Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 212 (2006) 540–565
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A numerical study for the performance of the Runge–Kutta discontinuous Galerkin method based on different numerical fluxes

Jianxian Qiu [a,b,1], Boo Cheong Khoo [b,2], Chi-Wang Shu [c,*,3]

[a] *Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, PR China*
[b] *Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore*
[c] *Division of Applied Mathematics, Brown University, Box F, Providence, RI 02912, USA*

Received 11 May 2005; accepted 13 July 2005
Available online 29 August 2005

Abstract

Runge–Kutta discontinuous Galerkin (RKDG) method is a high order finite element method for solving hyperbolic conservation laws employing useful features from high resolution finite volume schemes, such as the exact or approximate Riemann solvers serving as numerical fluxes, TVD Runge–Kutta time discretizations, and limiters. In most of the RKDG papers in the literature, the Lax–Friedrichs numerical flux is used due to its simplicity, although there are many other numerical fluxes which could also be used. In this paper, we systematically investigate the performance of the RKDG method based on different numerical fluxes, including the first-order monotone fluxes such as the Godunov flux, the Engquist–Osher flux, etc., and second-order TVD fluxes, 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: 65M60; 65M99; 35L65

* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: jxqiu@nju.edu.cn (J. Qiu), mpekbc@nus.edu.sg (B.C. Khoo), shu@dam.brown.edu (C.-W. Shu).

[1] Research partially supported by NNSFC Grant 10371118, Nanjing University Talent Development Foundation and NUS Research Project R-265-000-118-112.
[2] Research partially supported by NUS Research Project R-265-000-118-112.
[3] Research partially supported by the Chinese Academy of Sciences while the author was in residence at the University of Science and Technology of China (Grant 2004-1-8) and at the Institute of Computational Mathematics and Scientific/Engineering Computing. Additional support is provided by ARO Grant W911NF-04-1-0291, NSF Grant DMS-0207451 and AFOSR Grant FA9550-05-1-0123.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.07.011

Keywords: Runge–Kutta discontinuous Galerkin method; Numerical flux; Approximate Riemann solver; Limiter; WENO finite volume scheme; High order accuracy

1. Introduction

In this paper, we investigate the performance of the Runge–Kutta discontinuous Galerkin (RKDG) method [4,3,2,1,5,6] based on different numerical fluxes for solving nonlinear hyperbolic conservation laws

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

with the objective of obtaining better performance by choosing suitable numerical fluxes.

The first discontinuous Galerkin (DG) method was introduced in 1973 by Reed and Hill [17], in the framework of neutron transport (steady state linear hyperbolic equations). A major development of the DG method was carried out by Cockburn et al. in a series of papers [4,3,2,1,5], in which they established a framework to easily solve *nonlinear* time dependent hyperbolic conservation laws (1.1) using explicit, nonlinearly stable high order Runge–Kutta time discretizations [20] and DG discretization in space with exact or approximate Riemann solvers as interface fluxes and total variation bounded (TVB) limiter [18] to achieve non-oscillatory properties for strong shocks. These schemes are termed RKDG methods. The RKDG method is a high order finite element method for solving hyperbolic conservation laws using also ideas from high resolution finite volume schemes, such as the exact or approximate Riemann solvers as numerical fluxes, TVD Runge–Kutta time discretizations, and limiters. RKDG methods have the advantage of flexibility in handling complicated geometry, *h–p* adaptivity, and efficiency of parallel implementation and has been used successfully in many applications. We refer to the recent special issue [7] for information on the recent development of DG methods.

An important component of the RKDG 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 RKDG papers in the literature, the 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, which could also be used in the context of RKDG methods. The Godunov flux [9,24] 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 Engquist–Osher (EO) flux [8,14,24] for the scalar case and its extension to systems (often referred to as the Osher–Solomon flux [14]) are smoother than the Godunov flux with an 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. The derivation of the EO flux depends on the integration in the phase space. Because of the existence of the explicit formulas, the evaluation of the EO flux is less costly than the Godunov flux, but is still more costly than other simpler approximate Riemann solvers. In 1983, Harten, Lax and van Leer presented the HLL flux [11] based on the approximate Riemann solver with only three constant states separated by two waves. The evaluation of the HLL flux is simple and fast, however it has the shortcoming of poor resolution for contact discontinuities, shear waves and material interfaces. In [26], a modification of the HLL flux, often referred to as the HLLC flux, was presented to overcome this defect of the HLL flux by restoring the missing contact and shear waves. The fluxes mentioned above are all two point, first order monotone fluxes. These fluxes have the form $\hat{f}(u^-, u^+)$ and are consistent with the physical flux in the sense that $\hat{f}(u,u) = f(u)$. There are also certain second order TVD (total variation diminishing) fluxes, which may depend on more than two points, e.g. $\hat{f}(u^l, u^-, u^+, u^r)$, but have the following *essentially two point* property: $\hat{f}(u^l, u, u, u^r) = f(u)$ for any u^l and u^r, which can also be used as numerical fluxes for the RKDG methods. An example of the essentially two point TVD fluxes is the flux limiter centered (FLIC) flux [24],

which combines a low order monotone flux and a high order flux with a flux limiter to guarantee the TVD property. Recently, Titarev and Toro initialized an approach to use second order essentially two point TVD fluxes [23], and the MUSTA and HLLC fluxes [22], instead of the first order monotone fluxes, as building blocks for Godunov type finite volume schemes. Their numerical results show an improvement, sometimes dramatic, on numerical resolutions when such fluxes are used.

