NUMERICAL MATHEMATICS: Theory, Methods and Applications
Numer. Math. Theor. Meth. Appl., Vol. **1**, No. 4, pp. 435-459 (2008)

Development and Comparison of Numerical Fluxes for LWDG Methods

Jianxian Qiu[*]

Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, China.

Received 12 October, 2007; Accepted (in revised version) 7 February, 2008

Abstract. The discontinuous Galerkin (DG) or local discontinuous Galerkin (LDG) method is a spatial discretization procedure for convection-diffusion equations, which employs useful features from high resolution finite volume schemes, such as the exact or approximate Riemann solvers serving as numerical fluxes and limiters. The Lax-Wendroff time discretization procedure is an alternative method for time discretization to the popular total variation diminishing (TVD) Runge-Kutta time discretizations. In this paper, we develop fluxes for the method of DG with Lax-Wendroff time discretization procedure (LWDG) based on different numerical fluxes for finite volume or finite difference schemes, including the first-order monotone fluxes such as the Lax-Friedrichs flux, Godunov flux, the Engquist-Osher flux etc. and the second-order TVD fluxes. We systematically investigate the performance of the LWDG methods based on these different numerical fluxes for convection terms with the objective of obtaining better performance by choosing suitable numerical fluxes. The detailed numerical study is mainly performed for the one-dimensional system case, addressing the issues of CPU cost, accuracy, non-oscillatory property, and resolution of discontinuities. Numerical tests are also performed for two dimensional systems.

AMS subject classifications: 65M60, 65M99, 35L65

Key words: Discontinuous Galerkin method, Lax-Wendroff type time discretization, numerical flux, approximate Riemann solver, limiter, WENO scheme, high order accuracy.

1. Introduction

In this paper, we develop fluxes for the method of DG with Lax-Wendroff time discretization procedure (LWDG) based on different numerical fluxes for finite volume or finite difference schemes, including the first-order monotone fluxes such as the Lax-Friedrichs flux, Godunov flux, the Engquist-Osher flux etc. and the second-order TVD fluxes, and investigate the performance of the LWDG method based on different numerical fluxes for convection terms for solving nonlinear convection-diffusion scalar equations or systems:

$$\begin{cases} u_t + \nabla \cdot f(u) = \nabla \cdot fd(u, \nabla \cdot u), \\ u(x,0) = u_0(x), \end{cases} \tag{1.1}$$

[*]Corresponding author. *Email address:* `jxqiu@nju.edu.cn` (J. Qiu)

where f and fd are convection and diffusion terms, respectively, with the objective of obtaining better performance by choosing suitable numerical fluxes.

The discontinuous Galerkin (DG) method [3–7] for solving hyperbolic conservation laws and its extension to time-dependent convection-diffusion equations, the local DG (LDG) methods [1, 8, 9] are high order finite element methods employing the useful features from high resolution finite volume schemes, such as the exact or approximate Riemann solvers, and total variation bounded (TVB) limiters [26].

DG or LDG method is a spatial discretization procedure, namely, it is a procedure to approximate the spatial derivative terms in (1.1). The time derivative term can be discretized by explicit, nonlinearly stable high order Runge-Kutta time discretizations [25,27], and the scheme is termed as RKDG or RKLDG scheme, respectively. An alternative approach could be using a Lax-Wendroff type time discretization procedure, which is also called the Taylor type referring to a Taylor expansion in time or the Cauchy-Kowalewski type referring to the similar Cauchy-Kowalewski procedure in partial differential equations (PDEs) [28]. This approach is based on the idea of the classical Lax-Wendroff scheme [17], and it relies on converting all the time derivatives in a temporal Taylor expansion into spatial derivatives by repeatedly using the PDE and its differentiated versions. The spatial derivatives are then discretized by, e.g. the DG approximations. The methods are termed as LWDG methods for conservation laws [20]. Lax-Wendroff type time discretization procedure is also used by Dumbser and Munz [10, 11], in which they developed the ADER (Arbitrary high order schemes using DERivatives, see [29]) discontinuous Galerkin method using generalized Riemann solvers [29]. ADER methods also use the Lax-Wendroff procedure to convert time derivatives to spatial derivatives. The Lax-Wendroff type time discretization was also used in high order finite volume schemes [14, 29] and finite difference schemes [22].

As pointed out in [20], the LWDG is a one step, explicit, high order finite element method, the limiter is performed once every time step. As a result, LWDG is more compact than RKDG and the Lax-Wendroff time discretization procedure is more cost effective than the Runge-Kutta time discretizations for certain problems including two dimensional Euler systems of compressible gas dynamics when nonlinear limiters are applied.

An important component of the DG methods for solving conservation laws (1.1) is the numerical flux, based on exact or approximate Riemann solvers, which is borrowed from finite difference and finite volume methodologies. In most of the DG papers in the literature, the two-point, first order monotone Lax-Friedrichs (LF) numerical flux is used due to its simplicity. However, there exist many other numerical fluxes based on various approximate Riemann solvers in the literature, such as other two-point, first order monotone fluxes and essentially two-point TVD flux, which could also be used in the context of DG methods. The local LF (LLF) numerical flux, the Godunov flux [13], the Engquist-Osher (EO) flux [12, 18] for the scalar case and its extension to systems (often referred to as the Osher-Solomon flux [18]), the HLL flux [15] and a modification of the HLL flux, often referred to as the HLLC flux [31] are based on the approximate Riemann solver, these fluxes are two-point, first order monotone fluxes. One of the essentially two-point TVD fluxes is the flux limiter centered (FLIC) flux [30] with the following *essentially two-point* property: $\hat{f}(u^l, u, u, u^r) = f(u)$ for any u^l and u^r, which combines a low order monotone flux and a high order flux with a flux limiter to guarantee the TVD property. The other fluxes such as generalized Riemann solvers [2,29] can also be used as numerical flux for DG methods.

In [20], the simple Lax-Friedrichs flux is used to design the LWDG schemes for conservation laws, followed the procedure of [20], in [24] we systematically studied and compared the performance of the LWDG method based on different numerical fluxes for conservation laws. But when the procedure of [20] is extended to convection-diffusion equations, the procedure to convert all time derivatives to spatial derivatives is becoming very complexity. In order to overcome this difficulty, in this paper, we develop a new LWDG scheme based on LDG [1,8,9] space discretization and Lax-Wendroff time discretization procedure in [22], in which we introduce the new intermediate variables for both time and spatial derivatives, and the procedure is more simply than the one based on the extension of [20]. The fluxes for the LWDG method based on different numerical fluxes for finite volume or finite difference schemes were developed, including the first-order monotone fluxes and the second-order TVD fluxes, then we investigate the performance of the LWDG method based on different numerical fluxes for solving nonlinear convection-diffusion equations.

We review and describe the details of LWDG methods and the fluxes under consideration in Section 2. We present extensive numerical experiments to compare their performance in Section 3. Concluding remarks are given in Section 4.

2. Description of the LWDG methods and the numerical fluxes

In this section, we describe the construction of the LWDG methods based on the procedure of LDG methods and Lax-Wendroff time discretization procedure adopted in [22] for convection-diffusion equations, and numerical fluxes for consideration. We start with the description of the LWDG method for convection-diffusion equations.

2.1. Description of LWDG

Consider the one dimensional convection-diffusion equations:

$$\begin{cases} u_t + f(u)_x = fd(u,u_x)_x, \\ u(x,0) = u_0(x). \end{cases} \tag{2.1}$$

We denote the cells by $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, the cell centers by $x_i = \frac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right)$ and the cell sizes by $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. Let Δt be the time step, $t^{n+1} = t^n + \Delta t$. We denote by $u^{(r)}$ the r-th order time derivative of u, namely $\frac{\partial^r u}{\partial t^r}$. We also use u', u'' and u''' to denote the first three time derivatives of u. By a temporal Taylor expansion we obtain

$$u(x,t^{n+1}) = \{u + \Delta t u' + \frac{\Delta t^2}{2}u'' + \frac{\Delta t^3}{6}u''' + \cdots\}(x,t^n). \tag{2.2}$$

If we would like to obtain $(k+1)$th order accuracy in time, we would need to approximate the first $k+1$ time derivatives: $u', \cdots, u^{(k+1)}$. We will proceed up to the third order in time in this paper, although the procedure can be naturally extended to any higher orders.

The DG solution as well as the test function space is given by

$$V_h^k = \{p : p|_{I_i} \in P^k(I_i), \forall i\},$$

where $P^k(I_i)$ is the space of polynomials of degree $\le k$ on the cell I_i. We adopt a local orthogonal basis over I_i, $\{v_l^{(i)}(x), l = 0,1,\cdots,k\}$, namely the scaled Legendre polynomials

$$v_0^{(i)}(x) = 1, \qquad v_1^{(i)}(x) = \frac{x - x_i}{\Delta x_i}, \qquad v_2^{(i)}(x) = \left(\frac{x - x_i}{\Delta x_i}\right)^2 - \frac{1}{12}, \cdots.$$

Other basis functions can be used as well, without changing the numerical method, since the finite element DG method depends only on the choice of space V_h^k, not on the choice of its basis functions.

