuracy robustness, and effectiveness of the proposed Runge-Kutta CDG methods. Concluding remarks are presented in Section 6.

2 Special RHD equations

This section introduces the governing equations of the special relativistic hydrodynamics (RHD). Similar to the non-relativistic case, the special RHD equations may be established by the laws of local baryon number conservation and energy-momentum conservation [22] and cast into the following covariant form

$$\begin{align*}
\partial_\alpha (\rho u^\alpha) &= 0, \\
\partial_\alpha \left( \rho u^\alpha u^\beta + pg^{\alpha\beta} \right) &= 0,
\end{align*}$$

(2.1)

where the Greek indices $\alpha$ and $\beta$ run from 0 to 3, $\partial_\alpha = \partial_{x^\alpha}$ stands for the covariant derivative, $g^{\alpha\beta}$ denotes the metric tensor and is restricted to the Minkowski tensor throughout the paper, i.e. $(g^{\alpha\beta})_{4 \times 4} = \text{diag}\{-1,1,1,1\}$, $\rho$, $u^\alpha$ and $p$ denote the rest-mass density, four-velocity vector, and pressure, respectively, and $h$ is the specific enthalpy defined by

$$h = 1 + e + \frac{p}{\rho},$$

(2.2)

here $e$ denotes the specific internal energy. For the sake of convenience, units in which the speed of light is equal to one will be used so that $x^\alpha = (t,x_1,x_2,x_3)^T$ and $u^\alpha = \gamma (1,v_1,v_2,v_3)^T$, where $\gamma = 1/\sqrt{1-v^2}$ is the Lorentz factor and $v := \sqrt{v_1^2 + v_2^2 + v_3^2}$ is the size of fluid velocity.

In order to close the above system (2.1), an equation of state (EOS) for the thermodynamical variables

$$p = p(\rho, e),$$

(2.3)

is needed. For example, the EOS for an ideal gas can be expressed in the $\Gamma$-law form

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

(2.4)

where $\Gamma$ is the adiabatic index, taken as $5/3$ for the mildly relativistic case and $4/3$ for the ultra-relativistic case.
The covariant form of special RHD equations (2.1) is usually written into a time-dependent system of conservation laws in the laboratory frame as follows

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i(U)}{\partial x_i} = 0,$$

(2.5)

where $U$ is conservative variable vector and $F_i$ denotes and the flux vector in the $x_i$ direction, $i = 1, \cdots, d$. For example, in the case of $d = 3$, the detailed expressions of $U$ and $F_i$ are

$$U = \left( D, m_1, m_2, m_3, E \right)^T,$n

$$F_1 = \left( Dv_1, m_1 v_1 + p, m_2 v_1, m_3 v_1 m_1 \right)^T,$n

$$F_2 = \left( Dv_2, m_1 v_2, m_2 v_2 + p, m_3 v_2 m_2 \right)^T,$n

$$F_3 = \left( Dv_3, m_1 v_3, m_2 v_3, m_3 v_3 + p, m_3 \right)^T,$$n

(2.6)

here $D = \rho \gamma$, $m_i = \rho h \gamma^2 v_i$, and $E = \rho h \gamma^2 - p$ denote the mass, $x_i$-momentum, and energy densities relative to the laboratory frame, respectively.

The formal structure of (2.5) is identical to that of the three-dimensional non-relativistic Euler equations. The momentum equations in (2.5) are only with a Lorentz-contracted momentum density replacing $\rho v_i$ in the non-relativistic Euler equations. When the fluid velocity is small ($v \ll 1 = c$) and the velocity of the internal (microscopic) motion of the fluid particles is small, the RHD equations (2.5) reduce to the non-relativistic Euler equations. The system (2.5) also satisfies the properties of the rotational invariance and the homogeneity as well as the hyperbolicity in time when (2.4) is used, see [61]. However, in comparison to the non-relativistic Euler equations, a strong coupling between the hydrodynamic equations is introduced and additional numerical difficulties are posed due to the relations between the laboratory quantities (the mass density $D$, the momentum density $m_i$, and the energy density $E$) and the quantities in the local rest frame (the mass density $\rho$, and the fluid velocity $v_i$, the internal energy density $e$). Especially, the flux $F_i$ in (2.5) can not be formulated in an explicit form of the conservative vector $U$ and the physical constraints $E \geq D$, $\rho > 0$, $p > 0$, and $v < 1$ have to be fulfilled. Thus, in practical computations of the system (2.5) by using the shock-capturing methods, the primitive variable vector $V = (\rho, v_1, \cdots, v_d, p)^T$ has to be first recovered from the known conservative vector $U = (D, m_1, \cdots, m_d, E)^T$ at each time step by numerically solving a nonlinear pressure equation such as

$$E + p = D\gamma + \frac{\Gamma}{\Gamma - 1} p \gamma^2,$$

(2.7)

where $\gamma = (1 - |m|^2/(E + p)^2)^{-1/2}$. Any standard root-finding algorithm, e.g. Newton’s iteration, may be used to solve (2.7) to get the pressure, and then $\gamma$, $\rho$, $e$, $h$, and $v_i$ in order, the readers are referred to [62] for the choice of initial guess.
3 Runge-Kutta CDG methods

This section gives the Runge-Kutta central DG methods for the hyperbolic conservation laws. For the sake of simplicity, one-dimensional scalar equation

\[
\frac{\partial u(x,t)}{\partial t} + \frac{\partial f(u(x,t))}{\partial x} = 0, \quad x \in \Omega,
\]

is taken as an example to introduce the Runge-Kutta CDG methods [28]. Similar to the non-central RKDG methods, the Runge-Kutta CDG methods also employ the discontinuous Galerkin finite element in the spatial discretization and the explicit Runge-Kutta method for the time discretization. Their difference between them is that Runge-Kutta CDG methods need two mutually dual meshes, see the one-dimensional schematic diagram in Fig. 1 for the mesh \( \{ C_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}), \forall j \in \mathbb{Z} \} \) and its dual mesh \( \{ D_{j+\frac{1}{2}} = (x_j, x_{j+1}), \forall j \in \mathbb{Z} \} \).

Figure 1: Schematic diagram of mutually dual meshes for the 1D Runge-Kutta CDG methods.

The aim of Runge-Kutta CDG methods is to find two approximate solutions \( u^C_h(x,t) \) and \( u^D_h(x,t) \) such that at any time \( t \in (0,T] \), they belong to the following finite spaces respectively

\[
\mathcal{V}^C := \left\{ v(x) \in L^1(\Omega) | v(x) \in P^K(C_j), x \in C_j \subset \Omega, \forall j \right\},
\]

\[
\mathcal{V}^D := \left\{ w(x) \in L^1(\Omega) | w(x) \in P^K(D_{j+\frac{1}{2}}), x \in D_{j+\frac{1}{2}} \subset \Omega, \forall j \right\}.
\]

Consider the CDG scheme for the approximate solution \( u^C_h \). Multiplying (3.1) by the test function \( v(x) \in P^K(C_j) \) and integrating it over the cell \( C_j \) by parts gives

\[
\frac{d}{dt} \int_{C_j} u^C_h v dx = \int_{C_j} f(u^C_h) \frac{\partial v}{\partial x} dx - f(u(x_{j+\frac{1}{2}}, t)) v(x_{j+\frac{1}{2}}) + f(u(x_{j-\frac{1}{2}}, t)) v(x_{j-\frac{1}{2}}).
\]

If replacing \( u \) at the left- and right-hand sides of (3.2) with the approximate solution \( u^C_h \) and \( u^D_h \), respectively, then one has

\[
\frac{d}{dt} \int_{C_j} u^C_h v dx = \frac{1}{\tau_{\text{max}}} \int_{C_j} (u^D_h - u^C_h) v(x) dx + \int_{C_j} f(u^C_h) \frac{\partial v}{\partial x} dx
\]

\[
- f(u^D_h(x_{j+\frac{1}{2}}, t)) v(x_{j+\frac{1}{2}}) + f(u^D_h(x_{j-\frac{1}{2}}, t)) v(x_{j-\frac{1}{2}}),
\]

where the first term at the right-hand side of (3.3) denotes the numerical dissipation term borrowing from the central scheme [27], and \( \tau_{\text{max}} \) denotes the maximum time step size.
allowed by the CFL condition. Because the approximate solution \( u_h^{D} \) is continuous at the boundary of cell \( C_p \), the fluxes \( f(u(x_{j+\frac{1}{2}}, t)) \) may be directly evaluated and thus numerical flux is not required in the CDG methods.

If choosing \( \phi_j^{(l)}(x) \), \( l = 0, \cdots, K \), to denote a basis of the space \( \mathbb{P}^k(C_j) \), then \( u_h^C \) may be expressed as

\[
u_h^C(x,t) = \sum_{l=0}^{K} u_{j}^{C,(l)}(t) \phi_j^{(l)}(x) =: u_j^C(x,t), \quad x \in C_j.
\]

Replacing \( v(x) \) in (3.3) with the basis function \( \phi_j^{(l)}(x) \) and using the numerical quadrature with \( q \) points to calculate the integral with flux \( f(u) \) gives the semi-discrete scheme for \( u_h^C \) as follows

\[
\begin{align*}
\sum_{l=0}^{K} \left( \int_{C_j} \phi_j^{(l)}(x) \phi_j^{(l)}(x) dx \right) \frac{du_j^{C,(l)}}{dt} &= \frac{1}{\tau_{\max}} \int_{C_j} (u_h^D - u_h^C) \phi_j^{(l)}(x) dx - f(u_h^D(x_{j+\frac{1}{2}}, t)) \phi_j^{(l)}(x_{j+\frac{1}{2}}) + f(u_h^D(x_{j-\frac{1}{2}}, t)) \phi_j^{(l)}(x_{j-\frac{1}{2}}) \\
&\quad + h_j \sum_{m=1}^{q} \omega_m^C f(u_h^D(x_m, t)) \frac{\partial}{\partial x} \phi_j^{(l)}(x_m), \quad \ell = 0, \cdots, K, \quad (3.4)
\end{align*}
\]

where \( h_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}} \) and \( x_m^C \) and \( \omega_m^C \) denote the point and weight for the numerical integration over the cell \( C_j \), \( m = 1, \cdots, q \). It needs to be pointed out that the first term at the right-hand side of (3.4) is an integral of piecewise polynomial and may be exactly calculated.