In this paper, we systematically study and compare the performance of the RKDG method based on different numerical fluxes, with the objective of obtaining better performance by choosing suitable numerical fluxes. We review and describe the details of the fluxes under consideration in Section 2, and present extensive numerical experiments in Section 3 to compare their performance. Concluding remarks are given in Section 4.

2. Review and implementation of the numerical fluxes for the RKDG methods

In this section, we review the numerical fluxes under consideration for the RKDG methods. We start with the description of the RKDG method in the one dimensional case and use the notations in [3], however we emphasize that the procedure described below does not depend on the specific basis chosen for the polynomials and works also in multi-dimensions. We would like to solve the one dimensional scalar conservation law

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

The computational domain is divided into N cells with boundary points $0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = L$. The points x_i are the centers of the cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and we denote the cell sizes by $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and the maximum cell size by $h = \max_i \Delta x_i$. The solution as well as the test function space is given by $V_h^k = \{p : \ p|_{I_i} \in P^k(I_i)\}$, where $P^k(I_i)$ is the space of polynomials of degree $\leqslant k$ on the cell I_i. We adopt a local orthogonal basis over I_i, $\{v_l^{(i)}(x), l = 0, 1, \ldots, k\}$, namely the scaled Legendre polynomials:

$$v_0^{(i)}(x) = 1, \qquad v_1^{(i)}(x) = \frac{x - x_i}{\Delta x_i/2}, \qquad v_2^{(i)}(x) = \left(\frac{x - x_i}{\Delta x_i/2}\right)^2 - \frac{1}{3}, \ldots$$

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

$$u^h(x,t) = \sum_{l=0}^{k} u_i^{(l)}(t) v_l^{(i)}(x), \quad \text{for } x \in I_i \tag{2.2}$$

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

$$u_i^{(l)}(t) = \frac{1}{a_l} \int_{I_i} u^h(x,t) v_l^{(i)}(x) \mathrm{d}x, \quad l = 0, 1, \ldots, k,$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t} u_i^{(l)} + \frac{1}{a_l} \left(- \int_{I_i} f(u^h(x,t)) \frac{\mathrm{d}}{\mathrm{d}x} v_l^{(i)}(x) \mathrm{d}x + \hat{f}(u_{i+1/2}^-, u_{i+1/2}^+) v_l^{(i)}(x_{i+1/2}) - \hat{f}(u_{i-1/2}^-, u_{i-1/2}^+) v_l^{(i)}(x_{i-1/2}) \right) = 0, \quad l = 0, 1, \ldots, k, \tag{2.3}$$

where $u_{i+1/2}^{\pm} = u^h(x_{i+1/2}^{\pm}, t)$ are the left and right limits of the discontinuous solution u^h at the cell interface $x_{i+1/2}$, and $\hat{f}(u^-, u^+)$ is the numerical flux based on an exact or approximate Riemann solver, which will be the main focus of discussion for this paper. The semidiscrete scheme (2.3) is discretized in time by a nonlinearly stable Runge–Kutta time discretization, e.g. the third order version [20]. The integral term in (2.3) can be computed either exactly or by a suitable numerical quadrature accurate to at least $O(h^{k+l+2})$.

In order to maintain stability and non-oscillatory property of the RKDG 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 shock detection technique by Krivodonova et al. in [13] to detect troubled-cells (we refer to [16] for a detailed investigation of various troubled-cell indicators), where a WENO limiter developed in [15] will be used for the reconstruction of first and higher order moments of the polynomials inside those troubled cells. We refer to [15] 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 component-wise indicator. For both the one dimensional and the two dimensional Euler equations, we use only the components of density and energy as indicator variables. We emphasize that the component-wise strategy is used only to identity troubled cells. Once these cells are identified, the WENO reconstructions in them are performed in local characteristic directions. We again refer to [15] and to [19] for more details of the reconstruction.

We now review the two point or essentially two point numerical fluxes under consideration. Numerical experiments to compare their performance for the RKDG method will be given in next section.

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

$$u = (\rho, \rho v, E)^{\mathrm{T}}, \qquad f(u) = (\rho v, \rho v^2 + p, v(E+p))^{\mathrm{T}}, \tag{2.4}$$

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. We will also use the sound speed $c = \sqrt{\gamma p/\rho}$ in the definition of some of the numerical fluxes.

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 RKDG 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

$$\hat{f}^{\mathrm{LF}}(u^-, u^+) = \frac{1}{2}[(f(u^-) + f(u^+)) - \alpha(u^+ - u^-)], \tag{2.5}$$

   where 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 between u^- and u^+.
2. *The Godunov flux*. The Godunov flux [9,24] 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

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

   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:

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

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

$$\hat{f}^{\mathrm{G}}(u^-, u^+) = \begin{cases} \min\limits_{u^- \leqslant u \leqslant u^+} f(u) & \text{if } u^- \leqslant u^+, \\ \max\limits_{u^+ \leqslant u \leqslant u^-} f(u) & \text{if } u^- > u^+. \end{cases} \tag{2.6}$$

   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 [24] and the references therein

for the details of the exact Riemann solver for systems in applications, such as the Euler equations (2.4), which are needed for the evaluation of the Godunov flux for such systems.