The numerical solution $u^h(x,t)$ in the space V_h^k at $t = t^n$ can be written as:

$$u^h(x,t^n) = \sum_{l=0}^{k} u_i^{(l)}(t) v_l^{(i)}(x), \qquad \text{for } x \in I_i, \tag{2.3}$$

and the degrees of freedom $u_i^{(l)}(t^n)$ are the moments defined by

$$u_i^{(l)}(t^n) = \frac{1}{a_l} \int_{I_i} u^h(x,t^n) v_l^{(i)}(x) dx, \qquad l = 0,1,\cdots,k,$$

where $a_l = \int_{I_i} (v_l^{(i)}(x))^2 dx$ are the normalization constants since the basis is not orthonormal. In order to determine the approximate solution, we evolve the degrees of freedom $u_i^{(l)}$:

$$\begin{aligned} u_i^{(l)}(t^{n+1}) =& u_i^{(l)}(t^n) + \Delta t u_i^{\prime(l)}(t^n) + \frac{\Delta t^2}{2} u_i^{\prime\prime(l)}(t^n) \\ &+ \cdots + \frac{\Delta t^{k+1}}{(k+1)!} u_i^{(k+1)(l)}(t^n), \quad l = 0,1,\cdots,k, \end{aligned} \tag{2.4}$$

where $u_i^{\prime(l)}(t^n), u_i^{\prime\prime(l)}(t^n), \cdots, u_i^{(k+1)(l)}(t^n), l = 0,1,\cdots,k$ are moments of temporal derivative terms $u', u'', \cdots, u^{(k+1)}$, respectively. The moments of the temporal derivative terms in (2.4) can be replaced with the spatial ones using the governing equation (2.1) with following procedure:

Step 1. We introduce the new variable $q = u_x$, we have:

$$\begin{cases} u' = (-f(u) + fd(u,q))_x, \\ q = u_x. \end{cases} \tag{2.5}$$

The expression of u' and q in the space V_h^k can be written as:

$$u'(x,t^n) = \sum_{l=0}^{k} u_i^{\prime(l)}(t^n) v_l^{(i)}(x), \qquad \text{for } x \in I_i, \tag{2.6}$$

$$q(x,t^n) = \sum_{l=0}^{k} q_i^{(l)}(t^n) v_l^{(i)}(x), \qquad \text{for } x \in I_i. \tag{2.7}$$

Step 1.1. Reconstruction of $q=u_x$ in V_h^k. Multiply $q=u_x$ with a basis $v_i^{(l)}, l=0,1,\cdots,k$ of V_h^k, and integrate over the cell I_i, we obtain:

$$\begin{aligned} a_l q_i^{(l)} =& \int_{I_i} u_x v_l^{(i)}(x)dx = u(x_{i+1/2},t^n)v_l^{(i)}(x_{i+1/2}) \\ & - u(x_{i-1/2},t^n)v_l^{(i)}(x_{i-1/2}) - \int_{I_i} u \frac{dv_l^{(i)}(x)}{dx}dx. \end{aligned} \tag{2.8}$$

In order to determine the approximate solution, we replace $u(x_{i+1/2},t^n)$ with a numerical flux $\hat{u}_{i+1/2}$. In this paper, we choose

$$\hat{u}_{i+1/2} = u^h(x_{i+1/2}^-,t^n), \quad v_l^{(i)}(x_{i+1/2}) = v_l^{(i)}(x_{i+1/2}^-), \quad v_l^{(i)}(x_{i-1/2}) = v_l^{(i)}(x_{i-1/2}^+).$$

The integral in (2.8), can be integrated exactly when $u(x,t^n)$ is replaced by approximate solution $u^h(x,t^n)$.

Step 1.2. Reconstruction of $u' = (-f(u)+fd(u,q))_x$ in V_h^k. Multiply u' with a basis $v_i^{(l)}, l=0,1,\cdots,k$ of V_h^k, and integrate over the cell I_i, we obtain:

$$\begin{aligned} a_l u_i'^{(l)} =& \int_{I_i} (-f(u)+fd(u,q))_x v_l^{(i)}(x)dx \\ =& (-f(u)+fd(u,q))(x,t^n)v_l^{(i)}(x)\Big|_{x=x_{i-1/2}}^{x=x_{i+1/2}} \\ & - \int_{I_i} (-f(u)+fd(u,q))\frac{dv_l^{(i)}(x)}{dx}dx. \end{aligned} \tag{2.9}$$

In order to determine the approximate solution, we replace fluxes f and fd at $x = x_{i+1/2}$ with numerical fluxes $\hat{f}_{i+1/2}$ and $\widehat{fd}_{i+1/2}$, respectively. In this paper, we choose

$$\begin{aligned} &\widehat{fd}_{i+1/2} = fd(u^h(x_{i+1/2}^+,t^n), q(x_{i+1/2}^+,t^n)), \\ &v_l^{(i)}(x_{i+1/2}) = v_l^{(i)}(x_{i+1/2}^-), \quad v_l^{(i)}(x_{i-1/2}) = v_l^{(i)}(x_{i-1/2}^+). \end{aligned}$$

The numerical fluxes for convection term $\hat{f}_{i+1/2}$ will be presented in detail in Subsection 2.2. The integral term in (2.9) can be computed either exactly or by a suitable numerical quadrature accurate to at least $\mathcal{O}(\Delta x^{k+l+2})$. In this paper we use two and three point Gaussian quadratures for $k=1$ and $k=2$ respectively.

Step 2. The reconstruction of the second time derivative and the variable $q'=(u')_x$:

$$\begin{cases} u'' = -(f(u)_t - fd(u,q)_t)_x = -(f'(u)u' - fd_u(u,q)u' - fd_q(u,q)q')_x, \\ q' = (u')_x. \end{cases} \tag{2.10}$$

are obtained as following. Notice that we will only need an approximation of order k, one order lower than before, because of the extra Δt factor in (2.4).

Step 2.1. The procedure of reconstruction of $q' = u'_x$ from u' is similar to that of q in step 1.1 with numerical flux $\hat{u}'_{i+1/2} = u'(x^-_{i+1/2}, t^n)$, but we can reconstruct q' just in V_h^{k-1}.

Step 2.2. The procedure of reconstruction of

$$u'' = -(f'(u)u' - fd_u(u,q)u' - fd_q(u,q)q')_x,$$

in V_h^{k-1} is similar to that in Step 1.2. As in Step 1.2, we split the fluxes $f'(u)u' - fd_u(u,q)u' - fd_q(u,q)q'$ into two parts, one comes from convection term, $f'(u)u'$; another from diffusion term, $-fd_u(u,q)u' - fd_q(u,q)q'$. Here we just replace $f'(u)u'$ at $x_{i+1/2}$ with a central average, that is, we use $\frac{1}{2}(f'(u^-_{i+1/2})(u')^-_{i+1/2} + f'(u^+_{i+1/2})(u')^+_{i+1/2})$ to replace value of $f'(u)u'$ at $x_{i+1/2}$. It seems that a more costly numerical flux approximation is *not* needed here to control spurious oscillations, presumably because this term is multiplied by an extra Δt anyway. For the term comes from diffusion, as in Step 1.2, we again use $(-fd_u(u,q)u' - fd_q(u,q)q')^+_{i+1/2}$ to replace value of $-fd_u(u,q)u' - fd_q(u,q)q'$ at $x_{i+1/2}$.

Step 3. The reconstruction of the third time derivative:

$$\begin{aligned} u_{ttt} = &-(f'(u)u'' + f''(u)(u')^2)_x - (fd_{uu}(u,q)(u')^2 + 2fd_{uq}(u,q)u'q' \\ &+ fd_{qq}(u,q)(q')^2 + fd_u(u,q)u'' + fd_q(u,q)q'')_x. \end{aligned}$$

Notice that we will only need an approximation of order $k-1$, one order lower than the second time derivative, because of the extra Δt^2 factor in (2.4), the procedure of reconstruction of the third time derivative is similar as that of the second time derivative, but we just reconstruct it in V_h^{k-2}.

For systems of conservation laws (2.1), $u(x,t) = (u^1(x,t),\cdots,u^m(x,t))^T$ is a vector and $f(u) = (f^1(u^1,\cdots,u^m),\cdots,f^m(u^1,\cdots,u^m))^T$ is a vector function of u. As before, the time derivatives in (2.2) are replaced by the spatial derivatives using the PDE. The DG discretization is then performed on each component.

For two dimensional cases, we consider the convection-diffusion equations:

$$\begin{cases} u_t + f(u)_x + g(u)_y = fd(u,u_x,u_y)_x + gd(u,u_x,u_y)_y, \\ u(x,y,0) = u_0(x,y). \end{cases} \tag{2.11}$$

By a temporal Taylor expansion we obtain

$$u(x,y,t+\Delta t) = u(x,y,t) + \Delta t u' + \frac{\Delta t^2}{2}u'' + \frac{\Delta t^3}{6}u''' + \cdots.$$

For example, for third order accuracy in time we would need to reconstruct three time derivatives: u', u'', u'''.

We again use the PDE (2.11) to replace time derivatives by spatial derivatives:

$$\begin{cases} u' = -(f(u) - fd(u,q,s))_x - (g(u) - fd(u,q,s))_y, \\ q = u_x, \quad s = u_y, \end{cases} \tag{2.12}$$

$$\begin{cases} u'' = -(f'(u)u' - fd_u u' - fd_q q' - fd_s s')_x \\ \qquad -(g'(u)u' - gd_u u' - gd_q q' - gd_s s')_y, \\ q' = u'_x, \quad s' = u'_y, \end{cases} \tag{2.13}$$

and

$$\begin{cases} u''' = -(f''(u)(u')^2 + f'(u)u'' - fd_{uu}(u')^2 - fd_{qq}(q')^2 - fd_{ss}(s')^2 \\ \qquad -2fd_{uq}u'q' - 2fd_{us}u's' - 2fd_{qs}q's' - fd_u u'' - fd_q q'' - fd_s s'')_x \\ \qquad -(g''(u)(u')^2 + g'(u)u'' - gd_{uu}(u')^2 - gd_{qq}(q')^2 - gd_{ss}(s')^2 \\ \qquad -2gd_{uq}u'q' - 2gd_{us}u's' - 2gd_{qs}q's' - gd_u u'' - gd_q q'' - gd_s s'')_y, \\ q'' = u''_x, \quad s'' = u''_y. \end{cases} \tag{2.14}$$

Then we follow the Step 1, Step 2 and Step 3 to reconstruct the first, the second and the third time derivatives, respectively.

In order to maintain stability and non-oscillatory property of the DG method for solving conservation laws (1.1) with strong shocks, a nonlinear limiter must be applied. In the numerical experiments in this paper, we will use the TVB limiter adopted in [7] only to detect troubled cells, where a WENO limiter developed in [23] will be used for the reconstruction of first and higher order moments of the polynomials inside those troubled cells. We refer to [23] for the details of this WENO reconstruction and will not repeat it here. For the case of hyperbolic systems, to identify the troubled cells, we could either use a component-wise indicator or a characteristic one. In this paper we will use the characteristic indicator.