The semi-discrete scheme for the approximate solution \( u_h^D \) may be similarly derived. If choosing a basis of the space \( \mathbb{P}^k(D_{j+\frac{1}{2}}) \) as \{ \( \phi_j^{(l)}(x) \), \( l = 0, \cdots, K \}\), then the semi-discrete scheme for \( u_h^D \) is given as follows

\[
\begin{align*}
\sum_{l=0}^{K} \left( \int_{D_{j+\frac{1}{2}}} \phi_j^{(l)}(x) \phi_j^{(l)}(x) dx \right) \frac{du_j^{D,(l)}}{dt} &= \frac{1}{\tau_{\max}} \int_{D_{j+\frac{1}{2}}} (u_h^C - u_h^D) \phi_j^{(l)}(x) dx - f(u_h^C(x_{j+1}, t)) \phi_j^{(l)}(x_{j+1}) + f(u_h^C(x_{j}, t)) \phi_j^{(l)}(x_{j}) \\
&\quad + h_{j+\frac{1}{2}} \sum_{m=1}^{q} \omega_m^D f(u_h^C(x_m, t)) \frac{\partial}{\partial x} \phi_j^{(l)}(x_m), \quad \ell = 0, \cdots, K. \quad (3.5)
\end{align*}
\]

**Remark 3.1.** If \( f(u) = u \), then two approximate solutions satisfy \[29\]

\[
\frac{1}{2} \frac{d}{dt} \int_{\Omega} \left( (u_h^C)^2 + (u_h^D)^2 \right) dx = -\frac{1}{\tau_{\max}} \int_{\Omega} (u_h^C - u_h^D)^2 dx \leq 0. \quad (3.6)
\]
It can be known, that is why the term $\frac{1}{\tau_{\text{max}}} \int_{C_j} (u^D_h - u^C_h) v(x) dx$ in (3.3) is called as numerical dissipation.

**Remark 3.2.** It is worth noting that the flux within the cell $C_j$ in (3.3) is evaluated by using $u^D_h$, but the approximate solution $u^D_h$ is not continuous at $x_j$, thus before using the numerical integration to evaluate the integral of flux over $C_j$ in (3.3), one has to split it into two parts

$$
\int_{x_j - \frac{1}{2}}^{x_{j+1} \frac{1}{2}} f(u^D_h) \frac{\partial v}{\partial x} dx = \int_{x_j - \frac{1}{2}}^{x_j} f(u^D_h) \frac{\partial v}{\partial x} dx + \int_{x_j}^{x_{j+1} \frac{1}{2}} f(u^D_h) \frac{\partial v}{\partial x} dx,
$$

(3.7)

and then use Gaussian quadrature with $K+1$ points to calculate two integrals at the right-hand side of the above equation. The flux integral in (3.5) should be similarly treated. Section 4.2 will give a further discussion on the evaluation of such flux integral.

Both semi-discrete CDG schemes (3.4) and (3.5) may be cast into the following abstract form

$$\frac{\partial u}{\partial t} = L(u),$$

which is a nonlinear system of ordinary differential equation of $U$ with respect to $t$, and thus the time derivatives may be further approximated to give the fully-discrete CDG methods may be derived for the degrees of freedom or the moments by using the third-order accurate TVD (total variation diminishing) Runge-Kutta method [44]

$$U^{(1)} = U^n + \Delta t_n L(U^n),$$

$$U^{(2)} = \frac{3}{4} U^n + \frac{1}{4} (U^{(1)} + \Delta t_n L(U^{(1)})),$$

$$U^{n+1} = \frac{1}{3} U^n + \frac{2}{3} (U^{(2)} + \Delta t_n L(U^{(2)})),$$

(3.8)

or the fourth-order accurate non-TVD Runge-Kutta method

$$U^{(1)} = U^n + \frac{1}{2} \Delta t_n L(U^n),$$

$$U^{(2)} = U^n + \frac{1}{2} \Delta t_n L(U^{(1)}),$$

$$U^{(3)} = U^n + \Delta t_n L(U^{(2)}),$$

$$U^{n+1} = \frac{1}{3} (U^{(1)} + 2U^{(2)} + 3U^{(3)} - U^n + \frac{1}{2} \Delta t_n L(U^{(3)})),$$

(3.9)

and so on.

As mentioned above, $\tau_{\text{max}}$ is determined by using the CFL condition. After determining $\tau_{\text{max}} = \tau_n$ at $t = t_n$, the practical time stepsize $\Delta t_n$ should satisfy $0 < \Delta t_n \leq \tau_n$. If denoting
\[ \theta = \Delta t_n / \tau_n, \text{ then } \theta \in (0,1]. \] For hyperbolic equations, \( \Delta t_n \) may usually be taken as \( \tau_n \), that is, \( \theta = 1 \).

Although the Runge-Kutta CDG methods are only introduced for one-dimensional scalar conservation law, their extension to one-dimensional RHD equations and two-dimensional rectangular mesh is easy and direct. Similar to the non-central RKDG methods, the limiting procedure is necessary for the Runge-Kutta CDG methods when the solution contains strong discontinuity. The WENO limiting procedure in Section 3.3 of [62] may be directly and independently applied to the solutions \( u^C_h \) and \( u^D_h \) of Runge-Kutta CDG methods by the following two steps:

- identify the “troubled” cells in the meshes \( \{C_j\} \) and \( \{D_{j+1/2}\} \), namely, those cells which might need the limiting procedure;
- replace the CDG solution polynomials \( u^C_h \) and \( u^D_h \) in those “troubled” cells with WENO reconstructed polynomials of degree \( (2K+1) \), denoted by \( u^{C,WENO}_h \) and \( u^{D,WENO}_h \), which maintain the original cell averages (conservation) and the accuracy, but have less numerical oscillation.

In order to save the length of paper, those details are omitted here.

## 4 Some discussions of Runge-Kutta CDG methods

In comparison to the non-central RKDG methods, the Runge-Kutta CDG methods has a numerical dissipation term in different form. Remark 3.1 has shown that such dissipation term is important for the \( L^2 \) stability of CDG methods. This section discusses the accuracy of Runge-Kutta CDG methods without the numerical dissipation by following Fourier type analysis in order to understand the impact of numerical dissipation term on the accuracy of CDG methods, see Section 4.1. Furthermore, this section will also discuss the calculation of flux integral over the cell mentioned in Remark 3.2, see Section 4.2. The general CDG method was previously studied by following a similar analysis in [29] and the CDG method without numerical dissipation was analyzed for diffusion equations in [30].

### 4.1 Accuracy of CDG methods without numerical dissipation

The semi-discrete version of Runge-Kutta CDG methods without numerical dissipation can be written as follows

\[ \frac{d}{dt} \int_{C_j} u^C_h v dx = \int_{C_j} f(u^D_h) \frac{\partial v}{\partial x} dx - f(u^D_h(x_{j+1/2}, t)) v(x_{j+1/2}) + f(u^D_h(x_{j-1/2}, t)) v(x_{j-1/2}), \quad \forall v(x) \in P^K(C_j), \] (4.1a)
\[
\frac{d}{dt} \int_{D_{j+\frac{1}{2}}} u_h^D \, w \, dx = \int_{D_{j+\frac{1}{2}}} f(u_h^C) \frac{\partial w}{\partial x} \, dx - f(u_h^C(x_{j+1}, t))w(x_{j+1}) \\
+ f(u_h^C(x_j, t))w(x_j), \quad \forall w(x) \in \mathbb{P}^k(D_{j+\frac{1}{2}}).
\] (4.1b)

The accuracy of $P^1$-based CDG methods without numerical dissipation is first discussed here by using the Fourier method similar to [59, 60]. Use (4.1) to solve the scalar equation

\[
u_t + u_x = 0, \quad x \in [0, 2\pi],
\] (4.2)

subject to the initial condition $u(x, 0) = \sin(x)$.

For the CDG solution on the mesh $\{C_j\}$. As shown in Fig. 2, for the sake of convenience, the degrees of freedom are chosen as the function values at $2N$ points distributed with equal distance

\[
u_{j-\frac{1}{4}}, \quad \nu_{j+\frac{1}{4}}, \quad j = 1, \ldots, N,
\]

instead of all order moments $\{u_j^{C,(l)}(t)\}$.