3. *The Engquist–Osher (EO) flux and the Osher–Solomon flux [8,14].* The Engquist–Osher (EO) flux [8] for the scalar case and its extension to systems (often referred to as the Osher–Solomon flux [14]) are smoother than the Godunov flux with an 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

$$\hat{f}^{\mathrm{EO}}(u^-,u^+)=\frac{1}{2}\left(f(u^-)+f(u^+)-\int_{u^-}^{u^+}|f'(u)|\mathrm{d}u\right). \tag{2.7}$$

For the system case, the explicit formulas for the Osher–Solomon flux for the Euler equations (2.4) is given as follows [14]: First we compute intermediate variables based on $u^{\pm}$:

$$\rho_1=\rho^+\left(\frac{(\gamma-1)(v^+-v^-)/2+c^++c^-}{c^+(1+(p^-/p^+)^{1/2\gamma}(\rho^-/\rho^+)^{-1/2})}\right)^{2/(\gamma-1)},$$

$$\rho_2=\rho^-\left(\frac{(\gamma-1)(v^+-v^-)/2+c^++c^-}{c^-(1+(p^+/p^-)^{1/2\gamma}(\rho^+/\rho^-)^{-1/2})}\right)^{2/(\gamma-1)},$$

$$p_1=p_2=p^-\left(\frac{\rho^-}{\rho_2}\right)^{-\gamma},\quad v_1=v_2=v^--\frac{2}{\gamma-1}\left(c^--\sqrt{\frac{\gamma p_2}{\rho_2}}\right),$$

$$\bar{\rho}_1=\rho^+\left(\frac{(\gamma-1)v^++2c^+}{(\gamma+1)c^+}\right)^{2/(\gamma-1)},\quad \bar{\rho}_2=\rho^-\left(\frac{-(\gamma-1)v^-+2c^-}{(\gamma+1)c^-}\right)^{2/(\gamma-1)},$$

$$\bar{p}_1=p^+(\bar{\rho}_1/\rho^+)^{\gamma},\quad \bar{p}_2=p^-(\bar{\rho}_2/\rho^-)^{\gamma},$$

$$\bar{v}_1=v^++\frac{2}{\gamma-1}(c^+-\sqrt{\gamma\bar{p}_1/\bar{\rho}_1}),\quad \bar{v}_2=v^--\frac{2}{\gamma-1}(c^--\sqrt{\gamma\bar{p}_2/\bar{\rho}_2}).$$

The Osher–Solomon flux is then still given by (2.7), with the integral computed as a sum of three parts based on three characteristic fields:

$$\int_{u^-}^{u^+}|f'(u)|\mathrm{d}u=\int_{\Gamma_1}+\int_{\Gamma_2}+\int_{\Gamma_3}(f'(u)^+-f'(u)^-)\mathrm{d}u, \tag{2.8}$$

where

$$\int_{\Gamma_1}f'(u)^+\,\mathrm{d}u=f(u)\begin{cases}u^+ & \text{if } \rho^+<\bar{\rho}_1,\\ \bar{u}_1 & \text{if } \rho^+\geqslant\bar{\rho}_1,\\ & \\ u_1 & \text{if } \rho_1<\bar{\rho}_1,\\ \bar{u}_1 & \text{if } \rho_1\geqslant\bar{\rho}_1.\end{cases},\qquad \int_{\Gamma_1}f'(u)^-\,\mathrm{d}u=f(u)\begin{cases}u^+ & \text{if } \rho^+\geqslant\bar{\rho}_1,\\ \bar{u}_1 & \text{if } \rho^+<\bar{\rho}_1,\\ & \\ u_1 & \text{if } \rho_1\geqslant\bar{\rho}_1,\\ \bar{u}_1 & \text{if } \rho_1<\bar{\rho}_1.\end{cases}$$

$$\int_{\Gamma_2}(f'(u)^+-f'(u)^-)\mathrm{d}u=(\rho_1-\rho_2)\,|v_1|\begin{pmatrix}1\\ v_1\\ v_1^2/2\end{pmatrix}.$$

$$\int_{\Gamma_3}f'(u)^+\,\mathrm{d}u=f(u)\begin{cases}u_2 & \text{if } \rho_2>\bar{\rho}_2,\\ \bar{u}_2 & \text{if } \rho_2\leqslant\bar{\rho}_2,\\ & \\ u^- & \text{if } \rho^->\bar{\rho}_2,\\ \bar{u}_2 & \text{if } \rho^-\leqslant\bar{\rho}_2.\end{cases},\qquad \int_{\Gamma_3}f'(u)^-\,\mathrm{d}u=f(u)\begin{cases}u_2 & \text{if } \rho_2\leqslant\bar{\rho}_2,\\ \bar{u}_2 & \text{if } \rho_2>\bar{\rho}_2,\\ & \\ u^- & \text{if } \rho^-\leqslant\bar{\rho}_2,\\ \bar{u}_2 & \text{if } \rho^->\bar{\rho}_2.\end{cases}$$

4. *The Harten–Lax–van Leer (HLL) flux [11,24].* The HLL flux [11] is based on the approximate Riemann solver with only three constant states separated by two waves. The HLL flux for the Euler equations (2.4) is given by

$$\hat{f}^{\mathrm{HLL}}(u^-,u^+)=\begin{cases} f(u^-) & \text{if } 0\leqslant s^-,\\ \frac{s^+f(u^-)-s^-f(u^+)+s^-s^+(u^+-u^-)}{s^+-s^-} & \text{if } s^-\leqslant 0\leqslant s^+,\\ f(u^+) & \text{if } s^+\leqslant 0,\end{cases} \tag{2.9}$$

where the lower and upper bounds of the wave speed, s^- and s^+, must be estimated. We use the pressure-velocity estimates [24]