2.2. Description of fluxes for the convection term for LWDG schemes

We now review the two-point or essentially two-point numerical fluxes for the convection term. Numerical experiments to compare their performance for the LWDG method will be given in next section.

For the one dimensional system case, we will consider Euler or Navier-Stokes equations of compressible gas dynamics, namely (2.1) with

$$u = (\rho, \rho v, E)^T, \qquad f(u) = (\rho v, \rho v^2 + p, v(E+p))^T, \tag{2.15}$$

with $fd(u,u_x) = 0$ and

$$fd(u,u_x) = (0, \frac{1}{\mathrm{Re}}(\frac{4}{3}v_{xx}), \frac{1}{\mathrm{Re}}[\frac{2}{3}(v^2)_{xx} + \frac{1}{(\gamma-1)\mathrm{Pr}}(c^2)_{xx}])^T$$

for Euler and Navier-Stokes equations, respectively, where ρ is the density, v is the velocity, E is the total energy, p is the pressure, which is related to the total energy by $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho v^2$ with $\gamma = 1.4$ for air, $c = \sqrt{\gamma p/\rho}$ is the sound speed. Pr is the Prandtl number, with Pr=0.7. Re is the Reynolds number.

1. The Lax-Friedrichs (LF) flux and the local LF (LLF) flux.

The LF flux is one of the simplest and most widely used building blocks for the DG method and high order finite volume methods such as the ENO and WENO schemes. However, the numerical viscosity of the LF flux is also the largest among monotone fluxes for scalar problems. The LF or the LLF flux is defined by:

$$\hat{f}_{i+1/2} = \frac{1}{2}(f^-_{i+1/2} + f^+_{i+1/2} - \alpha(u^+_{i+1/2} - u^-_{i+1/2})), \tag{2.16}$$

where $u^\pm_{i+1/2}$ and $f^\pm_{i+1/2}$ are the left and right limits of the discontinuous solution u^h and f at the cell interface $x_{i+1/2}$, respectively. For the LF flux, α is taken as an upper bound over the whole line for $|f'(u)|$ in the scalar case, or for the absolute value of eigenvalues of the Jacobian for the system case, and for the LLF flux α is taken as an upper bound of $|f'(u)|$ between u^- and u^+.

2. The Godunov flux.

The Godunov flux [13,30] is based on the *exact* Riemann solver, which has the smallest numerical viscosity among all monotone fluxes for the scalar case but could be very costly to evaluate in the system case, as it often lacks explicit formulas and relies on iterative procedures for its evaluation. The Godunov flux is defined as

$$\hat{f}^G(u^-,u^+) = f(u(0)),$$

where $u(0)$ is the solution of the local Riemann problem at $x/t = 0$ (the solution of the local Riemann problem is a function of the single variable x/t only due to self-similarity), i.e. the exact solution to the conservation law (2.1) with the initial condition:

$$u(x,0) = \begin{cases} u^- & \text{for} \quad x \le 0, \\ u^+ & \text{for} \quad x > 0. \end{cases}$$

For the scalar case, the Godunov flux can be expressed in a closed form as

$$\hat{f}^G(u^-,u^+) = \begin{cases} \min_{u^- \le u \le u^+} f(u) & \text{if} \quad u^- \le u^+, \\ \max_{u^+ \le u \le u^-} f(u) & \text{if} \quad u^- > u^+. \end{cases} \tag{2.17}$$

However, for most nonlinear systems, the Godunov flux cannot be expressed in a closed form. Its evaluation would in general require an iterative procedure. We refer to [30] and the references therein for the details of the exact Riemann solver for systems in applications, such as the Euler equations (2.15), which are needed for the evaluation of the Godunov flux for such systems.

3. The Engquist-Osher (EO) flux and the Osher-Solomon flux [12,18].

The Engquist-Osher (EO) flux [12] for the scalar case and its extension to systems (often referred to as the Osher-Solomon flux [18]) are smoother than the Godunov flux with almost as small numerical viscosity, and have the advantage of explicit formulas for the scalar case and for some well known physical systems, such as the Euler equations of compressible gas dynamics.

For the scalar case the EO flux is given by:

$$\hat{f}^{EO}(u^-,u^+)=\frac{1}{2}\left(f(u^-)+f(u^+)-\int_{u^-}^{u^+}|f'(u)|du\right), \tag{2.18}$$

For the system case, the explicit formulas for the Osher-Solomon flux for the Euler equations (2.15) refers to [18,21]; we do not repeat it here to save space.

4. The Harten-Lax-van Leer (HLL) flux [15,30].

The HLL flux [15] is based on the approximate Riemann solver with only three constant states separated by two waves. The HLL flux for the Euler equations (2.15) is given by:

$$\hat{f}^{HLL}(u^-,u^+)=\begin{cases} f(u^-), & \text{if} \quad 0\le s^-,\\ \frac{s^+f(u^-)-s^-f(u^+)+s^-s^+(u^+-u^-)}{s^+-s^-}, & \text{if} \quad s^-\le 0\le s^+,\\ f(u^+), & \text{if} \quad s^+\le 0. \end{cases} \tag{2.19}$$

where the lower and upper bounds of the wave speed, s^- and s^+, must be estimated. We use the pressure-velocity estimates [30]

$$s^-=v^--c^-q^-, \qquad s^*=v^*, \qquad s^+=v^++c^+q^+, \tag{2.20}$$

where, for $K=\pm$,

$$q^K=\begin{cases} 1, & \text{if} \quad p^*\le p^K,\\ (1+\frac{\gamma+1}{2\gamma}(p^*/p^K-1))^{1/2}, & \text{if} \quad p^*>p^K \end{cases}$$

with

$$p^*=\frac{1}{2}(p^-+p^+)-\frac{1}{2}(v^+-v^-)\overline{\rho}\,\overline{c}, \qquad v^*=\frac{1}{2}(v^-+v^+)-\frac{p^+-p^-}{2\overline{\rho}\,\overline{c}},$$
$$\overline{\rho}=\frac{1}{2}(\rho^-+\rho^+), \qquad \overline{c}=\frac{1}{2}(c^-+c^+).$$

5. The HLLC flux – a modification of the HLL flux [30,31].

The HLLC flux is a modification of the HLL flux, whereby the missing contact and shear waves are restored. The HLLC flux for the Euler equations (2.15) is given by:

$$\hat{f}^{HLLC}(u^-,u^+)=\begin{cases} f(u^-), & \text{if} \quad 0\le s^-,\\ f(u^-)+s^-(u^{*-}-u^-), & \text{if} \quad s^-\le 0\le s^*,\\ f(u^+)+s^+(u^{*+}-u^+), & \text{if} \quad s^*\le 0\le s^+,\\ f(u^+), & \text{if} \quad s^+\le 0, \end{cases} \tag{2.21}$$

where, for $K=\pm$,

$$u^{*K}=\rho^K\frac{s^K-v^K}{s^K-s^*}\begin{bmatrix} 1\\ s^*\\ \frac{E^K}{\rho^K}+(s^*-v^K)[s^*+\frac{p^K}{\rho^K(s^K-v^K)}] \end{bmatrix}. \tag{2.22}$$

The definitions of s^-, s^* and s^+ are given in (2.20).

6. The first-order centered (FORCE) flux [30].

The FORCE flux is given by:

$$\hat{f}^{FORCE}(u^-,u^+) = \frac{1}{2}\left(\hat{f}^{LF}(u^-,u^+) + \hat{f}^R(u^-,u^+)\right), \tag{2.23}$$

where $\hat{f}^R$ is the second order Richtmyer flux given by

$$\hat{f}^R(u^-,u^+) = f(u^*), \qquad u^* = \frac{1}{2}\left(u^- + u^+ - \frac{\Delta t}{\Delta x}(f(u^+) - f(u^-))\right). \tag{2.24}$$

This flux is the average of the LF flux and the second order Richtmyer flux, hence its viscosity is smaller than that of the LF flux.

7. A flux limiter centered (FLIC) flux [30].

The general flux limiter approach combines a low order monotone flux and a high order flux. The FLIC flux we use has the FORCE flux as the low order flux and the Richtmyer flux as the high order flux:

$$\hat{f}^{FLIC}(u^-,u^+) = \hat{f}^{FORCE}(u^-,u^+) + \phi_{i+1/2}[\hat{f}^R(u^-,u^+) - \hat{f}^{FORCE}(u^-,u^+)], \tag{2.25}$$

where $\phi_{i+1/2}$ is a flux limiter. There are several possible choices for the flux limiter such as the superbee, van Leer and the minbee flux limiters [30]. Following [30], for the Euler equation we use the following procedure: we first define $q = E$ (total energy) and set

$$r^-_{i+1/2} = \frac{\Delta q_{i-1/2}}{\Delta q_{i+1/2}}, \qquad r^+_{i+1/2} = \frac{\Delta q_{i+3/2}}{\Delta q_{i+1/2}},$$

where $\Delta q_{i-1/2} = \overline{q}_i - \overline{q}_{i-1}$, and $\overline{q}_i$ is the cell average of q on the cell I_i. We then compute a single flux limiter

$$\phi_{i+1/2} = \min(\phi(r^-_{i+1/2}),\ \phi(r^+_{i+1/2})),$$

and apply it to all components of the flux. In this paper we use the minbee limiter:

$$\phi(r) = \begin{cases} 0, & r \le 0, \\ r, & 0 \le r \le 1, \\ 1, & r \ge 1. \end{cases}$$

Clearly, if $u^- = u^+ = u$, then $\hat{f}^{FLIC}(u,u) = f(u)$. Hence even if the FLIC flux depends on more than the two-point u^- and u^+ through the limiter $\phi_{i+1/2}$ and we are abusing notations when we denote it by $\hat{f}^{FLIC}(u^-,u^+)$, it is indeed an essentially two-point flux as defined before, hence can be used as a flux for the DG method.

In next section we will use these fluxes to perform numerical experiments.

Table 1: CPU time (in seconds) for the LWDG methods with different fluxes, for the accuracy test problem of Euler equations. Total CPU time for $N = 10, 20, \cdots, 1280$ cells is recorded.