![Figure 2: Schematic diagram of degrees of freedom.](image)

Within the cell $C_j$, the solution $u_j^C(x, t)$ may be expressed as

\[
u_j^C(x, t) = \nu_{j-\frac{1}{4}}(t)\phi_{j-\frac{1}{4}}(x) + \nu_{j+\frac{1}{4}}(t)\phi_{j+\frac{1}{4}}(x),
\]

where $\phi_{j \pm \frac{1}{4}}(x)$ denote the basis functions, given by

\[
\phi_{j-\frac{1}{4}}(x) = \frac{1}{2} - \frac{2(x-x_j)}{h}, \quad \phi_{j+\frac{1}{4}}(x) = \frac{1}{2} + \frac{2(x-x_j)}{h}.
\]

Similarly, within the cell $D_{j+\frac{1}{2}}$ the solution $u_j^D(x, t)$ may be written as follows

\[
u_j^D(x, t) = \nu_{j+\frac{1}{4}}(t)\phi_{j+\frac{1}{4}}(x) + \nu_{j+\frac{3}{4}}(t)\phi_{j+\frac{3}{4}}(x).
\]

If replacing $v(x)$ in (4.1) with $\phi_{j-\frac{1}{4}}(x)$ and $\phi_{j+\frac{1}{4}}(x)$, and $w(x)$ with $\phi_{j+\frac{1}{4}}(x)$ and $\phi_{j+\frac{3}{4}}(x)$,
respectively, and performing the mass matrix inversion, then one has

\[
\begin{align*}
\frac{du^C_{j-\frac{1}{4}}}{dt} &= \frac{1}{4h}(5u^D_{j-\frac{3}{4}} - u^D_{j-\frac{1}{4}} - 5u^D_{j+\frac{1}{4}} + u^D_{j+\frac{3}{4}}), \\
\frac{du^C_{j+\frac{1}{4}}}{dt} &= \frac{1}{4h}(-u^D_{j-\frac{1}{4}} + 5u^D_{j-\frac{3}{4}} + u^D_{j+\frac{3}{4}} - 5u^D_{j+\frac{1}{4}}), \\
\frac{du^D_{j+\frac{1}{4}}}{dt} &= \frac{1}{4h}(5u^C_{j-\frac{1}{4}} - u^C_{j+\frac{1}{4}} - 5u^C_{j+\frac{3}{4}} + u^C_{j+\frac{5}{4}}), \\
\frac{du^D_{j+\frac{3}{4}}}{dt} &= \frac{1}{4h}(-u^C_{j-\frac{1}{4}} + 5u^C_{j+\frac{1}{4}} + u^C_{j+\frac{3}{4}} - 5u^C_{j+\frac{5}{4}}),
\end{align*}
\]

which may be rewritten as follows

\[
\frac{d}{dt}\begin{pmatrix}
  u^C_{j-\frac{1}{4}} \\
  u^C_{j+\frac{1}{4}} \\
  u^C_{j+\frac{3}{4}} \\
  u^C_{j+\frac{5}{4}}
\end{pmatrix} = A\begin{pmatrix}
  u^C_{j-\frac{1}{4}} \\
  u^C_{j-\frac{1}{4}} \\
  u^C_{j+\frac{1}{4}} \\
  u^C_{j+\frac{3}{4}}
\end{pmatrix} + B\begin{pmatrix}
  u^C_{j-\frac{1}{4}} \\
  u^C_{j-\frac{1}{4}} \\
  u^C_{j+\frac{1}{4}} \\
  u^C_{j+\frac{3}{4}}
\end{pmatrix} + C\begin{pmatrix}
  u^C_{j+\frac{1}{2}} \\
  u^C_{j+\frac{3}{2}} \\
  u^C_{j+\frac{5}{2}} \\
  u^C_{j+\frac{7}{2}}
\end{pmatrix},
\]

(4.3)

where three coefficient matrices are respectively given by

\[
A = \begin{pmatrix}
  0 & 0 & \frac{5}{4h} & -\frac{1}{3h} \\
  0 & 0 & -\frac{1}{4h} & \frac{5}{3h} \\
  0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0
\end{pmatrix}, \quad
B = \begin{pmatrix}
  0 & 0 & -\frac{5}{4h} & \frac{1}{4h} \\
  0 & 0 & \frac{1}{4h} & -\frac{5}{4h} \\
  \frac{5}{4h} & -\frac{1}{4h} & 0 & 0 \\
  -\frac{1}{4h} & \frac{5}{4h} & 0 & 0
\end{pmatrix}, \quad
C = \begin{pmatrix}
  0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 \\
  -\frac{5}{4h} & \frac{1}{4h} & 0 & 0 \\
  \frac{5}{4h} & -\frac{1}{4h} & 0 & 0
\end{pmatrix}.
\]

Because the solution of (4.2) is periodic and the mesh is uniform, the solution of (4.3) may be assumed to be of the following form

\[
\begin{pmatrix}
  u^C_{j-\frac{1}{4}}(t) \\
  u^C_{j+\frac{1}{4}}(t) \\
  u^C_{j+\frac{3}{4}}(t) \\
  u^C_{j+\frac{5}{4}}(t)
\end{pmatrix} = \begin{pmatrix}
  \hat{u}^C_{\eta_0 - \frac{1}{4}}(t) \\
  \hat{u}^C_{\eta_0 + \frac{1}{4}}(t) \\
  \hat{u}^C_{\eta_0 + \frac{3}{4}}(t) \\
  \hat{u}^C_{\eta_0 + \frac{5}{4}}(t)
\end{pmatrix} e^{iqx},
\]

(4.4)

where \(i = \sqrt{-1}\) is the imaginary unit. Substituting (4.4) into (4.3) gives

\[
\frac{d}{dt}\begin{pmatrix}
  \hat{u}^C_{\eta_0 - \frac{1}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{1}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{3}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{5}{4}}
\end{pmatrix} = G(q,h)\begin{pmatrix}
  \hat{u}^C_{\eta_0 - \frac{1}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{1}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{3}{4}} \\
  \hat{u}^C_{\eta_0 + \frac{5}{4}}
\end{pmatrix},
\]

(4.5)
where $G(q,h)$ denotes the amplification matrix and is defined by

$$G(q,h) = Ae^{-i\alpha} + B + Ce^{i\alpha},$$

(4.6)

here $\alpha = qh$ and $h$ denotes the spatial step size. Four eigenvalues of $G$ are

$$\lambda_{1,2} = \frac{1}{h} \sqrt{2\cos \alpha - 2}, \quad \lambda_{3,4} = \frac{3}{2h} \sqrt{2\cos \alpha - 2},$$

and corresponding right eigenvectors may be taken as follows

$$r_1 = \frac{e^{-i\alpha} \sqrt{2\cos \alpha - 2}}{\sqrt{2\cos \alpha - 2}}, \quad r_2 = \frac{e^{i\alpha} \sqrt{2\cos \alpha - 2}}{\sqrt{2\cos \alpha - 2}}, \quad r_3 = \frac{-e^{-i\alpha} \sqrt{2\cos \alpha - 2}}{\sqrt{2\cos \alpha - 2}}, \quad r_4 = \frac{-e^{i\alpha} \sqrt{2\cos \alpha - 2}}{\sqrt{2\cos \alpha - 2}}.$$

Thus the solution of Eq. (4.3) may be expressed as follows

$$(u^c_j, u^d_j) = (u^c_{j+\frac{1}{4}}, u^d_{j+\frac{1}{4}}, u^c_{j+\frac{1}{4}}, u^d_{j+\frac{1}{4}})^T = c_1 e^{i\alpha x_j + \lambda t} r_1 + c_2 e^{i\alpha x_j + \lambda t} r_2 + c_3 e^{i\alpha x_j + \lambda t} r_3 + c_4 e^{i\alpha x_j + \lambda t} r_4,$$

where $c_i, 1 \leq i \leq 4$, are four undetermined coefficients.

Let us discuss the accuracy of methods. Take $q = 1$ and define

$$u^c_{j-\frac{1}{4}}(0) = e^{ix}, \quad u^c_{j+\frac{1}{4}}(0) = e^{ix}, \quad u^d_{j-\frac{1}{4}}(0) = e^{ix}, \quad u^d_{j+\frac{1}{4}}(0) = e^{ix},$$

(4.7)

then their imaginary parts satisfy the initial condition $u(x,0) = \sin(x)$. It should be pointed out that, the initial degrees of freedom in the DG methods are generally derived by using the $L^2$ projection to the initial condition, but the approach for setting initial value [60] is used here and does not effect the final results on accuracy.

The undetermined coefficients $c_i, 1 \leq i \leq 4$, may be determined by (4.7) as follows

$$c_{1,2} = \pm \frac{1}{2} \frac{\sqrt{2\cosh - 2}}{e^{-ih} - 1} \cos \frac{h}{4} + \frac{1}{4} (e^{\frac{3h}{4}} + e^{\frac{h}{4}}),$$

$$c_{3,4} = \pm i \frac{1}{2} \frac{\sqrt{2\cosh - 2}}{e^{-ih} - 1} \sin \frac{h}{4} + \frac{1}{4} (e^{\frac{3h}{4}} - e^{\frac{h}{4}}),$$

which may give the expression of $u^c_{j-\frac{1}{4}}(t)$. Using the Taylor expansion to the imaginary part of $u^c_{j-\frac{1}{4}}(t)$ with respect to $h$ gives

$$\text{Im}\{u^c_{j-\frac{1}{4}}(t)\} = \sin(x_{j-\frac{1}{4}} - t) + \frac{h}{4} \left( \cos(x_{j-\frac{1}{4}} - t) - \cos\left(x_{j-\frac{1}{4}} - \frac{3}{2} t\right) \right) + O(h^2).$$

(4.8)
The solutions of $P^1$-based methods with numerical dissipation satisfy [29]

$$
\text{Im}\{u_{j-\frac{1}{4}}^C(t)\} = \sin(x_{j-\frac{1}{4}} - t) + \sigma \sin(x_{j-\frac{1}{4}} - t) h^2 + \mathcal{O}(h^3),
$$

where $\sigma$ is a constant only depending on $\tau_{\text{max}}/h$. Comparing them gives their obvious difference. The similar differences may be given by using the Taylor expansion to the expression of $u_{j-\frac{1}{4}}^C(t), u_{j+\frac{1}{2}}^D(t)$, and $u_{j+\frac{1}{2}}^D(t)$. From the above analysis, it is seen that the $P^1$-based Runge-Kutta CDG methods without numerical dissipation term are only first-order accurate in space.