$$s^-=v^--c^-q^-,\qquad s^*=v^*,\qquad s^+=v^++c^+q^+, \tag{2.10}$$

where, for $K=\pm$,

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

with

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

and

$$\bar{\rho}=\frac{1}{2}(\rho^-+\rho^+),\qquad \bar{c}=\frac{1}{2}(c^-+c^+).$$

5. *The HLLC flux – a modification of the HLL flux [26,24].* 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 (2.4) is given by

$$\hat{f}^{\mathrm{HLLC}}(u^-,u^+)=\begin{cases} f(u^-) & \text{if } \ \leqslant s^-,\\ f(u^-)+s^-(u^{*-}-u^-) & \text{if } s^-\leqslant 0\leqslant s^*,\\ f(u^+)+s^+(u^{*+}-u^+) & \text{if } s^*\leqslant 0\leqslant s^+,\\ f(u^+) & \text{if } s^+\leqslant 0,\end{cases} \tag{2.11}$$

where, for $K=\pm$,

$$u^{*K}=\rho^K\frac{s^K-v^K}{s^K-s^*}\begin{bmatrix}1\\ s^*\\ \frac{E^K}{\rho^K}+(s^*-v^K)[s^*+\frac{p^K}{\rho^K(s^K-v^K)}]\end{bmatrix}. \tag{2.12}$$

The definitions of s^-, s^* and s^+ are given in (2.10).

6. *The first-order centered (FORCE) flux [24].* The FORCE flux is given by

$$\hat{f}^{\mathrm{FORCE}}(u^-,u^+)=\frac{1}{2}(\hat{f}^{\mathrm{LF}}(u^-,u^+)+\hat{f}^{\mathrm{R}}(u^-,u^+)), \tag{2.13}$$

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

$$\hat{f}^{\mathrm{R}}(u^-,u^+)=f(u^*),\qquad u^*=\frac{1}{2}\left(u^-+u^+-\frac{\Delta t}{\Delta x}(f(u^+)-f(u^-))\right). \tag{2.14}$$

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 [24].* 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:

$$\hat{f}^{\mathrm{FLIC}}(u^-,u^+)=\hat{f}^{\mathrm{FORCE}}(u^-,u^+)+\phi_{i+1/2}[\hat{f}^{\mathrm{R}}(u^-,u^+)-\hat{f}^{\mathrm{FORCE}}(u^-,u^+)], \tag{2.15}$$

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. Following [24,22], 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 $\Delta q_{i-1/2}=\bar{q}_i-\bar{q}_{i-1}$, and $\bar{q}_i$ 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:

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

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

8. *The multi-stage predictor-corrector (MUSTA) flux [25].* The MUSTA flux is a multi-stage predictor-corrector flux. Following [22] we use the FORCE flux as the predictor flux. The procedure to evaluate a L-stage MUSTA flux can be described as following: first we set $u^-_0=u^-$ and $u^+_0=u^+$ for the initial stage $l=0$, then we perform the following steps:
   (a) Compute the FORCE flux $\hat{f}^{\mathrm{FORCE}}_l=\hat{f}^{\mathrm{FORCE}}(u^-_l,u^+_l)$ on the data at the stage l.
   (b) If the desired number of total stages L has been reached (that is $l=L$), then the computation of the MUSTA flux is complete and the final flux is given by $\hat{f}^{\mathrm{MUSTA}}(u^-,u^+)=\hat{f}^{\mathrm{FORCE}}_l$. Otherwise, continue to compute the values for the next stage using:

$$u^-_{l+1}=u^-_l-\frac{\Delta t}{\Delta x}(\hat{f}^{\mathrm{FORCE}}_l-f(u^-_l)),\qquad u^+_{l+1}=u^+_l-\frac{\Delta t}{\Delta x}(f(u^+_l)-\hat{f}^{\mathrm{FORCE}}_l),$$

   and proceed back to step (a).
   In this paper, we use $L=2$ as suggested in [22].

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

3. Numerical results

In this section, we perform extensive numerical experiments to compare the performance of the RKDG 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 Compaq Digital personal workstation, 600 au alpha-599 MHZ with 256 MB ram. We denote the RKDG scheme with the flux "X" as RKDG-X, such as RKDG-LF for the RKDG scheme with the LF flux. In our numerical experiments, the CFL numbers are taken as 0.3, 0.18 and 0.1 for $k = 1$, $k = 2$ and $k = 3$ (second, third and fourth order accuracy), respectively.

Example 3.1. We solve the one dimensional nonlinear system of Euler equations (2.4). 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 is $\rho(x, t) = 1 + 0.2\sin(\pi(x - t))$, $v(x,t) = 1$, $p(x,t) = 1$. We compute the solution up to $t = 2$. In Table 1, we provide a CPU time comparison for the RKDG schemes with different fluxes. The numerical errors and the orders of accuracy for the density ρ, and ratios of the numerical errors for comparison with the RKDG-LF scheme are shown in Tables 2–4.

We can see that the RKDG-LF scheme costs the least CPU time for each of the cases $k = 1, 2$ and 3, but at the same it has the largest numerical errors.

On the CPU time, the RKDG-G and RKDG-EO schemes cost about twice that of the RKDG-LF scheme, the RKDG-HLL and RKDG-HLLC schemes cost about 10–20% more than that of the RKDG-LF scheme, the RKDG-FORCE and RKDG-FLIC schemes cost about the same as that of the RKDG-LF scheme, and the RKDG-MUSTA scheme costs about 15–25% more than that of the RKDG-LF scheme. 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 CPU costs of the RKDG method using these different fluxes.