Flux	LF	LLF	G	EO	HLL	HLLC	FORCE	FLIC
$k = 1$	23.64	23.80	43.93	47.27	27.82	28.16	25.19	26.76
$k = 2$	80.20	81.75	112.89	120.21	87.77	88.24	83.31	86.65

Table 2: CPU time (in seconds) for the LWDG methods with different fluxes, for the accuracy test problem of Navier-Stokes equations with Reynolds number Re=10000. Total CPU time for $N = 10, 20, \cdots, 1280$ cells is recorded.

Flux	LF	LLF	G	EO
$k = 1$	59.63	59.69	81.43	83.67
$k = 2$	269.68	270.44	305.23	312.30
Flux	HLL	HLLC	FORCE	FLIC
$k = 1$	63.51	63.68	61.54	62.02
$k = 2$	276.94	277.20	272.65	274.13

3. Numerical results

In this section we perform extensive numerical experiments to compare the performance of the LWDG method based on the eight different fluxes outlined in the previous section. The detailed numerical study is mainly performed for the one dimensional system case, addressing the issues of CPU cost, accuracy, non-oscillatory property, and resolution of discontinuities. Numerical tests are also performed for two dimensional systems. In all the figures, we plot only the cell averages of the numerical solution. For CPU time comparison, all the computations are performed on a Dell Server 1850 with CPU Xeon P4-3.4GHz and 4GB ram. We denote the LWDG scheme with the flux "X" as LWDG-X, such as LWDG-LF for the LWDG scheme with the LF flux. In our numerical experiments, the CFL numbers $cflc$ for convection terms are taken as 0.2 and 0.12, and for diffusion terms the CFL numbers $cfld$ are taken as 0.01 and 0.003 for diffusion terms, for $k = 1$ and $k = 2$ (second and third order accuracy), respectively, and time step

$$\Delta t = 1/(\frac{\alpha}{cflc * \Delta x} + \frac{1}{Re * cfld * \Delta x^2}),$$

where, α is taken as an upper bound over the whole line for the absolute value of eigenvalues of $f'(u)$.

Example 3.1. We solve the one dimensional nonlinear system of Euler and Navier-Stokes equations (2.15). The initial condition is set to be $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, with a 2-periodic boundary condition. The exact solution for Euler equations is $\rho(x,t) = 1 + 0.2\sin(\pi(x - t))$, $v(x,t) = 1$, $p(x,t) = 1$. For Navier-Stokes equations, we add a source term to ensure the exact solution is also $\rho(x,t) = 1 + 0.2\sin(\pi(x - t))$, $v(x,t) = 1$, $p(x,t) = 1$. We compute the solution up to $t = 2$ and TVB constant $M = 0.01$. In Tables 1 and 2 we provide a CPU time comparison for the LWDG schemes with different

Table 3: Euler equations. The numerical errors and the orders of accuracy for the density ρ, and the ratios of the numerical errors by LWDG with different fluxes compared with those by the LWDG-LF scheme. $k = 1$.

N	flux	L_1 error	L_1 order	error ratio	L_∞ error	L_∞ order	error ratio
80	LF	3.2821E-05	2.0080	1.0000	1.4322E-04	1.9628	1.0000
	LLF	3.3058E-05	2.0083	1.0072	1.5099E-04	1.9683	1.0543
	G	4.4033E-05	2.0566	1.3416	1.9697E-04	1.9316	1.3753
	EO	4.4033E-05	2.0566	1.3416	1.9697E-04	1.9316	1.3753
	HLL	3.9933E-05	2.0478	1.2167	1.8936E-04	1.9300	1.3222
	HLLC	4.4033E-05	2.0566	1.3416	1.9697E-04	1.9316	1.3753
	FORCE	3.2895E-05	2.0085	1.0023	1.4395E-04	1.9624	1.0051
	FLIC	3.2854E-05	2.0079	1.0010	1.4486E-04	1.9584	1.0115
320	LF	2.0428E-06	2.0020	1.0000	9.1257E-06	1.9907	1.0000
	LLF	2.0598E-06	2.0015	1.0083	9.4642E-06	2.0021	1.0371
	G	2.6846E-06	2.0112	1.3142	1.2727E-05	1.9842	1.3947
	EO	2.6846E-06	2.0112	1.3142	1.2727E-05	1.9842	1.3947
	HLL	2.4510E-06	2.0075	1.1998	1.2251E-05	1.9836	1.3424
	HLLC	2.6846E-06	2.0112	1.3142	1.2727E-05	1.9842	1.3947
	FORCE	2.0471E-06	2.0020	1.0021	9.1752E-06	1.9906	1.0054
	FLIC	2.0456E-06	2.0019	1.0014	9.3557E-06	1.9786	1.0252
1280	LF	1.2755E-07	2.0005	1.0000	5.7305E-07	1.9977	1.0000
	LLF	1.2859E-07	1.9968	1.0082	5.9471E-07	2.0451	1.0378
	G	1.6688E-07	2.0026	1.3084	8.0194E-07	1.9961	1.3994
	EO	1.6688E-07	2.0026	1.3084	8.0189E-07	1.9961	1.3993
	HLL	1.5265E-07	2.0016	1.1969	7.7210E-07	1.9960	1.3474
	HLLC	1.6688E-07	2.0026	1.3084	8.0189E-07	1.9961	1.3993
	FORCE	1.2781E-07	2.0005	1.0021	5.7619E-07	1.9977	1.0055
	FLIC	1.2764E-07	2.0005	1.0007	5.9196E-07	1.9993	1.0330

fluxes. The numerical errors and the orders of accuracy for the density ρ, and ratios of the numerical errors for comparison with the LWDG-LF scheme are shown in Tables 3-6.

We can see that the LWDG-LF scheme costs the least CPU time for each of the cases $k = 1$ and 2. For Euler equations, the LWDG-G and LWDG-EO schemes cost 100% more than that of the LWDG-LF scheme for $k = 1$ and 40% for $k = 2$, the LWDG-HLL and LWDG-HLLC schemes cost about 20% more than that of the LWDG-LF scheme for $k = 1$ and 10% for $k = 2$, the LWDG-LLF, LWDG-FORCE and LWDG-FLIC schemes cost a little more than that of the LWDG-LF scheme for both $k = 1$ and 2. For Navier-Stokes equations, the LWDG-G and LWDG-EO schemes cost 40% more than that of the LWDG-LF scheme for $k = 1$ and 15% for $k = 2$, the LWDG-HLL and LWDG-HLLC schemes cost about 8% more than that of the LWDG-LF scheme for $k = 1$ and 3% for $k = 2$. The LWDG-LLF, LWDG-FORCE and LWDG-FLIC schemes cost a little more than that of the LWDG-LF scheme for both $k = 1$ and 2 and for both Euler and Navier-Stokes equations. Of course, this CPU time comparison depends on our specific implementation of these fluxes and also on the specific test case (for the Godunov flux which has an iteration procedure and may converge with different number of steps for different solutions), but it does give the correct ball-park of the relative

Table 4: Euler equations. The numerical errors and the orders of accuracy for the density ρ, and the ratios of the numerical errors by LWDG with different fluxes compared with those by the LWDG-LF scheme. $k=2$.

N	flux	L_1 error	L_1 order	error ratio	L_∞ error	L_∞ order	error ratio
80	LF	5.5999E-07	2.9838	1.0000	2.6869E-06	2.9770	1.0000
	LLF	5.2588E-07	2.9890	0.9391	2.5654E-06	2.9829	0.9548
	G	2.7593E-07	3.0008	0.4927	1.6137E-06	2.9971	0.6006
	EO	2.7592E-07	3.0008	0.4927	1.6137E-06	2.9971	0.6006
	HLL	3.1371E-07	2.9993	0.5602	1.7692E-06	2.9986	0.6584
	HLLC	2.7592E-07	3.0008	0.4927	1.6137E-06	2.9971	0.6006
	FORCE	5.5433E-07	2.9844	0.9899	2.6668E-06	2.9777	0.9925
	FLIC	5.5454E-07	2.9849	0.9903	2.6670E-06	2.9779	0.9926
320	LF	8.7809E-09	2.9990	1.0000	4.2201E-08	2.9985	1.0000
	LLF	8.2333E-09	2.9997	0.9376	4.0197E-08	2.9994	0.9525
	G	4.3146E-09	2.9989	0.4914	2.5332E-08	2.9947	0.6003
	EO	4.3110E-09	3.0000	0.4910	2.5231E-08	2.9998	0.5979
	HLL	4.9024E-09	3.0000	0.5583	2.7605E-08	3.0007	0.6541
	HLLC	4.3110E-09	3.0000	0.4910	2.5231E-08	2.9998	0.5979
	FORCE	8.6910E-09	2.9990	0.9898	4.1879E-08	2.9985	0.9924
	FLIC	8.6918E-09	2.9992	0.9899	4.1879E-08	2.9985	0.9924
1280	LF	1.3723E-10	2.9999	1.0000	6.6135E-10	2.9961	1.0000
	LLF	1.2862E-10	2.9969	0.9373	6.2901E-10	2.9979	0.9511
	G	7.5319E-11	2.8541	0.5488	5.2448E-10	2.6509	0.7930
	EO	6.7365E-11	2.9999	0.4909	3.9633E-10	2.9925	0.5993
	HLL	7.6601E-11	3.0000	0.5582	4.3210E-10	2.9970	0.6534
	HLLC	6.7365E-11	2.9999	0.4909	3.9635E-10	2.9925	0.5993
	FORCE	1.3582E-10	2.9999	0.9898	6.5632E-10	2.9961	0.9924
	FLIC	1.3583E-10	3.0000	0.9898	6.5634E-10	2.9960	0.9924

CPU costs of the LWDG method using these different fluxes.

On the numerical errors, for the case of $k=1$, the L_1 and L_∞ errors of LWDG-LF are smallest for the same meshes among all schemes, but for the case of $k=2$, errors of LWDG-LF are largest.