In the following, we use the $P^1$-based method to solve Eq. (4.2) in order to numerically demonstrate (4.8). In order to reduce the errors arising from the time discretization, the fourth-order accurate Runge-Kutta method (3.9) is used with the time stepsize $\Delta t = 0.01h$. Fig. 3 shows the time evolution of error at the point $x_{j-\frac{1}{4}}$. Except for a few moments, numerical results are highly consistent with the theoretical result given by (4.8). Moreover, Table 1 presents $l_1$ and $l_\infty$ errors of solution at $t = 15$, as well as the results estimated in theory. It is seen that the numerical results are in good agreement with the theoretical analysis.

![Figure 3](image_url)

Figure 3: The symbol "○" denotes the difference between the numerical and exact solutions of (4.2) at point $x_{j-\frac{1}{4}}$, $u_j^C(x_{j-\frac{1}{4}}, t) - u(x_{j-\frac{1}{4}}, t)$, while the solid line denote the result obtained by (4.8), $h(\cos(x_{j-\frac{1}{4}} - t) - \cos(x_{j-\frac{1}{4}} - \frac{3}{4}t)) / 4$, where $j = 50$. Numerical solution is obtained by using the $P^1$-based Runge-Kutta CDG methods without numerical dissipation but with 200 cells.

It is difficult to use the above Fourier method to accuracy of $P^2$- and $P^3$-based Runge-Kutta CDG methods without numerical dissipation. For this reason, the numerical experiments are provided to replace the above Fourier method. Table 2 lists errors and orders of solutions obtained by using the $P^2$- and $P^3$-based Runge-Kutta CDG methods without numerical dissipation. It is seen that the convergence rate of $P^2$-based Runge-Kutta CDG
Table 1: $l^1$ and $l^\infty$ errors and orders at $t = 15$ obtained by the $P^1$-based Runge-Kutta CDG methods without numerical dissipation for Eq. (4.2). The fourth-order Runge-Kutta method is employed and $h$ denotes the spatial stepsize.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$l^1$ error</th>
<th>order</th>
<th>$l^\infty$ error</th>
<th>order</th>
<th>$l^1$ error</th>
<th>order</th>
<th>$l^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2\pi/40$</td>
<td>1.90e-01</td>
<td></td>
<td>9.57e-02</td>
<td></td>
<td>1.80e-01</td>
<td></td>
<td>9.13e-02</td>
<td></td>
</tr>
<tr>
<td>$2\pi/80$</td>
<td>9.13e-02</td>
<td>1.05</td>
<td>4.64e-02</td>
<td>1.04</td>
<td>8.98e-02</td>
<td>1.00</td>
<td>4.53e-02</td>
<td>1.01</td>
</tr>
<tr>
<td>$2\pi/160$</td>
<td>4.51e-02</td>
<td>1.02</td>
<td>2.28e-02</td>
<td>1.03</td>
<td>4.49e-02</td>
<td>1.00</td>
<td>2.25e-02</td>
<td>1.01</td>
</tr>
<tr>
<td>$2\pi/320$</td>
<td>2.25e-02</td>
<td>1.00</td>
<td>1.13e-02</td>
<td>1.01</td>
<td>2.25e-02</td>
<td>1.00</td>
<td>1.12e-02</td>
<td>1.00</td>
</tr>
<tr>
<td>$2\pi/640$</td>
<td>1.12e-02</td>
<td>1.00</td>
<td>5.62e-03</td>
<td>1.01</td>
<td>1.12e-02</td>
<td>1.00</td>
<td>5.61e-03</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 2: Same as Table 1 except for $P^2$- and $P^3$- based Runge-Kutta CDG methods.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$P^2$ $l^1$ error</th>
<th>order</th>
<th>$P^2$ $l^\infty$ error</th>
<th>order</th>
<th>$P^3$ $l^1$ error</th>
<th>order</th>
<th>$P^3$ $l^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2\pi/40$</td>
<td>6.67e-05</td>
<td></td>
<td>2.62e-05</td>
<td></td>
<td>3.24e-06</td>
<td></td>
<td>1.83e-06</td>
<td></td>
</tr>
<tr>
<td>$2\pi/80$</td>
<td>6.80e-06</td>
<td>3.29</td>
<td>2.04e-06</td>
<td>3.68</td>
<td>4.01e-07</td>
<td>3.01</td>
<td>2.24e-07</td>
<td>3.03</td>
</tr>
<tr>
<td>$2\pi/160$</td>
<td>8.92e-07</td>
<td>2.93</td>
<td>3.73e-07</td>
<td>2.46</td>
<td>5.01e-08</td>
<td>3.00</td>
<td>2.77e-08</td>
<td>3.01</td>
</tr>
<tr>
<td>$2\pi/320$</td>
<td>9.24e-08</td>
<td>3.27</td>
<td>3.22e-08</td>
<td>3.53</td>
<td>6.26e-09</td>
<td>3.00</td>
<td>3.45e-09</td>
<td>3.01</td>
</tr>
<tr>
<td>$2\pi/640$</td>
<td>1.20e-08</td>
<td>2.94</td>
<td>4.79e-09</td>
<td>2.75</td>
<td>7.82e-10</td>
<td>3.00</td>
<td>4.31e-10</td>
<td>3.00</td>
</tr>
</tbody>
</table>

methods without numerical dissipation is essentially consistent with the predicated value 3, but the convergence rate of $P^3$-based Runge-Kutta CDG methods without numerical dissipation is about 3, lesser than the predicated value 4.

4.2 Discussion on the flux integrals over the cell

As mentioned in Remark 3.2, because the DG solution $u^D_h$ is discontinuous at the point $x_j$ which is an internal point of the cell $C_j = (x_{j-1}, x_{j+1})$, the integral of “flux” $f (u^D_h) \frac{\partial u}{\partial x}$ over $C_j$ becomes (3.7). If the Gaussian quadrature is used to evaluate such flux integral, then the numerical integration point number is twice the non-central RKDG methods. When Runge-Kutta CDG methods are used to solve two-dimensional conservation laws on the dual meshes displayed in Fig. 4, the the numerical integration point number becomes four times that of the non-central RKDG methods.

In order to reduce the computational cost of numerical integration, an attempt may be considered that only approximate solution on the dual mesh is used to evaluate the flux on the cell boundary for the DG approximations on the mesh, thus the integrals of DG solution over the dual cell may be avoided and the cost of numerical integration is
Figure 4: Schematic diagram of two-dimensional dual mesh for Runge-Kutta CDG methods.

hopefully reduced. Specifically, if using

$$\int_{C_j} f(u^C_h) \frac{\partial v}{\partial x} \, dx$$

to replace

$$\int_{C_j} f(u^D_h) \frac{\partial v}{\partial x} \, dx,$$

then semi-discrete CDG methods may be expressed as follows

$$\frac{d}{dt} \int_{C_j} u^C_h v \, dx = \frac{1}{\tau_{\text{max}}} \int_{C_j} (u^D_h - u^C_h) v(x) \, dx + \int_{C_j} f(u^C_h) \frac{\partial v}{\partial x} \, dx$$

$$- f(u^D_h(x_{j+\frac{1}{2}})) v(x_{j+\frac{1}{2}}) + f(u^D_h(x_{j-\frac{1}{2}})) v(x_{j-\frac{1}{2}}), \forall v(x) \in P^K(C_j),$$

$$\frac{d}{dt} \int_{D_{j+\frac{1}{2}}} u^D_h w \, dx = \frac{1}{\tau_{\text{max}}} \int_{D_{j+\frac{1}{2}}} (u^C_h - u^D_h) w(x) \, dx + \int_{D_{j+\frac{1}{2}}} f(u^D_h) \frac{\partial w}{\partial x} \, dx$$

$$- f(u^C_h(x_{j+\frac{1}{2}})) w(x_{j+\frac{1}{2}}) + f(u^C_h(x_{j-\frac{1}{2}})) w(x_{j-\frac{1}{2}}), \forall w(x) \in P^K(D_{j+\frac{1}{2}}).$$

In the following, the Runge-Kutta CDG methods based on (4.9) is called as the new Runge-Kutta CDG methods, otherwise the old Runge-Kutta CDG methods. A natural problem is whether such change does effect the stability? Consider the $P^1$-based methods. If using the similar way to that in Section 4.1 and some simple algebraic operations, and applying the scheme (4.9) to Eq. (4.2), then the evolution equation of the degrees of freedom may be derived as follows

$$\frac{d}{dt} \begin{pmatrix}
    u^C_{j+\frac{1}{2}} \\
    u^C_{j-\frac{3}{2}} \\
    u^C_{j+\frac{3}{2}} \\
    u^D_{j+\frac{1}{2}} \\
    u^D_{j-\frac{1}{2}} \\
    u^D_{j+\frac{3}{2}} \\
    u^D_{j-\frac{3}{2}}
\end{pmatrix}
= A \begin{pmatrix}
    u^C_{j+\frac{1}{2}} \\
    u^C_{j-\frac{3}{2}} \\
    u^C_{j+\frac{3}{2}} \\
    u^D_{j+\frac{1}{2}} \\
    u^D_{j-\frac{1}{2}} \\
    u^D_{j+\frac{3}{2}} \\
    u^D_{j-\frac{3}{2}}
\end{pmatrix}
+ B \begin{pmatrix}
    u^C_{j+\frac{1}{2}} \\
    u^C_{j-\frac{3}{2}} \\
    u^C_{j+\frac{3}{2}} \\
    u^D_{j+\frac{1}{2}} \\
    u^D_{j-\frac{1}{2}} \\
    u^D_{j+\frac{3}{2}} \\
    u^D_{j-\frac{3}{2}}
\end{pmatrix}
+ C \begin{pmatrix}
    u^C_{j+\frac{1}{2}} \\
    u^C_{j-\frac{3}{2}} \\
    u^C_{j+\frac{3}{2}} \\
    u^D_{j+\frac{1}{2}} \\
    u^D_{j-\frac{1}{2}} \\
    u^D_{j+\frac{3}{2}} \\
    u^D_{j-\frac{3}{2}}
\end{pmatrix},$$