On the numerical errors, for the case of $k = 1$ and 3, the L_1 and L_∞ errors of all other schemes for the same meshes are about 40% and 50% of that by the RKDG-LF scheme, except for the RKDG-FLIC scheme, which has errors about 80% of that by the RKDG-LF scheme. For the case of $k = 2$, however, the RKDG-FLIC scheme has the smallest errors among all the schemes, which is about half of that by the RKDG-LF scheme. The errors by the other schemes for the $k = 2$ case are about 70% of that by the RKDG-LF scheme. 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.

We can also see that all schemes achieve their designed orders of accuracy, as expected.

Example 3.2. We repeat the numerical experiments of the previous example using the following Riemann initial condition for the Lax problem:

$$(\rho, v, p) = (0.445, 0.698, 3.528) \quad \text{for } x \leqslant 0; \quad (\rho, v, p) = (0.5, 0, 0.571) \quad \text{for } x > 0.$$

Table 1
CPU time (in seconds) for the RKDG methods with different fluxes, for the accuracy test problem

k	LF	G	EO	HLL	HLLC	FORCE	FLIC	MUSTA
1	4.31	10.89	9.34	5.12	5.24	4.34	4.51	5.55
2	11.36	22.83	19.71	12.53	12.70	11.40	11.51	13.45
3	111.16	191.70	171.50	120.70	120.84	112.24	113.37	127.65

Total CPU time for $N = 10$, 20, 40, 80 and 160 cells is recorded.

Table 2
Euler equations, $\rho(x, 0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, using N equally spaced cells with different fluxes, $t = 2$, L_1 and L_∞ errors of density ρ, $k = 1$

N	Flux	L_1 error	L_1 order	Error ratio	L_∞ error	L_∞ order	Error ratio
10	LF	8.4968E−03		1.0000	2.4595E−02		1.0000
	G	3.6256E−03		0.4267	8.2959E−03		0.3373
	EO	3.6256E−03		0.4267	8.2959E−03		0.3373
	HLL	3.3024E−03		0.3887	8.4497E−03		0.3435
	HLLC	3.6256E−03		0.4267	8.2959E−03		0.3373
	FORCE	3.0171E−03		0.3551	8.0013E−03		0.3253
	FLIC	5.6267E−03		0.6622	1.3839E−02		0.5627
	MUSTA	3.3987E−03		0.4000	8.1473E−03		0.3313
20	LF	1.9328E−03	2.1362	1.0000	5.4403E−03	2.1766	1.0000
	G	7.6077E−04	2.2527	0.3936	2.6850E−03	1.6275	0.4935
	EO	7.6077E−04	2.2527	0.3936	2.6850E−03	1.6275	0.4935
	HLL	6.7937E−04	2.2812	0.3515	2.5678E−03	1.7183	0.4720
	HLLC	7.6077E−04	2.2527	0.3936	2.6850E−03	1.6275	0.4935
	FORCE	6.1231E−04	2.3008	0.3168	2.4096E−03	1.7315	0.4429
	FLIC	1.4035E−03	2.0033	0.7262	3.8455E−03	1.8475	0.7069
	MUSTA	7.0238E−04	2.2747	0.3634	2.5642E−03	1.6678	0.4713
40	LF	4.4660E−04	2.1136	1.0000	1.5222E−03	1.8376	1.0000
	G	1.7313E−04	2.1356	0.3877	7.5026E−04	1.8394	0.4929
	EO	1.7313E−04	2.1356	0.3877	7.5026E−04	1.8394	0.4929
	HLL	1.5419E−04	2.1395	0.3453	7.2005E−04	1.8344	0.4730
	HLLC	1.7313E−04	2.1356	0.3877	7.5026E−04	1.8394	0.4929
	FORCE	1.4041E−04	2.1246	0.3144	6.5719E−04	1.8744	0.4317
	FLIC	3.4300E−04	2.0327	0.7680	1.1858E−03	1.6973	0.7790
	MUSTA	1.5991E−04	2.1350	0.3581	7.1467E−04	1.8432	0.4695
80	LF	1.0799E−04	2.0481	1.0000	3.9930E−04	1.9306	1.0000
	G	4.1311E−05	2.0673	0.3826	1.9684E−04	1.9304	0.4930
	EO	4.1311E−05	2.0673	0.3826	1.9684E−04	1.9304	0.4930
	HLL	3.6862E−05	2.0645	0.3414	1.8922E−04	1.9280	0.4739
	HLLC	4.1311E−05	2.0673	0.3826	1.9684E−04	1.9304	0.4930
	FORCE	3.3860E−05	2.0520	0.3136	1.7074E−04	1.9445	0.4276
	FLIC	8.3990E−05	2.0299	0.7778	3.2199E−04	1.8808	0.8064
	MUSTA	3.8043E−05	2.0716	0.3523	1.8644E−04	1.9386	0.4669

This is a demanding test case in terms of controlling spurious oscillations. The computational domain is $[-5, 5]$ with 200 cells, and the final time is $t = 1.3$. In Figs. 1–3, the computed densities ρ are plotted against the exact solution and against the numerical solution computed by the RKDG-LF scheme on the same mesh, zoomed at the region $1 \leqslant x \leqslant 4$ which contains the contact discontinuity and the shock.

From Figs. 1–3, we can see that the results computed by the RKDG-G, RKDG-EO and RKDG-HLLC schemes are slightly better than that computed by the RKDG-LF scheme, in terms of the resolution of the discontinuities, and the results computed by all other schemes are similar to that computed by the RKDG-LF scheme.