For the case of $k=1$, the L_1 and L_∞ errors of LWDG-G, LWDG-EO, LWDG-HLL and LWDG-HLLC schemes are about 30%, 30%, 20% and 30% larger than those by the LWDG-LF scheme for the same meshes, respectively. For the case of $k=2$, the L_1 and L_∞ errors of LWDG-G, LWDG-EO, LWDG-HLL and LWDG-HLLC schemes are about 30-70% of those by the LWDG-LF scheme for the same meshes. The L_1 and L_∞ errors of the LWDG-LLF, LWDG-FORCE and LWDG-FLIC schemes are similar to those of the LWDG-LF scheme for both $k=1$ and 2 cases. This indicates that we have to be cautious when discussing about the accuracy advantage of various fluxes as this may depend on the order of accuracy of the scheme for the nonlinear systems case, though it was clearly shown that the Godunov flux is always superior in accuracy compared to the more diffusive LLF flux for DG schemes of any order of accuracy between two and six for elastic waves problem in [19].

Example 3.2. We consider the interaction of blast waves of the Euler equation (2.15) with

Table 5: Navier-Stokes equations, Reynolds number Re=10000. The numerical errors and the orders of accuracy for the density ρ, and the ratios of the numerical errors by LWDG with different fluxes compared with those by the LWDG-LF scheme. $k=1$.

N	flux	L_1 error	L_1 order	error ratio	L_∞ error	L_∞ order	error ratio
80	LF	3.2836E-05	2.0081	1.0000	1.4311E-04	1.9628	1.0000
	LLF	3.3074E-05	2.0083	1.0072	1.5093E-04	1.9682	1.0547
	G	4.3346E-05	2.0684	1.3201	1.9562E-04	1.9347	1.3669
	EO	4.3345E-05	2.0684	1.3200	1.9562E-04	1.9348	1.3669
	HLL	3.9211E-05	2.0603	1.1941	1.8420E-04	1.9488	1.2871
	HLLC	4.3346E-05	2.0684	1.3201	1.9562E-04	1.9347	1.3669
	FORCE	3.2908E-05	2.0087	1.0022	1.4383E-04	1.9625	1.0050
	FLIC	3.2908E-05	2.0071	1.0022	1.4383E-04	1.9628	1.0050
320	LF	2.0437E-06	2.0020	1.0000	9.1191E-06	1.9907	1.0000
	LLF	2.0607E-06	2.0016	1.0083	9.4578E-06	2.0026	1.0371
	G	2.5784E-06	2.0310	1.2616	1.2521E-05	1.9922	1.3731
	EO	2.5784E-06	2.0310	1.2616	1.2521E-05	1.9922	1.3731
	HLL	2.3314E-06	2.0349	1.1408	1.1373E-05	2.0222	1.2472
	HLLC	2.5784E-06	2.0310	1.2616	1.2521E-05	1.9922	1.3731
	FORCE	2.0472E-06	2.0023	1.0018	9.1636E-06	1.9910	1.0049
	FLIC	2.0472E-06	2.0023	1.0018	9.1636E-06	1.9910	1.0049
1280	LF	1.2759E-07	2.0006	1.0000	5.7269E-07	1.9976	1.0000
	LLF	1.2863E-07	1.9987	1.0082	5.9438E-07	2.0508	1.0379
	G	1.5707E-07	2.0119	1.2311	7.8323E-07	1.9983	1.3676
	EO	1.5707E-07	2.0119	1.2311	7.8323E-07	1.9983	1.3676
	HLL	1.4019E-07	2.0221	1.0988	6.8822E-07	2.0166	1.2017
	HLLC	1.5707E-07	2.0119	1.2311	7.8323E-07	1.9983	1.3676
	FORCE	1.2771E-07	2.0013	1.0010	5.7472E-07	1.9989	1.0035
	FLIC	1.2771E-07	2.0013	1.0010	5.7472E-07	1.9989	1.0035

the initial condition:

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

A reflecting boundary condition is applied to both ends. See [14,32]. The computational domain is $[0,1]$, the final time is $t=0.038$, and TVB constant $M=100.0$.

In Tables 7-8, we provide a CPU time comparison for the LWDG schemes with different fluxes. From these tables, we can see that the relation of CPU times by the LWDG with different fluxes is also similar to those in the previous example. We can see again that the LWDG-LF scheme costs the least CPU time for each of the cases $k=1$ and 2. For Euler equations, the LWDG-G and LWDG-EO schemes cost 60% more than that of the LWDG-LF scheme for $k=1$ and 30% for $k=2$, the LWDG-HLL and LWDG-HLLC schemes cost about 12% more than that of the LWDG-LF scheme for $k=1$ and $k=2$. For Navier-Stokes equations, the LWDG-G and LWDG-EO schemes cost 40% more than that of the LWDG-LF scheme for $k=1$ and 15% for $k=2$, the LWDG-HLL and LWDG-HLLC schemes cost about

Table 6: Navier-Stokes equations, Reynolds number Re=10000. The numerical errors and the orders of accuracy for the density ρ, and the ratios of the numerical errors by LWDG with different fluxes compared with those by the LWDG-LF scheme. $k=2$.

N	flux	L_1 error	L_1 order	error ratio	L_∞ error	L_∞ order	error ratio
80	LF	6.6271E-07	2.9152	1.0000	3.0107E-06	2.9632	1.0000
	LLF	6.1296E-07	2.9271	0.9249	2.8109E-06	2.9729	0.9336
	G	2.6369E-07	3.0343	0.3979	1.5096E-06	3.0465	0.5014
	EO	2.6369E-07	3.0343	0.3979	1.5096E-06	3.0465	0.5014
	HLL	2.9357E-07	3.0474	0.4430	1.6416E-06	3.0542	0.5453
	HLLC	2.6369E-07	3.0343	0.3979	1.5096E-06	3.0465	0.5014
	FORCE	6.5512E-07	2.9162	0.9886	2.9845E-06	2.9641	0.9913
	FLIC	6.5512E-07	2.9168	0.9886	2.9845E-06	2.9644	0.9913
320	LF	9.8267E-09	3.0780	1.0000	4.4056E-08	3.0799	1.0000
	LLF	9.0992E-09	3.0755	0.9260	4.1328E-08	3.0761	0.9381
	G	3.7028E-09	3.0900	0.3768	1.8875E-08	3.2172	0.4284
	EO	3.7029E-09	3.0899	0.3768	1.8875E-08	3.2172	0.4284
	HLL	3.9485E-09	3.1300	0.4018	2.0355E-08	3.2215	0.4620
	HLLC	3.7029E-09	3.0899	0.3768	1.8875E-08	3.2172	0.4284
	FORCE	9.7083E-09	3.0784	0.9880	4.3656E-08	3.0800	0.9909
	FLIC	9.7083E-09	3.0784	0.9880	4.3656E-08	3.0800	0.9909
1280	LF	1.1563E-10	3.2375	1.0000	5.2659E-10	3.2260	1.0000
	LLF	1.0809E-10	3.2242	0.9348	4.9649E-10	3.2273	0.9428
	G	5.7600E-11	2.9323	0.4981	1.4708E-10	3.5890	0.2793
	EO	5.5540E-11	2.9841	0.4803	1.4710E-10	3.5888	0.2794
	HLL	5.8208E-11	2.9870	0.5034	1.6386E-10	3.5524	0.3112
	HLLC	5.5540E-11	2.9841	0.4803	1.4708E-10	3.5890	0.2793
	FORCE	1.1423E-10	3.2370	0.9879	5.2174E-10	3.2260	0.9908
	FLIC	1.1423E-10	3.2370	0.9879	5.2174E-10	3.2260	0.9908

Table 7: CPU time (in seconds) for the LWDG methods with different fluxes, for the blast wave problem of Euler equations. Total CPU time for $N=200$ and 400 cells is recorded.

Flux	LF	LLF	G	EO	HLL	HLLC	FORCE	FLIC
$k=1$	2.18	2.27	3.29	3.75	2.48	2.49	2.27	2.37
$k=2$	7.12	7.47	9.25	10.03	7.91	7.95	7.30	7.47

Table 8: CPU time (in seconds) for the LWDG methods with different fluxes, for the blast wave problem of Navier-Stokes equations with Reynolds number Re=10000. Total CPU time for $N=200$ and 400 cells is recorded.

Flux	LF	LLF	G	EO	HLL	HLLC	FORCE	FLIC
$k=1$	2.63	2.72	3.75	4.22	2.96	2.95	2.74	2.83
$k=2$	9.50	9.94	11.67	12.44	10.35	10.46	9.73	9.88

12% more than that of the LWDG-LF scheme for $k=1$ and 5% for $k=2$. The LWDG-LLF, LWDG-FORCE and LWDG-FLIC schemes cost a little more than that of the LWDG-LF scheme for both $k=1$ and 2 and for both Euler and Navier-Stokes equations.