(4.10)
where

\[
A = \begin{pmatrix}
0 & 0 & \frac{1}{16\tau_{\max}} + \frac{5}{4\bar{h}} & \frac{13}{16\tau_{\max}} + \frac{5}{4\bar{h}} \\
0 & 0 & \frac{-1}{16\tau_{\max}} - \frac{1}{4\bar{h}} & \frac{3}{16\tau_{\max}} - \frac{1}{4\bar{h}} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix},
\]

\[
B = \begin{pmatrix}
\frac{-1}{\tau_{\max}} & \frac{3}{2\bar{h}} & \frac{-1}{\tau_{\max}} + \frac{3}{2\bar{h}} & \frac{3}{16\tau_{\max}} + \frac{1}{4\bar{h}} & \frac{-1}{16\tau_{\max}} + \frac{1}{4\bar{h}} \\
\frac{3}{16\tau_{\max}} + \frac{5}{4\bar{h}} & \frac{-1}{16\tau_{\max}} + \frac{13}{16\tau_{\max}} + \frac{5}{4\bar{h}} & 3 & 0 & 0 \\
\frac{1}{16\tau_{\max}} - \frac{5}{4\bar{h}} & \frac{13}{16\tau_{\max}} - \frac{5}{4\bar{h}} & \frac{-1}{\tau_{\max}} - \frac{3}{2\bar{h}} & -\frac{3}{2\bar{h}} \\
\frac{1}{16\tau_{\max}} - \frac{5}{4\bar{h}} & \frac{1}{16\tau_{\max}} - \frac{5}{4\bar{h}} & 0 & 0 & 0
\end{pmatrix},
\]

\[
C = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\frac{3}{16\tau_{\max}} + \frac{1}{4\bar{h}} & \frac{-1}{16\tau_{\max}} + \frac{1}{4\bar{h}} & 0 & 0 \\
\frac{13}{16\tau_{\max}} - \frac{5}{4\bar{h}} & \frac{-5}{16\tau_{\max}} - \frac{5}{4\bar{h}} & 0 & 0
\end{pmatrix}.
\]

Because the mesh is uniform and the periodic condition is considered here, the solution is still assumed to be

\[
\begin{pmatrix}
\hat{u}^C_{\ell, \frac{1}{4}}(t) \\
\hat{u}^C_{\ell, \frac{3}{4}}(t) \\
\hat{\nu}^D_{\ell, \frac{1}{4}}(t) \\
\hat{\nu}^D_{\ell, \frac{3}{4}}(t)
\end{pmatrix} = e^{i\alpha x_j}
\begin{pmatrix}
\hat{u}^C_{\ell, -\frac{1}{4}}(t) \\
\hat{u}^C_{\ell, \frac{3}{4}}(t) \\
\hat{\nu}^D_{\ell, -\frac{1}{4}}(t) \\
\hat{\nu}^D_{\ell, \frac{3}{4}}(t)
\end{pmatrix}.
\]

If denoting \( u(t) = (\hat{u}^C_{\ell, -\frac{1}{4}}(t), \hat{u}^C_{\ell, \frac{3}{4}}(t), \hat{\nu}^D_{\ell, -\frac{1}{4}}(t), \hat{\nu}^D_{\ell, \frac{3}{4}}(t))^T \), then Eq. (4.10) reduces to

\[
\frac{d}{dt} u(t) = Gu(t),
\]

where the definition of amplification matrix \( G(q, h) \) is the same as that in (4.6), and its four eigenvalues are

\[
\begin{align*}
\lambda_1 &= \frac{1}{\tau_{\max}} \left( -1 + \frac{1}{8} (e^{\frac{\mu}{2}} (4\mu - 1) - e^{-\frac{\mu}{2}} (4\mu + 1) + \sqrt{a}) \right), \\
\lambda_2 &= \frac{1}{\tau_{\max}} \left( -1 + \frac{1}{8} (e^{\frac{\mu}{2}} (4\mu - 1) - e^{-\frac{\mu}{2}} (4\mu + 1) - \sqrt{a}) \right), \\
\lambda_3 &= \frac{1}{\tau_{\max}} \left( -1 - \frac{1}{8} (e^{\frac{\mu}{2}} (4\mu - 1) - e^{-\frac{\mu}{2}} (4\mu + 1) + \sqrt{b}) \right), \\
\lambda_4 &= \frac{1}{\tau_{\max}} \left( -1 - \frac{1}{8} (e^{\frac{\mu}{2}} (4\mu - 1) - e^{-\frac{\mu}{2}} (4\mu + 1) - \sqrt{b}) \right),
\end{align*}
\]
here $\mu = \tau_{\text{max}}/h$ denotes the CFL number, and $a$ and $b$ are given by
\[
\begin{align*}
    a &= 42 - 32\mu^2 + (16\mu^2 + 24\mu - 3)e^{\alpha i} + (16\mu^2 - 24\mu - 3)e^{-\alpha i} + 96\mu(e^{\frac{\alpha i}{2}} - e^{-\frac{\alpha i}{2}}), \\
    b &= 42 - 32\mu^2 + (16\mu^2 + 24\mu - 3)e^{\alpha i} + (16\mu^2 - 24\mu - 3)e^{-\alpha i} - 96\mu(e^{\frac{\alpha i}{2}} - e^{-\frac{\alpha i}{2}}).
\end{align*}
\]

Because the above expressions of eigenvalues are more complicated, the special case of small $\alpha$ is only considered here. Using the Taylor expansions to (4.12) with respect to $\alpha$ gives
\[
\begin{align*}
    \lambda_1 &= \frac{1}{\tau_{\text{max}}} \left( -\frac{1}{2} + 2\mu \alpha i + \frac{4}{3} \mu^2 + \frac{1}{16} \alpha^2 \right) + O(\alpha^3), \\
    \lambda_2 &= \frac{1}{\tau_{\text{max}}} \left( -2 - \mu \alpha i - \frac{4}{3} \mu^2 \alpha^2 \right) + O(\alpha^3), \\
    \lambda_3 &= \frac{1}{\tau_{\text{max}}} \left( -\frac{3}{2} - \frac{1}{16} \alpha^2 \right) + O(\alpha^3), \\
    \lambda_4 &= -\frac{1}{\tau_{\text{max}}} \mu \alpha i + O(\alpha^3).
\end{align*}
\]

In the following, we discuss the stability of the fully discrete version of (4.11) with Runge-Kutta time discretizations and the time stepsize $\Delta t_n = \tau_{\text{max}}$. If the first-order accurate Euler method is employed, then the fully discrete scheme becomes
\[
    u_{n+1} = (I + \Delta t_n G) u_n. \tag{4.13}
\]

It is easy to get that for small $\alpha$, the inequality
\[
    |1 + \Delta t_n \lambda_2| = 1 + \frac{11}{6} \mu^2 \alpha^2 + O(\alpha^3) > 1,
\]
holds, thus the fully discrete scheme (4.13) is unstable. It is similar to the old $P^1$-based Runge-Kutta CDG methods.

If the second-order Runge-Kutta method
\[
\begin{align*}
    u^{(1)} &= u^n + \Delta t_n Gu^n, \\
    u_{n+1} &= \frac{1}{2} u^n + \frac{1}{2} (u^{(1)} + \Delta t_n Gu^{(1)}), \tag{4.14}
\end{align*}
\]
is used to discretize the time derivative, then the fully-discrete scheme may be formed as follows
\[
    u_{n+1} = \left( I + \Delta t_n G + \frac{1}{2} \Delta t_n^2 G^2 \right) u^n. \tag{4.15}
\]
If the inequality
\[
    \varphi(\mu) := \max_{1 \leq i \leq 4} \left| 1 + \Delta t_n \lambda_i + \frac{\Delta t_n^2 \lambda_i^2}{2} \right| \leq 1,
\]
Table 3: Numerically estimated maximum CFL numbers, where $K$ denotes the degree of polynomial basis, and $\nu$ is the order of Runge-Kutta method.

<table>
<thead>
<tr>
<th></th>
<th>non-central DG</th>
<th>old CDG</th>
<th>new CDG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$\nu = 2$</td>
<td>0.333</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\nu = 3$</td>
<td>0.409</td>
<td>0.209</td>
<td>0.130</td>
</tr>
<tr>
<td>$\nu = 4$</td>
<td>0.464</td>
<td>0.235</td>
<td>0.145</td>
</tr>
</tbody>
</table>

holds, then the scheme (4.15) is stable. However, when $\alpha$ is smaller, the previous analysis tells us that

$$
q(\mu) \geq 1 + \Delta t_n^2 \lambda_2 + \frac{\Delta t_n^4 \lambda_2^2}{2} = 1 + \frac{4}{3} \mu^2 \alpha^2 + O(\alpha^3) > 1.
$$

It means that for any small $\mu$, the method (4.15) is unstable, but the old $P_1$-based Runge-Kutta CDG methods with second-order accurate Runge-Kutta methods (4.14) are stable under the certain CFL condition.

If the higher-order Runge-Kutta time discretization or $K > 1$, then it is difficult to analyze analytically its stability. For this reason, the CFL numbers are numerically estimated. Table 3 lists the admissible maximum CFL numbers of new Runge-Kutta CDG methods (4.9) with $\nu$th order Runge-Kutta method, and old Runge-Kutta CDG methods as well as RKDG methods. It is seen that the CFL numbers of new methods are smaller than the old, especially for the $P_2$-based Runge-Kutta CDG methods, but the difference between the new and old $P_3$-based methods becomes small. Thus we may expect that the new $P_3$-based method is likely to improve the computational efficiency.