Example 3.3. A higher order scheme would show its advantage when the solution contains both shocks and complex smooth region structures. A typical example for this is the problem of shock interaction with entropy waves [21]. We solve the Euler equations (2.4) with a moving Mach = 3 shock interacting with sine waves in density, i.e. initially

Table 3
Euler equations, $\rho(x, 0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, using N equally spaced cells with different fluxes, $t = 2$, L_1 and L_∞ errors of density ρ, $k = 2$

N	Flux	L_1 error	L_1 order	Error ratio	L_∞ error	L_∞ order	Error ratio
10	LF	1.7763E−04		1.0000	1.1382E−03		1.0000
	G	1.3013E−04		0.7326	7.8672E−04		0.6912
	EO	1.3013E−04		0.7326	7.8672E−04		0.6912
	HLL	1.4522E−04		0.8175	8.5161E−04		0.7482
	HLLC	1.3013E−04		0.7326	7.8672E−04		0.6912
	FORCE	1.6055E−04		0.9038	9.1115E−04		0.8005
	FLIC	1.0245E−04		0.5767	6.1017E−04		0.5361
	MUSTA	1.4157E−04		0.7970	8.3600E−04		0.7345
20	LF	2.2682E−05	2.9692	1.0000	1.3684E−04	3.0562	1.0000
	G	1.6089E−05	3.0158	0.7093	1.0213E−04	2.9454	0.7464
	EO	1.6089E−05	3.0158	0.7093	1.0213E−04	2.9454	0.7464
	HLL	1.8110E−05	3.0034	0.7984	1.1146E−04	2.9336	0.8146
	HLLC	1.6089E−05	3.0158	0.7093	1.0213E−04	2.9454	0.7464
	FORCE	2.0259E−05	2.9864	0.8932	1.2082E−04	2.9149	0.8829
	FLIC	1.2322E−05	3.0556	0.5432	7.4390E−05	3.0360	0.5436
	MUSTA	1.7638E−05	3.0047	0.7776	1.0908E−04	2.9381	0.7972
40	LF	2.9794E−06	2.9285	1.0000	1.8952E−05	2.8520	1.0000
	G	1.9979E−06	3.0095	0.6706	1.2877E−05	2.9876	0.6794
	EO	1.9979E−06	3.0095	0.6706	1.2877E−05	2.9876	0.6794
	HLL	2.2603E−06	3.0022	0.7586	1.4136E−05	2.9791	0.7459
	HLLC	1.9979E−06	3.0095	0.6706	1.2877E−05	2.9876	0.6794
	FORCE	2.5415E−06	2.9948	0.8530	1.5330E−05	2.9784	0.8089
	FLIC	1.5203E−06	3.0188	0.5103	9.2356E−06	3.0098	0.4873
	MUSTA	2.1974E−06	3.0048	0.7375	1.3773E−05	2.9855	0.7267
80	LF	3.9343E−07	2.9209	1.0000	2.4632E−06	2.9437	1.0000
	G	2.4929E−07	3.0026	0.6336	1.6132E−06	2.9967	0.6549
	EO	2.4929E−07	3.0026	0.6336	1.6132E−06	2.9968	0.6549
	HLL	2.8235E−07	3.0009	0.7177	1.7690E−06	2.9984	0.7182
	HLLC	2.4929E−07	3.0026	0.6336	1.6132E−06	2.9968	0.6549
	FORCE	3.1783E−07	2.9993	0.8079	1.9237E−06	2.9944	0.7810
	FLIC	1.8934E−07	3.0053	0.4812	1.1524E−06	3.0026	0.4678
	MUSTA	2.7448E−07	3.0011	0.6977	1.7267E−06	2.9958	0.7010

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{for } x < -4; \\ (1 + \varepsilon\sin(5x), 0, 1) & \text{for } x \geqslant -4. \end{cases}$$

Here, we take $\varepsilon = 0.2$. The computational domain is $[-5, 5]$ with 300 cells, the final time is $t = 1.8$. In Figs. 4–6, the computed densities ρ are plotted against the reference "exact" solution, computed using a fifth order WENO scheme [12] using 2000 grid points, and against the solution computed by the RKDG-LF scheme on the same mesh, zoomed at the region $0.5 \leqslant x \leqslant 2.5$ which contains the complicated wave pattern in the smooth part of the solution.

Similar to the previous two examples, we can observe an improvement of the resolution for the complicated wave pattern in this example for all (other) schemes over the RKDG-LF scheme. The performance of all the other schemes are similar for this example.

Table 4
Euler equations, $\rho(x, 0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, using N equally spaced cells with different fluxes, $t = 2$, L_1 and L_∞ errors of density ρ, $k = 3$