In Figs. 1-4, the computed densities ρ with 400 cells are plotted against the reference

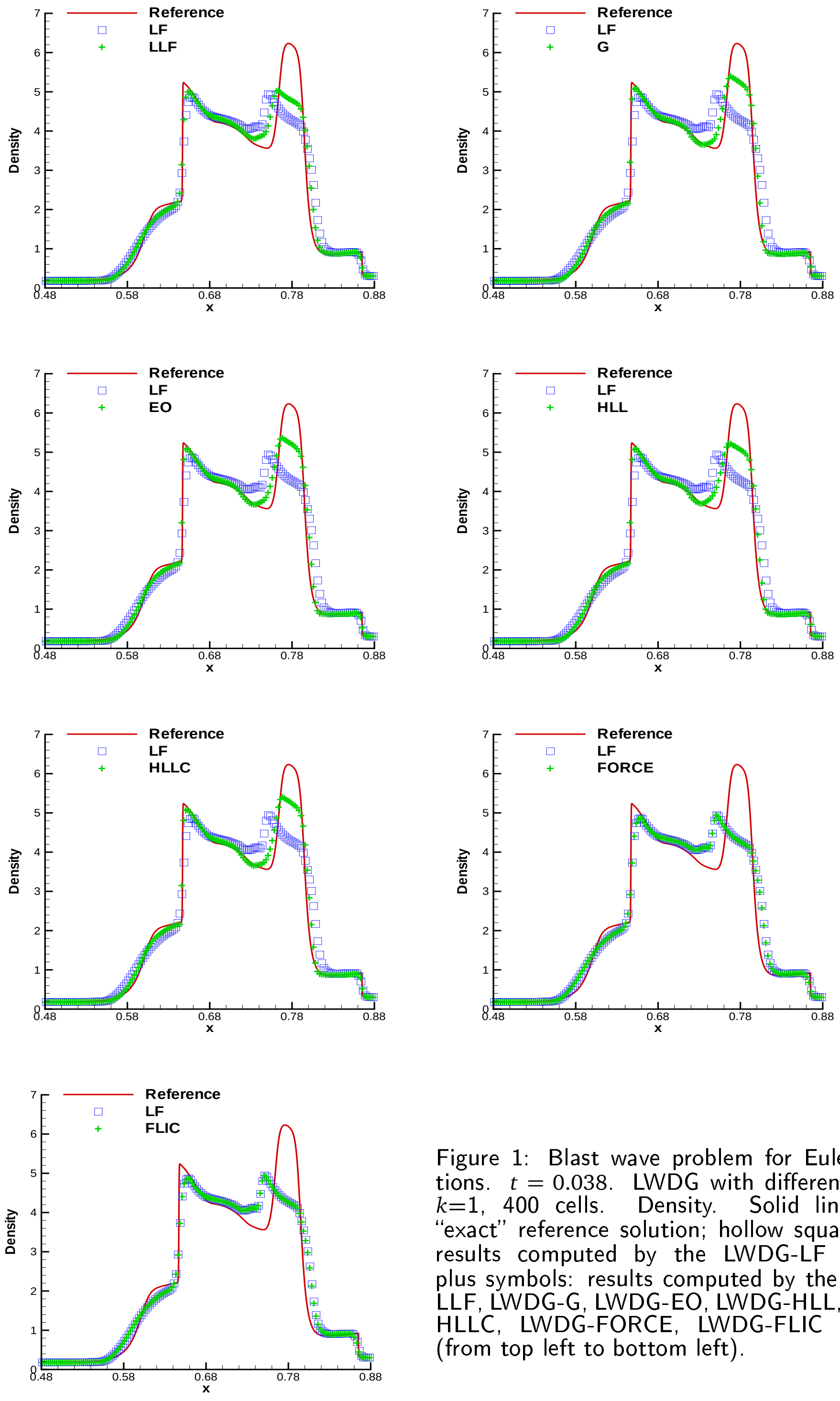


Figure 1: Blast wave problem for Euler equations. $t = 0.038$. LWDG with different fluxes, k=1, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the LWDG-LF scheme; plus symbols: results computed by the LWDG-LLF, LWDG-G, LWDG-EO, LWDG-HLL, LWDG-HLLC, LWDG-FORCE, LWDG-FLIC schemes (from top left to bottom left).

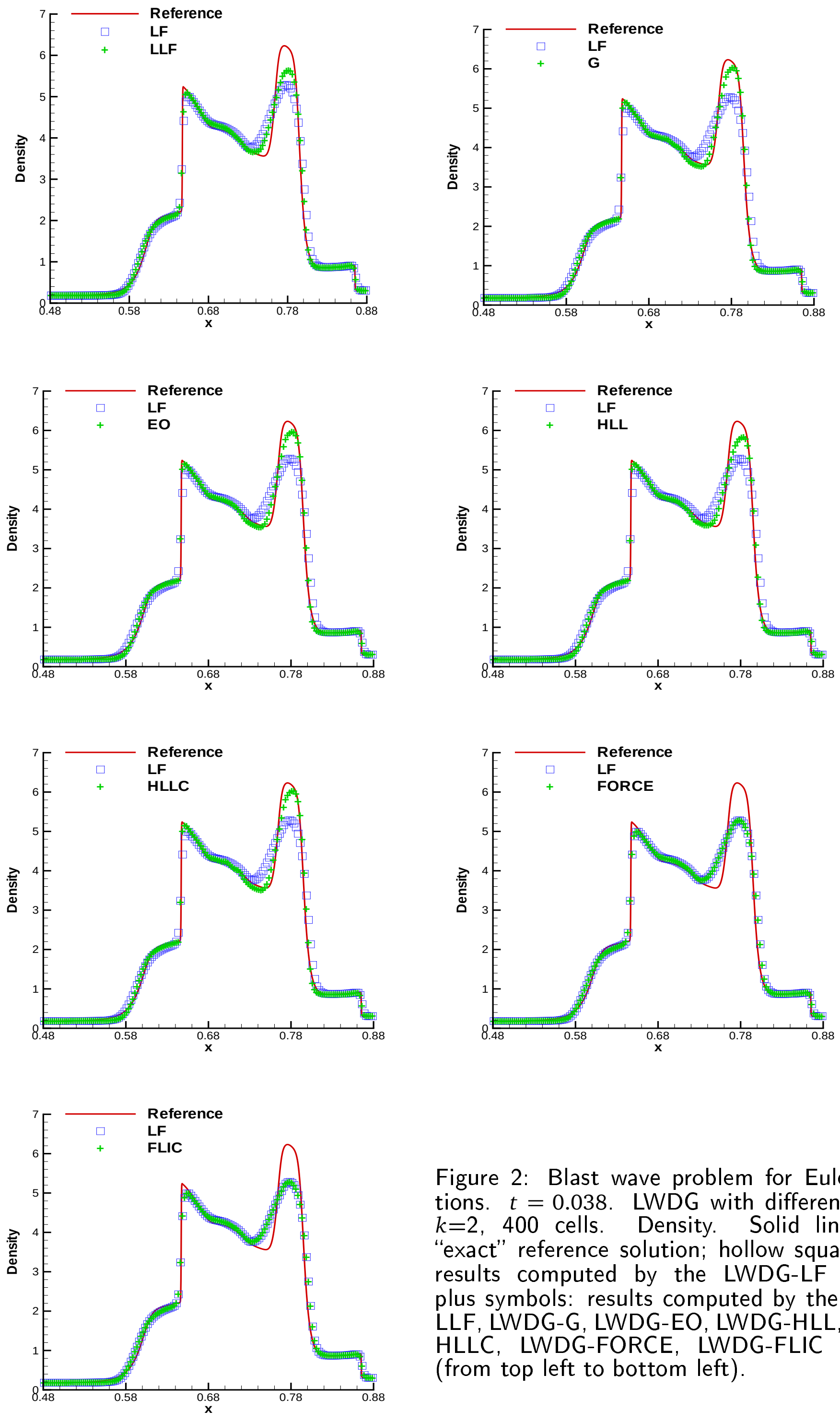
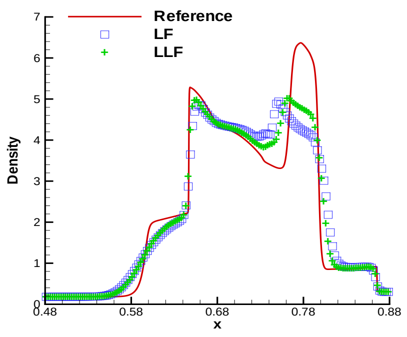
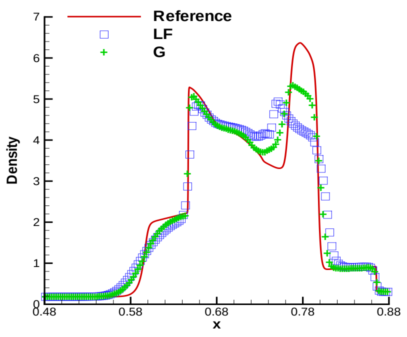
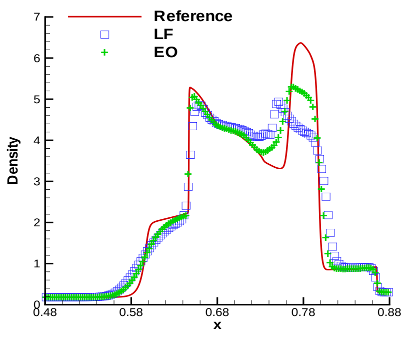
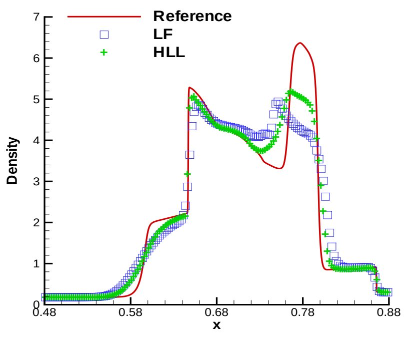


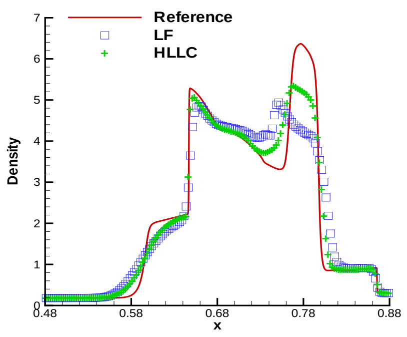
Figure 2: Blast wave problem for Euler equations. $t = 0.038$. LWDG with different fluxes, k=2, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the LWDG-LF scheme; plus symbols: results computed by the LWDG-LLF, LWDG-G, LWDG-EO, LWDG-HLL, LWDG-HLLC, LWDG-FORCE, LWDG-FLIC schemes (from top left to bottom left).