**Remark 4.1.** The CFL number of Runge-Kutta CDG methods is dependent on the size of $\theta = \Delta t_n / \tau_{\text{max}}$. In general, if $\theta$ is smaller, the CFL number may become bigger. For example, if $\theta = 0.3$ and the third-order explicit Runge-Kutta time discretization is employed, then the CFL number of old $P_1, P_2, P_3$-based Runge-Kutta CDG methods may be taken as 2.57, 1.5, and 1, respectively.

**Remark 4.2.** The maximum CFL number of the old $P_1$-based Runge-Kutta CDG methods with second-order explicit Runge-Kutta time discretization is about 0.439, which is lesser than that in [29]. Moreover, our numerical experiments show that when the CFL number $\mu = 0.44$, the old $P_1$-based Runge-Kutta CDG methods with second-order explicit Runge-Kutta time discretization becomes unstable in solving (4.2) because $q \approx 1.00014$.

**Remark 4.3.** It is worth mentioning that the CFL number of new $P_1$-based Runge-Kutta CDG methods with fourth-order Runge-Kutta time discretization is lesser than with the third-order Runge-Kutta method. This situation is not too common.

In order to demonstrate the accuracy of new methods and further compare them to the old, the Runge-Kutta CDG methods are used to solve the initial-boundary value prob-
Two-dimensional Burgers equation

\[ u_t + \left( \frac{u^2}{2} \right)_x + \left( \frac{u^2}{2} \right)_y = 0, \]  

(4.16)

with the initial data \( u(x,y,0) = 0.5 + \sin \left( \pi (x+y)/2 \right) \), the computational domain \([0,4] \times [0,4]\), and the periodic boundary conditions. Table 4 gives the errors and orders at \( t = 0.5/\pi \) obtained by using the new and old Runge-Kutta CDG methods with or without limiter in global. Up to the output time \( t = 0.5/\pi \), the solution is still smooth. Those data show that two kinds of \( p^K \)-based Runge-Kutta CDG methods may achieve the theoretical order \( K+1 \), and the global use of WENO limiter may keep the accuracy of Runge-Kutta CDG methods. Table 5 presents the CPU times for two kinds of \( p^K \)-based Runge-Kutta CDG methods. It is seen that for the scalar equation, the advantage of new methods is not obvious in comparison to the old, but we may expect that the new methods are faster than the old in solving the RHD equations.

Table 4: The \( l^1 \) errors and orders \( t = 0.5/\pi \) of the new and old \( p^K \)-based Runge-Kutta CDG methods for the Burgers equation (4.16). The fourth order Runge-Kutta time discretization and \( N \times N \) cells are used.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( N )</th>
<th>( l^1 ) error</th>
<th>order</th>
<th>( l^1 ) error</th>
<th>order</th>
<th>( l^1 ) error</th>
<th>order</th>
<th>( l^1 ) error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p^1 )</td>
<td>10</td>
<td>4.60e-01</td>
<td>–</td>
<td>4.74e-01</td>
<td>–</td>
<td>1.64e-00</td>
<td>–</td>
<td>1.16e+00</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.10e-01</td>
<td>2.06</td>
<td>1.13e-01</td>
<td>2.07</td>
<td>4.82e-01</td>
<td>1.77</td>
<td>3.07e-01</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>2.80e-02</td>
<td>1.98</td>
<td>2.85e-02</td>
<td>1.98</td>
<td>1.18e-01</td>
<td>2.03</td>
<td>6.93e-02</td>
<td>2.15</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6.97e-03</td>
<td>2.00</td>
<td>7.09e-03</td>
<td>2.01</td>
<td>3.09e-02</td>
<td>1.94</td>
<td>1.60e-02</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>1.75e-03</td>
<td>2.00</td>
<td>1.77e-03</td>
<td>2.00</td>
<td>7.94e-03</td>
<td>1.96</td>
<td>4.10e-03</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>4.36e-04</td>
<td>2.00</td>
<td>4.43e-04</td>
<td>2.00</td>
<td>1.99e-03</td>
<td>2.00</td>
<td>1.04e-03</td>
<td>1.98</td>
</tr>
<tr>
<td>( p^2 )</td>
<td>10</td>
<td>8.13e-02</td>
<td>–</td>
<td>7.98e-02</td>
<td>–</td>
<td>4.40e-01</td>
<td>–</td>
<td>3.10e-01</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.18e-02</td>
<td>2.78</td>
<td>1.20e-02</td>
<td>2.77</td>
<td>6.02e-02</td>
<td>2.87</td>
<td>3.95e-02</td>
<td>2.97</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1.41e-03</td>
<td>3.07</td>
<td>1.52e-03</td>
<td>2.98</td>
<td>5.87e-03</td>
<td>3.41</td>
<td>3.12e-03</td>
<td>3.67</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1.74e-04</td>
<td>3.02</td>
<td>1.91e-04</td>
<td>2.99</td>
<td>3.19e-04</td>
<td>4.15</td>
<td>2.15e-04</td>
<td>3.86</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>2.16e-05</td>
<td>3.01</td>
<td>2.40e-05</td>
<td>2.99</td>
<td>2.74e-05</td>
<td>3.54</td>
<td>2.28e-05</td>
<td>3.24</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>2.69e-06</td>
<td>3.00</td>
<td>3.01e-06</td>
<td>3.00</td>
<td>3.11e-06</td>
<td>3.14</td>
<td>2.74e-06</td>
<td>3.06</td>
</tr>
<tr>
<td>( p^3 )</td>
<td>10</td>
<td>3.25e-02</td>
<td>–</td>
<td>2.95e-02</td>
<td>–</td>
<td>3.45e-01</td>
<td>–</td>
<td>2.44e-01</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.84e-03</td>
<td>4.15</td>
<td>1.80e-03</td>
<td>4.04</td>
<td>3.04e-02</td>
<td>3.51</td>
<td>2.04e-02</td>
<td>3.58</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1.26e-04</td>
<td>3.87</td>
<td>1.28e-04</td>
<td>3.81</td>
<td>9.61e-04</td>
<td>4.98</td>
<td>5.89e-04</td>
<td>5.12</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>7.75e-06</td>
<td>4.02</td>
<td>8.40e-06</td>
<td>3.94</td>
<td>1.43e-05</td>
<td>6.07</td>
<td>9.74e-06</td>
<td>5.92</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>4.84e-07</td>
<td>4.00</td>
<td>5.40e-07</td>
<td>3.96</td>
<td>4.18e-07</td>
<td>5.10</td>
<td>3.90e-07</td>
<td>4.64</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>3.03e-08</td>
<td>4.00</td>
<td>3.43e-08</td>
<td>3.98</td>
<td>2.39e-08</td>
<td>4.12</td>
<td>2.40e-08</td>
<td>4.02</td>
</tr>
</tbody>
</table>
Table 5: CPU times (second) for new and old Runge-Kutta CDG methods solving the initial-boundary problem of Burgers equation (4.16). 320 × 320 cells.

<table>
<thead>
<tr>
<th></th>
<th>without limiter</th>
<th>with limiter in global</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>new</td>
<td>old</td>
</tr>
<tr>
<td>$P^1$</td>
<td>130.3</td>
<td>88.8</td>
</tr>
<tr>
<td>$P^2$</td>
<td>878.5</td>
<td>585.4</td>
</tr>
<tr>
<td>$P^3$</td>
<td>2590.5</td>
<td>2560.7</td>
</tr>
</tbody>
</table>

5 Numerical results

This section uses our $P^K$-based Runge-Kutta CDG methods with WENO limiter presented in the last section, $K = 1, 2, 3$, to solve several initial value problems or initial-boundary-value problems of one- and two-dimensional RHD equations in order to demonstrate the accuracy and effectiveness of Runge-Kutta CDG methods. The Runge-Kutta CDG methods will be compared to the RKDG methods. Moreover, because the solutions of Runge-Kutta CDG methods on two mutually dual meshes are almost identical each other, only the solution on one mesh $\{C_j\}$ or $\{\tilde{C}_{jk}\}$ will be shown in the following.

5.1 1D case

For the 1D computations, the uniform mesh is used, that is, the spatial step size $h_{j+\frac{1}{2}}$ is constant. The CFL numbers $\mu$ of $P^1$-, $P^2$-, $P^3$-based Runge-Kutta CDG methods are taken as 0.4, 0.3, 0.2, respectively, and $\theta = \Delta t_n/\tau_n = 1$. Unless otherwise stated, $M = 50$ is used in the TVB modified minmod function and the third-order accurate TVD Runge-Kutta (3.8) is employed and the time step size is determined by

$$\Delta t_n = \theta \tau_n = \frac{\theta \mu h_{j+\frac{1}{2}}}{\max_{i,j} \{ |\lambda^{(i)}(U_{j})|, |\lambda^{(i)}(U_{j+\frac{1}{2}})| \}},$$

where the eigenvalues $\lambda^{(i)}(U)$ may be found in [62].

Example 5.1 (Riemann Problem 1). The initial data are

$$(\rho, v_1, p)(x,0) = \begin{cases} (1,0.9,1), & x < 0.5, \\ (1,0,10), & x > 0.5, \end{cases}$$

and $\Gamma = 4/3$. As the time increases, the initial discontinuity will be decomposed into a slowly left-moving shock wave, a contact discontinuity, and a right-moving shock wave.

Fig. 5 presents the densities at $t=0.4$ calculated by using the Runge-Kutta CDG methods and RKDG methods. As can be seen from those plots, the numerical solutions of
Figure 5: Example 5.1: the densities $\rho$ at $t = 0.4$. The symbol "○" denotes numerical solution with 200 cells, while the solid line is exact solution. Left: $P^K$-based Runge-Kutta CDG methods; right: $P^K$-based RKDG methods. From top to bottom: $K = 1, 2, 3$.