N	Flux	L_1 error	L_1 order	Error ratio	L_∞ error	L_∞ order	Error ratio
10	LF	1.3960E−05		1.0000	7.4910E−05		1.0000
	G	5.0388E−06		0.3610	3.6676E−05		0.4896
	EO	5.0388E−06		0.3610	3.6676E−05		0.4896
	HLL	4.7196E−06		0.3381	3.5570E−05		0.4748
	HLLC	5.0388E−06		0.3610	3.6676E−05		0.4896
	FORCE	4.5041E−06		0.3227	3.2429E−05		0.4329
	FLIC	8.2216E−06		0.5889	4.8367E−05		0.6457
	MUSTA	4.7486E−06		0.3402	3.4585E−05		0.4617
20	LF	8.7938E−07	3.9886	1.0000	5.3615E−06	3.8045	1.0000
	G	3.1134E−07	4.0165	0.3540	2.3064E−06	3.9911	0.4302
	EO	3.1134E−07	4.0165	0.3540	2.3064E−06	3.9911	0.4302
	HLL	2.8769E−07	4.0361	0.3272	2.2228E−06	4.0002	0.4146
	HLLC	3.1134E−07	4.0165	0.3540	2.3064E−06	3.9911	0.4302
	FORCE	2.7194E−07	4.0499	0.3092	2.0056E−06	4.0152	0.3741
	FLIC	5.8630E−07	3.8097	0.6667	3.6178E−06	3.7408	0.6748
	MUSTA	2.8878E−07	4.0395	0.3284	2.1466E−06	4.0100	0.4004
40	LF	4.9053E−08	4.1641	1.0000	3.0392E−07	4.1409	1.0000
	G	1.9400E−08	4.0044	0.3955	1.4535E−07	3.9881	0.4782
	EO	1.9401E−08	4.0043	0.3955	1.4533E−07	3.9882	0.4782
	HLL	1.7868E−08	4.0091	0.3643	1.3999E−07	3.9890	0.4606
	HLLC	1.9401E−08	4.0043	0.3955	1.4533E−07	3.9882	0.4782
	FORCE	1.6859E−08	4.0117	0.3437	1.2548E−07	3.9985	0.4129
	FLIC	3.8059E−08	3.9453	0.7759	2.3856E−07	3.9227	0.7849
	MUSTA	1.7933E−08	4.0093	0.3656	1.3497E−07	3.9913	0.4441
80	LF	3.0871E−09	3.9900	1.0000	1.8645E−08	4.0268	1.0000
	G	1.2104E−09	4.0026	0.3921	9.1348E−09	3.9920	0.4899
	EO	1.2117E−09	4.0011	0.3925	9.1023E−09	3.9970	0.4882
	HLL	1.1148E−09	4.0025	0.3611	8.7668E−09	3.9971	0.4702
	HLLC	1.2117E−09	4.0011	0.3925	9.1023E−09	3.9970	0.4882
	FORCE	1.0516E−09	4.0030	0.3406	7.8457E−09	3.9994	0.4208
	FLIC	2.4004E−09	3.9869	0.7775	1.5130E−08	3.9789	0.8115
	MUSTA	1.1186E−09	4.0029	0.3623	8.4466E−09	3.9982	0.4530

Example 3.4. We consider the interaction of blast waves of the Euler equation (2.4) with the initial condition:

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

A reflecting boundary condition is applied to both ends. See [27,10]. The computational domain is $[0, 1]$ with 400 cells. The final time is $t = 0.038$. In Figs. 7–9, the computed densities ρ are plotted against the reference "exact" solution, computed using a fifth order WENO scheme [12] using 2000 grid points, and against the solution computed by the RKDG-LF scheme on the same mesh, zoomed at the region $0.53 \leqslant x \leqslant 0.88$ which contains the contact discontinuities and shocks in the solution.