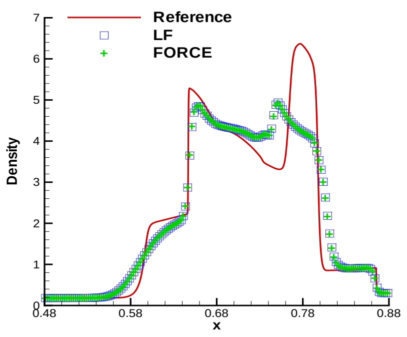












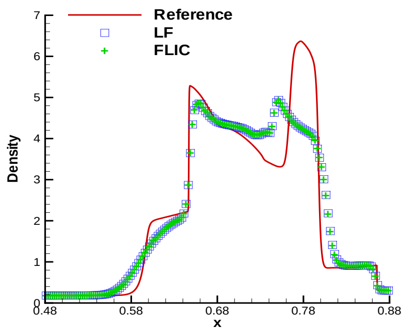
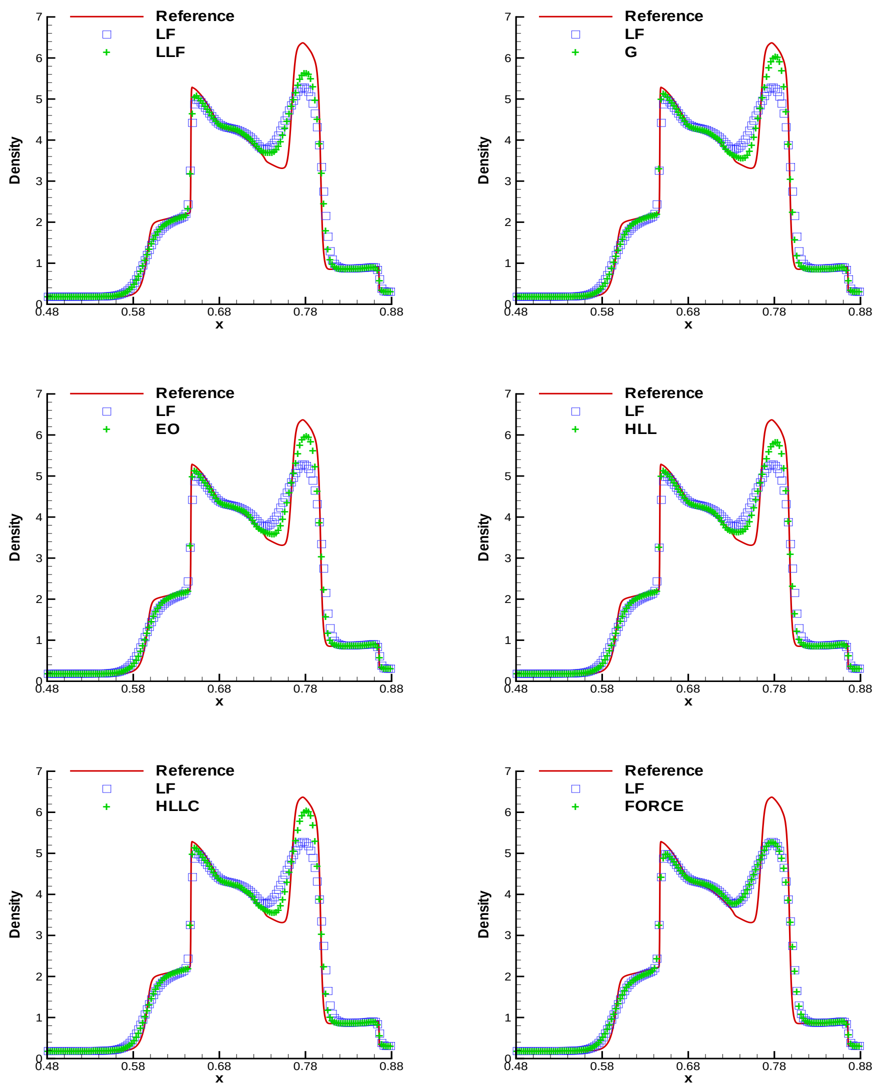


Figure 3: Blast wave problem for Navier-Stokes equations, Reynolds number Re=10000. $t = 0.038$. LWDG with different fluxes, k=1, 400 cells.Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the LWDG-LF scheme; plus symbols: results computed by the LWDG-LLF, LWDG-G, LWDG-EO, LWDG-HLL, LWDG-HLLC, LWDG-FORCE, LWDG-FLIC schemes (from top left to bottom left).



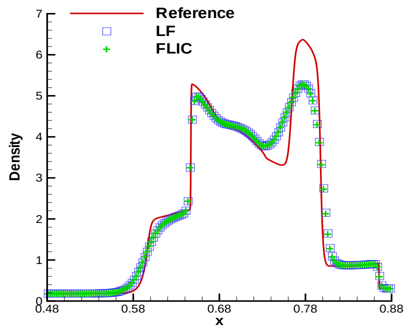


Figure 4: Blast wave problem for Navier-Stokes equations, Reynolds number Re=10000. t = 0.038. LWDG with different fluxes, k=2, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the LWDG-LF scheme; plus symbols: results computed by the LWDG-LLF LWDG-G, LWDG-EO, LWDG-HLL, LWDG-HLLC, LWDG-FORCE, LWDG-FLIC schemes (from top left to bottom left).

Table 9: CPU time (in hours) for the LWDG methods to compute the double Mach reflection problem for the two meshes of 480×120 and 960×240 cells.

$N_x\times N_y$	480×120		960×240		1920×480	
Flux	$k=1$	$k=2$	$k=1$	$k=2$	$k=1$	$k=2$
LF	0.4531	1.2406	3.0993	10.1298	22.4780	100.8831
HLL	0.6528	1.6633	3.9091	13.0633	30.9580	128.0188
HLLC	0.5945	1.4230	3.7654	12.4587	25.9656	129.2997

"exact" solution, computed using a fifth-order WENO scheme [22] with 5000 grid points, and against the solution computed by the LWDG-LF scheme on the same mesh, zoomed at the region $0.53\leq x\leq 0.88$ which contains the contact discontinuities and shocks in the solution.

The resolution of the LWDG-LF scheme is the worst among all schemes. For the case of $k=1$, the resolution of the LWDG-G, LWDG-EO and LWDG-HLLC schemes is the best, followed closely by that of the LWDG-HLL scheme, and the resolution of these four schemes is much better than that of the other four schemes. The resolution of the LWDG-LLF, LWDG-FORCE and LWDG-FLIC schemes is similar to that of the LWDG-LF scheme. For the case $k=2$, we also observe that the resolution of the LWDG-G, LWDG-EO and LWDG-HLLC schemes is the best, followed closely by that of the LWDG-HLL and LWDG-LLF schemes, and the resolution of these five schemes is much better than that of the other three schemes. The resolution of the LWDG-FORCE and LWDG-FLIC schemes is similar to that of the LWDG-LF scheme.

Example 3.3. Double Mach reflection. This problem is originally from [32]. The computational domain for this problem is $[0,4]\times[0,1]$. The reflecting wall lies at the bottom, starting from $x=\frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned at $x=\frac{1}{6}, y=0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x=0$ to $x=\frac{1}{6}$ and a reflective boundary condition is used for the rest. At the top boundary, the flow values are set to describe the exact motion of a Mach 10 shock. We compute the solution up to $t=0.2$, and TVB constant $M=100.0$. Based on our numerical experimental results for the one dimensional case, we only test the schemes LWDG-LF, LWDG-HLL and LWDG-HLLC which seem to be the good choice when all factors such as the cost of CPU time, numerical errors and resolution of discontinuities in the solution are considered.

The LWDG methods with WENO limiters, for three uniform meshes, with 480×120, 960×240 and 1920×480 cells, and two different orders of accuracy (for k=1 and k=2, second and third order), are used in the numerical experiments. In Table 9 we again document the CPU time by the LWDG schemes with different fluxes. We can see that the LWDG-HLL scheme costs about 30% more CPU time than the LWDG-LF scheme, and the LWDG-HLLC scheme costs about 15-30% more CPU time than the LWDG-LF scheme, for the same accuracy order and same mesh. To save space, we show only the simulation results on the most refined mesh with 1920×480 cells in Figs. 5 and 6, and the "zoomed-

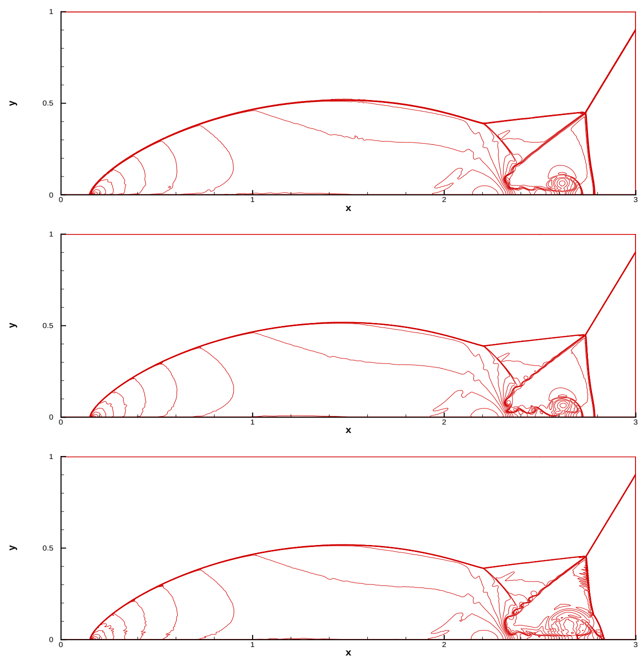


Figure 5: Double Mach reflection problem. 1920×480 cells. 30 equally spaced density contours from 1.5 to 22.7. k=1. From top to bottom are LWDG with LF, HLL and HLLC fluxes, respectively.

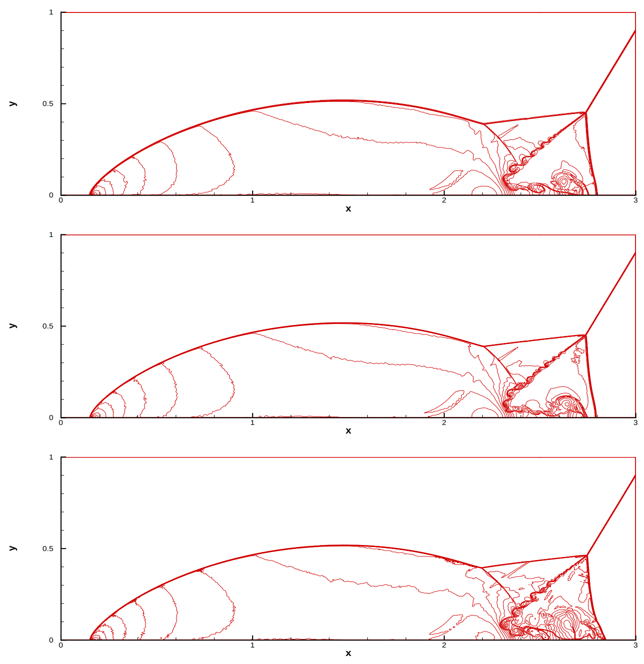


Figure 6: Double Mach reflection problem. 1920×480 cells. 30 equally spaced density contours from 1.5 to 22.7. k=2. From top to bottom are LWDG with LF, HLL and HLLC fluxes, respectively.