Runge-Kutta CDG methods and RKDG methods are in good agreement with the exact solutions, but there exist obvious oscillations in the densities behind the left-moving shock wave obtained by using the $P^2$- and $P^3$-based RKDG methods, while no obvious oscillation is observed in the densities obtained by Runge-Kutta CDG methods. Such phenomenon is also observed in the velocities and pressures, see Figs. 6 and 7. The
“troubled” cells identified by the RKDG methods is more than the Runge-Kutta CDG methods, see Fig. 8.

Example 5.2 (Riemann problem 2). The initial data of second Riemann problem is

\[(\rho, v_1, p)(x,0) = \begin{cases} 
(10,0,0,1000), & x < 0.5, \\
(1,0,0,0.01), & x > 0.5,
\end{cases}\]
and $\Gamma = 5/3$. The solution of this problem will contain a left-moving rarefaction wave, a contact discontinuity, and a right-moving shock wave as $t > 0$. The speed of the contact is almost identical to the shock wave so that it is much more challenging for the numerical methods than the first Riemann problem.

The Runge-Kutta CDG methods with new and old calculations of the flux integral over the cell are considered here. The maximum CFL numbers are taken as those in Table 3. Figs. 9 and 10 display the solutions at $t = 0.4$ obtained by new and old Runge-Kutta
CDG methods with 200 cells. The density obtained by the old Runge-Kutta CDG methods is slightly better than the new. The CPU times for them with 800 cells are estimated in Table 6. The data show that the new $P^3$-based CDG method is faster than the old, there is no obvious difference between two $P^1$-based CDG method, but the new $P^2$-based CDG method is slower than the old due to a relatively harsh stability condition for the new $P^2$-based CDG method.
Figure 9: Example 5.2: The densities $\rho$ (left) and velocities $v_1$ (right) at $t = 0.4$. The solid line denotes the exact solution, while the symbol "\textcircled{0}" and "\texttimes" are the solutions obtained by using the new and old Runge-Kutta CDG methods with 200 uniform cells. From top to bottom: $K = 1, 2, 3$.

5.2 2D case

This section solves some 2D RHD problems by using Runge-Kutta CDG methods on the uniform rectangular meshes. Those problems are the two-dimensional smooth problem, Riemann problems, and shock-bubble interaction problem. The spatial stepsizes in the $x$ and $y$ directions are denoted by $h_x$ and $h_y$ respectively. Unless otherwise stated, only the
Figure 10: Same as Fig. 9 except for the pressure \( p \) (left) and specific internal energy \( e \) (right).

Table 6: Example 5.2: CPU times (second) of new and old Runge-Kutta CDG methods with 800 cells.

<table>
<thead>
<tr>
<th></th>
<th>new method</th>
<th>old method</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p^1 )</td>
<td>15.7</td>
<td>15.3</td>
</tr>
<tr>
<td>( p^2 )</td>
<td>55.3</td>
<td>38.4</td>
</tr>
<tr>
<td>( p^3 )</td>
<td>74.2</td>
<td>81.7</td>
</tr>
</tbody>
</table>
third-order accurate TVD Runge-Kutta time discretization (3.8) is employed and the time step size is taken as

\[
\Delta t_n = \theta \tau_n = \frac{\theta \mu}{\max_{j,k} \left( \frac{\max_i |\lambda_j^{(1)}(U^{(0)})|}{h_x} + \frac{\max_i |\lambda_j^{(2)}(U^{(0)})|}{h_y} \right)},
\]

where \( \mu \) denotes the CFL number, and “\( \max_{j,k} \)” denotes the maximum value over the cells \( C_{j,k} \) and \( D_{j+\frac{1}{2},k+\frac{1}{2}} \), while the values of \( \mu \) and \( \theta \) will be given in the coming examples.

**Example 5.3** (Smooth problem). This smooth problem has been used in [62] to test the accuracy of numerical methods. The initial data for the primitive variables \( V \) is set as

\[
V(x,y,0) = \begin{cases} 
(1 + 0.2 \sin(2\pi(x \cos \alpha + y \sin \alpha)), 0, 2, 0, 1) \\
(0.035145216124503, 0, 0, 0.162931056509027), & x > 0, \ y > 0, \\
(0.1, 0.7, 0, 1), & x < 0, \ y > 0, \\
(0.5, 0, 0.1), & x < 0, \ y < 0, \\
(0.1, 0, 0.7, 1), & x > 0, \ y < 0,
\end{cases}
\]

where \( \alpha = 30^\circ \) denotes the angle of the sine wave propagation direction relative to the \( x \)-axis. The computational domain \( \Omega = [0, 2/\sqrt{3}] \times [0, 2] \) is specified with the periodic boundary conditions, and divided into \( N \times 2N \) uniform cells.

Table 7 lists the \( l^1 \) errors of density and orders at \( t = 1 \) obtained by using the Runge-Kutta CDG methods, where the fourth-order accurate Runge-Kutta time discretization is employed, \( \theta = 1 \), and the CFL number \( \mu \) is taken as 0.3, 0.25, and 0.2 for \( P^1 \)-, \( P^2 \)-, and \( P^3 \)-based methods, respectively. Those results show that the theoretical order \( K+1 \) of the \( P^K \)-based Runge-Kutta CDG methods may be achieved.

**Example 5.4** (Riemann problem 1). The initial data of the first 2D Riemann problem are

\[
(\rho, v_1, v_2, p)(x, y, 0) = \begin{cases} 
(0.035145216124503, 0, 0, 0, 0.162931056509027), & x > 0, \ y > 0, \\
(0.1, 0.7, 0, 1), & x < 0, \ y > 0, \\
(0.5, 0, 0.1), & x < 0, \ y < 0, \\
(0.1, 0, 0.7, 1), & x > 0, \ y < 0,
\end{cases}
\]

where the left and bottom discontinuities are two contact discontinuities and the top and right are two shock waves with the speed of 0.934563275373844.

In our computations, \( \theta \) is taken as 1 or 0.5, and the value of \( \mu \theta \) is fixed as 0.3, 0.25, and 0.2 for the \( P^1 \)-, \( P^2 \)-, \( P^3 \)-based Runge-Kutta CDG methods, respectively. The results at \( t = 0.8 \) obtained by the \( P^K \)-based Runge-Kutta CDG methods are presented in Figs. 11, 12, and 13. Fig. 14 gives the density at \( t = 0.8 \) along the line \( y = x \). Table 8 shows the percentage of “troubled” cells. It is seen that the resolution of \( P^K \)-based RKDG methods is better than \( P^K \)-based Runge-Kutta CDG methods, when \( \mu \theta \) is fixed, and the Runge-Kutta CDG methods with small \( \theta \) improve the resolution of the discontinuity better than the case of big \( \theta \), especially for the \( P^1 \)-based CDG method.
Table 7: Example 5.3: $l^1$ errors of the density and orders $t = 1$ of the $p^k$-based Runge-Kutta CDG methods with $N \times 2N$ cells.

<table>
<thead>
<tr>
<th>$N$</th>
<th>without limiter</th>
<th>with limiter in global</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l^1$ error</td>
<td>order</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>10</td>
<td>9.09e-03</td>
<td>–</td>
</tr>
<tr>
<td>20</td>
<td>1.28e-03</td>
<td>2.83</td>
</tr>
<tr>
<td>40</td>
<td>3.02e-04</td>
<td>2.08</td>
</tr>
<tr>
<td>80</td>
<td>7.56e-05</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>1.89e-05</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>4.72e-06</td>
<td>2.00</td>
</tr>
</tbody>
</table>

$P^1$

<table>
<thead>
<tr>
<th>$N$</th>
<th>without limiter</th>
<th>with limiter in global</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l^1$ error</td>
<td>order</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>10</td>
<td>3.43e-04</td>
<td>–</td>
</tr>
<tr>
<td>20</td>
<td>4.24e-05</td>
<td>3.02</td>
</tr>
<tr>
<td>40</td>
<td>5.28e-06</td>
<td>3.01</td>
</tr>
<tr>
<td>80</td>
<td>6.59e-07</td>
<td>3.00</td>
</tr>
<tr>
<td>160</td>
<td>8.23e-08</td>
<td>3.00</td>
</tr>
<tr>
<td>320</td>
<td>1.03e-08</td>
<td>3.00</td>
</tr>
</tbody>
</table>

$P^2$

<table>
<thead>
<tr>
<th>$N$</th>
<th>without limiter</th>
<th>with limiter in global</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l^1$ error</td>
<td>order</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>10</td>
<td>2.53e-05</td>
<td>–</td>
</tr>
<tr>
<td>20</td>
<td>1.55e-06</td>
<td>4.03</td>
</tr>
<tr>
<td>40</td>
<td>9.61e-08</td>
<td>4.01</td>
</tr>
<tr>
<td>80</td>
<td>5.99e-09</td>
<td>4.00</td>
</tr>
<tr>
<td>160</td>
<td>3.75e-10</td>
<td>4.00</td>
</tr>
<tr>
<td>320</td>
<td>2.34e-11</td>
<td>4.00</td>
</tr>
</tbody>
</table>

$P^3$

Table 8: Example 5.4: The percentage of "troubled" cells at $t = 0.8$.

<table>
<thead>
<tr>
<th>non-central DG</th>
<th>CDG $\theta = 1$</th>
<th>CDG $\theta = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^1$</td>
<td>0.19</td>
<td>0.04</td>
</tr>
<tr>
<td>$P^2$</td>
<td>5.51</td>
<td>4.78</td>
</tr>
<tr>
<td>$P^3$</td>
<td>10.68</td>
<td>8.14</td>
</tr>
</tbody>
</table>

Table 9: Example 5.4: CPU times (second) of the CDG and non-central DG methods.

<table>
<thead>
<tr>
<th></th>
<th>CDG</th>
<th>non-central DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^1$</td>
<td>5728.7</td>
<td>2491.5</td>
</tr>
<tr>
<td>$P^2$</td>
<td>16722.4</td>
<td>6119.8</td>
</tr>
<tr>
<td>$P^3$</td>
<td>54490.8</td>
<td>19287.8</td>
</tr>
</tbody>
</table>
Figure 11: Example 5.4: The contour plots of density logarithm $\log \rho$ at (30 equally spaced contour lines from $-1.46$ to $-0.18$) and the “troubled” cells $t = 0.8$ obtained with $300 \times 300$ cells. Top: $P^1$-based DG, middle: $P^1$-based CDG with $\theta = 1$, bottom: $P^1$-based CDG with $\theta = 0.5$. 
Figure 12: Same as Fig. 11 except for the $P^2$-based Runge-Kutta CDG methods.
Figure 13: Same as Fig. 11 except for the $P^3$-based Runge-Kutta CDG methods.
Example 5.5 (Riemann problem 2). It is about the interaction of four contact discontinuities (vortex sheets) with the same sign (the negative sign) for the ideal relativistic fluid. The initial data may be found in [62]. The new and old Runge-Kutta CDG methods are used to solve this problem. In order to compare the CPU times, the maximum CFL numbers in Table 3 and $\theta = 1$ are considered for them. Fig. 15 displays the contours of density at $t = 0.8$. It is seen that the resolutions of the new $P^1$- and $P^2$-based Runge-Kutta CDG methods are slightly worse than the old, but the difference between two $P^3$-based methods are not obvious. Their CPU times listed in Table 10 show that for the $P^2$ case, the new method is slower than the old, but the new $P^3$-based method has a great advantage. It is mainly because the CFL number of the new $P^2$-based method is about half of that of the old, while the CFL number of the new $P^3$-based method is about two-thirds of the old.

<table>
<thead>
<tr>
<th></th>
<th>new method</th>
<th>old method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^1$</td>
<td>2324.5</td>
<td>2460.3</td>
</tr>
<tr>
<td>$P^2$</td>
<td>14210.2</td>
<td>13611.9</td>
</tr>
<tr>
<td>$P^3$</td>
<td>30702.3</td>
<td>48538.7</td>
</tr>
</tbody>
</table>

Example 5.6 (Shock and light bubble interaction). This example describes the interaction between the shock wave and a light bubble. The setup of the problem is as follows. Initially, within the computational domain $[0,325] \times [-45,45]$, there is a left-moving shock wave at $x = 265$ with the left and right states

$$(\rho, v_1, v_2, p) = \begin{cases} (1,0,0,0.05), & x < 265, \\ (1.865225080631180, -0.196781107378299, 0, 0.15), & x > 265. \end{cases}$$

A cylindrical bubble is centered at $(215,0)$ with the radius of 25 in front of the initial shock wave. The fluid within the bubble is in a mechanical equilibrium with the surrounding
Figure 15: Example 5.5: The contour plots of density logarithm $\log \rho$ at $t = 0.8$ (30 equally spaced contour lines from $-1.98$ to $0.56$) obtained by using $p^K$-based Runge-Kutta CDG methods with $300 \times 300$ cells. Left: the old, right: the new. From top to bottom, $K = 1, 2, 3$. 
fluid and lighter than the ambient fluid. The detailed state of the fluid within the bubble is taken as \((\rho, v_1, v_2, p) = (0.1358, 0, 0, 0.05)\). In our computations, the domain is divided into \(500 \times 140\) uniform cells, the CFL numbers \(\mu\) for the \(P^1_{\bullet}\), \(P^2_{\bullet}\), \(P^3_{\bullet}\)-based Runge-Kutta CDG methods are taken as 0.6, 0.5, 0.4, respectively, and \(\theta = 0.5\). Moreover, the parameter \(M\) in the modified TVB minmod function is taken as 0.005, the reflective boundaries are specified at \(y = \pm 45\), and the fluid states on two boundaries in \(x\)-direction are set to the left and right shock wave states, respectively.

Figs. 16 and 17 show the schlieren images of density and the distributions of “troubled” cells at \(t = 270\), while Figs. 18 and 19 give corresponding results at \(t = 450\). It is seen

![Figure 16](image1.png)

Figure 16: Example 5.6: The schlieren images of density at \(t = 270\) with \(500 \times 140\) cells. Left: \(P^K_{\bullet}\)-based Runge-Kutta CDG methods; right: \(P^K_{\bullet}\)-based RKDG methods. From top to bottom, \(K = 1, 2, 3\).

![Figure 17](image2.png)

Figure 17: Same as Fig. 16 except for the “troubled” cells at \(t = 270\).
from those plots that the $p^k$-based RKDG methods resolve the complex wave structure better than the $p^k$-based Runge-Kutta CDG methods, $p^3$-based RKDG methods gives relatively fine wave structure, the distribution of “troubled” cells is very consistent with the solutions, and the “troubled” cell proportions are almost similar for the same order of the two types of DG methods. Table 11 gives the percentage of “troubled” cells at two different times. The CPU times are estimated in Table 12 and show that the $p^1$- and $p^3$-based Runge-Kutta CDG methods are faster than corresponding RKDG methods, but the $p^2$-based method is an exception.
6 Conclusions

It is much more difficult to solve the relativistic hydrodynamical (RHD) equations than the non-relativistic case. The appearance of Lorentz factor enhances the nonlinearity of the RHD equations, the fluxes cannot be formulated in an explicit form of the conservative vector, and there are some inherent physical constraints on the physical state. In practical computations of the RHD system, the primitive variable vector has to be first recovered from the known conservative vector by iteratively solving a nonlinear pressure equation and then the fluxes are evaluated at each time step.

The paper developed the $p^K$-based Runge-Kutta CDG methods for the one- and two-dimensional special RHD equations, $K = 1, 2, 3$. In comparison to RKDG methods, the Runge-Kutta CDG methods found two approximate solutions defined on mutually dual meshes. For each mesh, the CDG approximate solutions on its dual mesh were used to calculate the flux values in the cell and on the cell boundary so that the approximate solutions on two mutually dual meshes were coupled with each other, and the use of numerical flux might be avoided. In addition, the Runge-Kutta CDG methods allowed the use of a larger CFL number.

The WENO limiter was adaptively implemented via two steps: the “troubled” cells were first identified by using a modified TVB minmod function, and then the WENO technique is used to locally reconstruct new polynomials of degree $(2K + 1)$ replacing the CDG solutions inside the “troubled” cells by using the cell average values of the CDG solutions in the neighboring cells as well as the original cell averages of the “troubled” cells.

The accuracy of the CDG without the numerical dissipation was analyzed and calculation of the flux integrals over the cells was also discussed. Because the DG approximate solutions were discontinuous at the cell interface in general, the integrals over each cell

Table 11: Example 5.6: The percentage of “troubled” cells at two different times.

<table>
<thead>
<tr>
<th></th>
<th>$t = 270$</th>
<th></th>
<th>$t = 450$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CDG</td>
<td>non-central DG</td>
<td>CDG</td>
<td>non-central DG</td>
</tr>
<tr>
<td>$p^1$</td>
<td>0.68</td>
<td>0.22</td>
<td>0.71</td>
<td>0.33</td>
</tr>
<tr>
<td>$p^2$</td>
<td>3.12</td>
<td>3.58</td>
<td>3.40</td>
<td>3.56</td>
</tr>
<tr>
<td>$p^3$</td>
<td>5.61</td>
<td>5.94</td>
<td>7.11</td>
<td>7.39</td>
</tr>
</tbody>
</table>

Table 12: Example 5.6: CPU times (second) of the CDG and non-central DG methods.

<table>
<thead>
<tr>
<th></th>
<th>CDG</th>
<th>non-central DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p^1$</td>
<td>1.93e4</td>
<td>4.03e3</td>
</tr>
<tr>
<td>$p^2$</td>
<td>6.17e4</td>
<td>1.14e4</td>
</tr>
<tr>
<td>$p^3$</td>
<td>1.73e5</td>
<td>3.47e4</td>
</tr>
</tbody>
</table>
of the DG approximate solutions defined on corresponding dual meshes became a sum of several integrals over subcell of the DG polynomial solutions on corresponding dual meshes, which would lead to that more numerical integration points are needed to ensure the accuracy of Runge-Kutta CDG methods. An attempt was made that only approximate solution on the dual mesh was used to evaluate the flux on the cell boundary for the DG approximations of RHD system on the mesh, thus the integrals of DG solutions over the dual cell might be avoided and the cost of numerical integration was hopefully reduced. For the linear scalar equation, the Fourier method and numerical experiments were used to analyze the stability of such new method, and estimate the CFL numbers for the stability.

Several numerical experiments demonstrated the accuracy, robustness, and discontinuity resolution of our methods. The results showed that the Runge-Kutta CDG methods with WENO limiter were robust and could capture the contact discontinuities, shock waves, and other complex wave structures well, the WENO limiter was only implemented in a few “troubled” cells, the Runge-Kutta CDG methods had obvious advantages in simulating the propagation of slow shock wave in comparison to the RKDG methods, see Example 5.1. Moreover, the new two-dimensional $P^3$-based Runge-Kutta CDG methods could more significantly improve the computational efficiency than the old. In solving RHD problems with large Lorentz factor, or strong discontinuities, or low rest-mass density or pressure etc., it is still very challenging for the present $P^K$-based Runge-Kutta CDG methods. To cure such difficulty, the $P^0$-based method may be locally used to replace the $P^K$-based. The genuinely effective way is to employ the physical-constraints preserving methods, see e.g. [50,52].

**Acknowledgments**

This work was partially supported by the Special Project on High-performance Computing under the National Key R&D Program (No. 2016YFB0200603), Science Challenge Project No. JCKY2016212A502, and the National Natural Science Foundation of China (No. 91330205, 91630310, & 11421101).

**References**