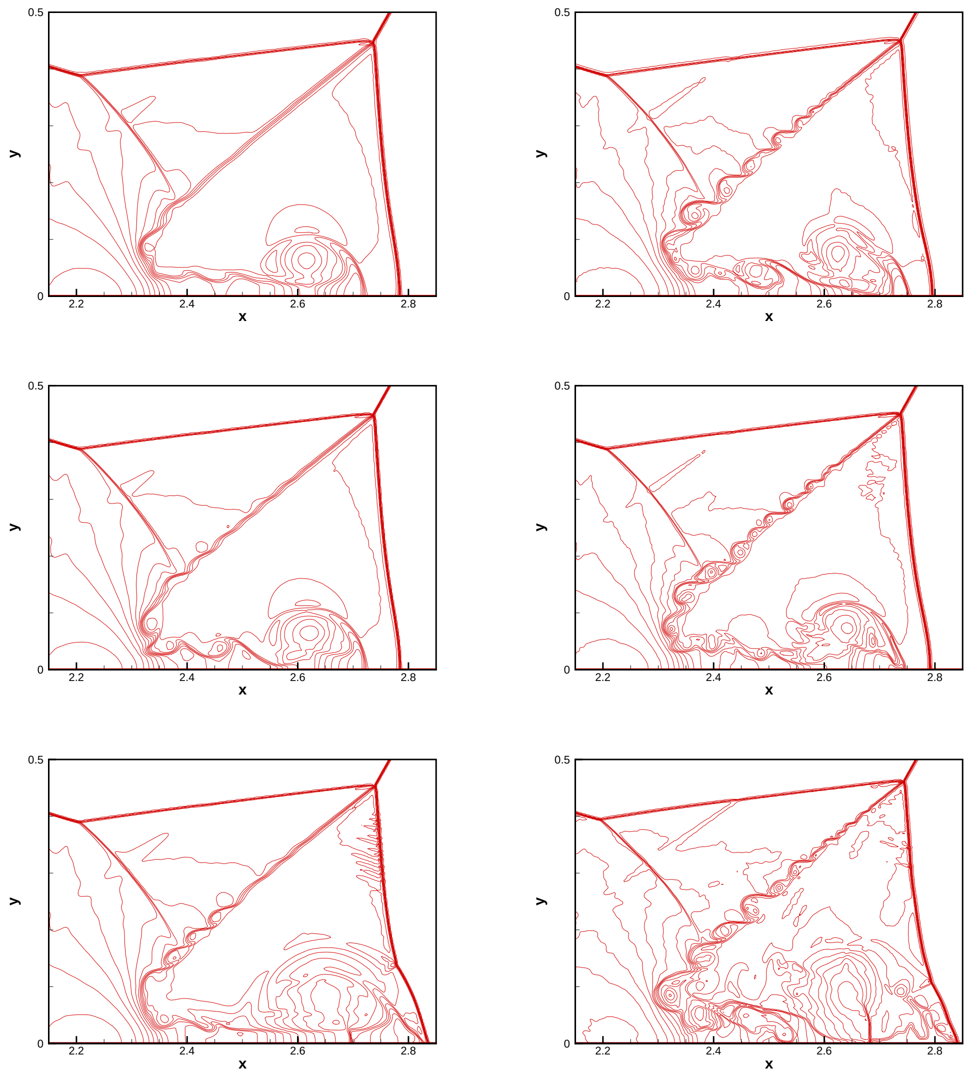


Figure 7: Double Mach reflection problem. 1920×480 cells. 30 equally spaced density contours from 1.5 to 22.7. Zoomed-in region to show more details. TVB constant k=1 (left) and k=2 (right). From top to bottom are LWDG with LF, HLL and HLLC fluxes, respectively.

in" figures around the double Mach stem to show more details in Fig. 7. All the figures are showing 30 equally spaced density contours from 1.5 to 22.7.

4. Concluding remarks

In this paper, we developed a new LWDG methods based on the local DG methods and Lax-Wendroff time procedure adopted in [22] for convection-diffusion equations, and numerical fluxes for finite volume or finite difference schemes are extended to numerical fluxes used for LWDG methods. We have systematically studied and compared a few different fluxes for the LWDG methods. Extensive one and two dimensional simulations on the hyperbolic systems of Euler and Navier-Stokes equations indicate that LWDG methods with the LF flux cost the least CPU time among all, but the resolution of solutions on the discontinuities are also the worst among all. The numerical errors of the LWDG method with the LF flux for a smooth problem seem to be the smallest for $k=1$ and largest for $k=2$ among all LWDG schemes. The LWDG methods with the Godunov or EO fluxes seem to cost significantly more CPU time than the LWDG-LF method. Similar to [21, 24], the HLLC flux might be the best choice, and the HLL flux is the next, as fluxes for the LWDG method when all factors such as the cost of CPU time, numerical errors and resolution of discontinuities in the solution are considered.

We also tested the Sod, the Lax and shock interaction with entropy waves problems [27], the relation of CPU times by the LWDG with different fluxes is similar to these of the blast wave problem; but unlike to the blast wave problem, it is difficult to differ the performance of resolution by the LWDG with different fluxes by eye, they are not shown in the paper to save space.

We have also tested the LWDG methods based on a few other numerical fluxes, such as the second-order Lax-Wendroff (LW) flux and the Warming-Beam (WB) flux. Our numerical tests indicate that spurious oscillations appear for the Lax shock tube problem for the LWDG-LW and LWDG-WB schemes, and the codes are unstable (they blow up) for the blast wave test case.

Acknowledgments The research of the author was partially supported by the European project ADIGMA on the development of innovative solution algorithms for aerodynamic simulations, NSFC grant 10671091, SRF for ROCS, SEM and JSNSF BK2006511. The author would like to thank the referees for the helpful suggestions.

References

[1] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations.* J. Comput. Phys. 131 (1997), 267-279.

[2] M. Ben-Artzi and J. Falcovitz, *A second-order Godunov-type scheme for compressible fluid dynamics*, J. Comput. Phys., 55 (1984), pp.1-32.

[3] 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), 545-581.

[4] 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), 90-113.

[5] 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), 411-435.

[6] B. Cockburn and C.-W. Shu, *The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws*, Math. Model. Numer. Anal. (M^2AN), 25 (1991), 337-361.

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

[8] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection diffusion systems.* SIAM J. Numer. Anal. 35 (1998), 2440-2463.

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

[10] M. Dumbser and C.-D. Munz, *Arbitrary high order discontinuous Galerkin schemes*, in *Numerical Methods for Hyperbolic and Kinetic Problems*, S. Cordier, T. Goudon, E. Sonnendrcker (editors), EMS Publishing House, 295-333, 2005.

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

[12] B. Engquist and S. Osher, *One sided difference approximation for nonlinear conservation laws*, Math. Comp., 36 (1981), 321-351.

[13] S.K. Godunov, *Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics*, Math. Sbornik, 47 (1959), 271-306.

[14] A. Harten, B. Engquist, S. Osher and S. Chakravathy, *Uniformly high order accurate essentially non-oscillatory schemes, III*, J. Comput. Phys., 71 (1987), 231-303.

[15] A. Harten, P. D. Lax and B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Review, 25 (1983), 35-61.

[16] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon and J.E. Flaherty, *Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws*, Appl. Numer. Math., 48 (2004), 323-338.

[17] P.D. Lax and B. Wendroff, *Systems of conservation laws*, Commu. Pure Appl. Math., 13 (1960), 217-237.

[18] S. Osher and F. Solomon, *Upwind difference schemes for hyperbolic conservation laws*, Math. Comp., 38 (1982), 339-374.

[19] J. de la Puente, M. Käser, M. Dumbser and H. Igel, *An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes IV: anisotropy,* Geophysical Journal International, 169 (2007), 1210-1228.

[20] 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), 4528-4543.

[21] J. Qiu, B.C. Khoo and C.-W. Shu, *A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes,* J. Comput. Phys., 212 (2006), 540-565.

[22] J. Qiu and C.-W. Shu, *Finite difference WENO schemes with Lax-Wendroff-type time discretizations*, SIAM J. Sci. Comput., 24 (2003), 2185-2198.

[23] J. Qiu and C.-W. Shu, *Runge-Kutta discontinuous Galerkin method using WENO limiters*, SIAM J. Sci. Comput., 26 (2005), 907-929.

[24] J. Qiu, *A numerical comparison of the Lax-Wendroff discontinuous Galerkin method based on different numerical fluxes,* J. Sci. Comput., 30 (2007), 345-367.

[25] C.-W. Shu, *Total-Variation-Diminishing time discretizations*, SIAM J. Sci. Stat. Comput., 9 (1988), 1073-1084.

[26] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comp., 49 (1987), 105-121.

[27] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing*

schemes II, J. Comput. Phys., 83 (1989), 32-78.

[28] M.E. Taylor, *Partial Differential Equation I: Basic Theory*, Applied Mathematical Sciences 115, Springer, 1996.

[29] V.A. Titarev and E.F. Toro, *ADER: arbitrary high order Godunov approach*, J. Sci. Comput., 17 (2002), 609-618.

[30] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics, a Practical Introduction*, Springer, Berlin, 1997.

[31] E.F. Toro, M. Spruce and W. Speares, *Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver*, J. Shock Waves 4 (1994), 25-34.

[32] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), 115-173.