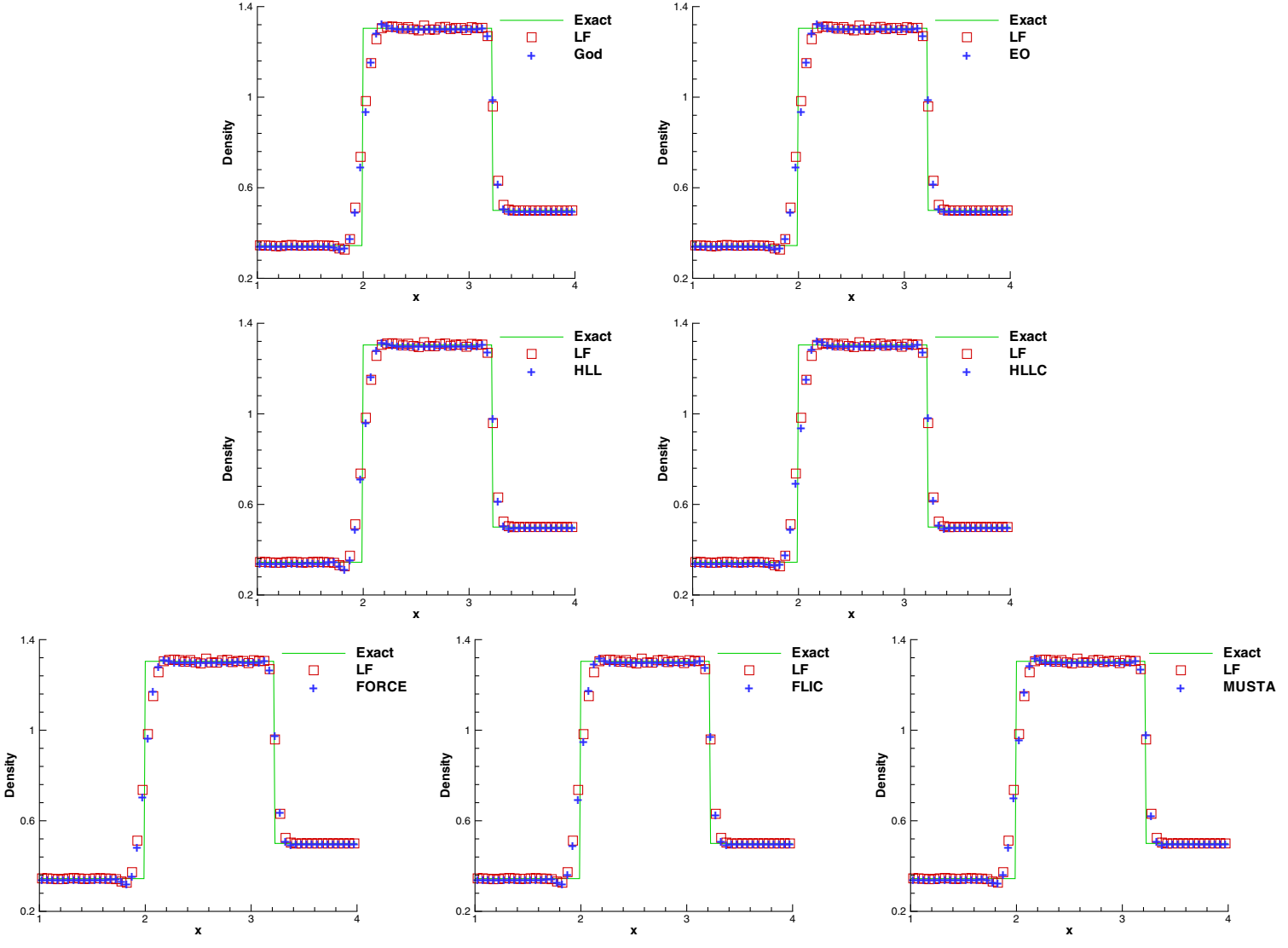


Fig. 1. Lax problem. $t = 1.3$. RKDG with different fluxes, $k = 1$, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

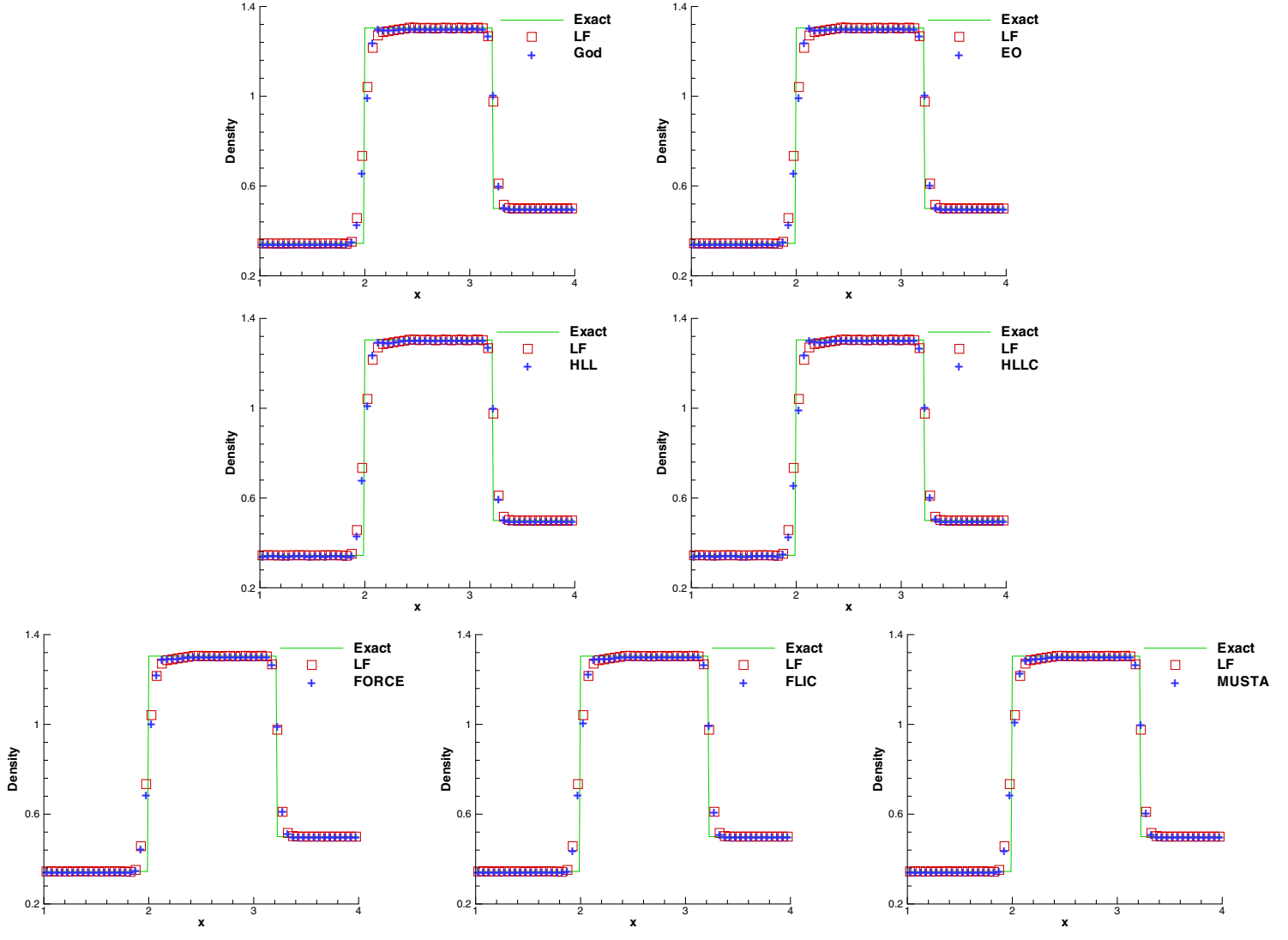


Fig. 2. Lax problem. $t = 1.3$. RKDG with different fluxes, $k = 2$, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

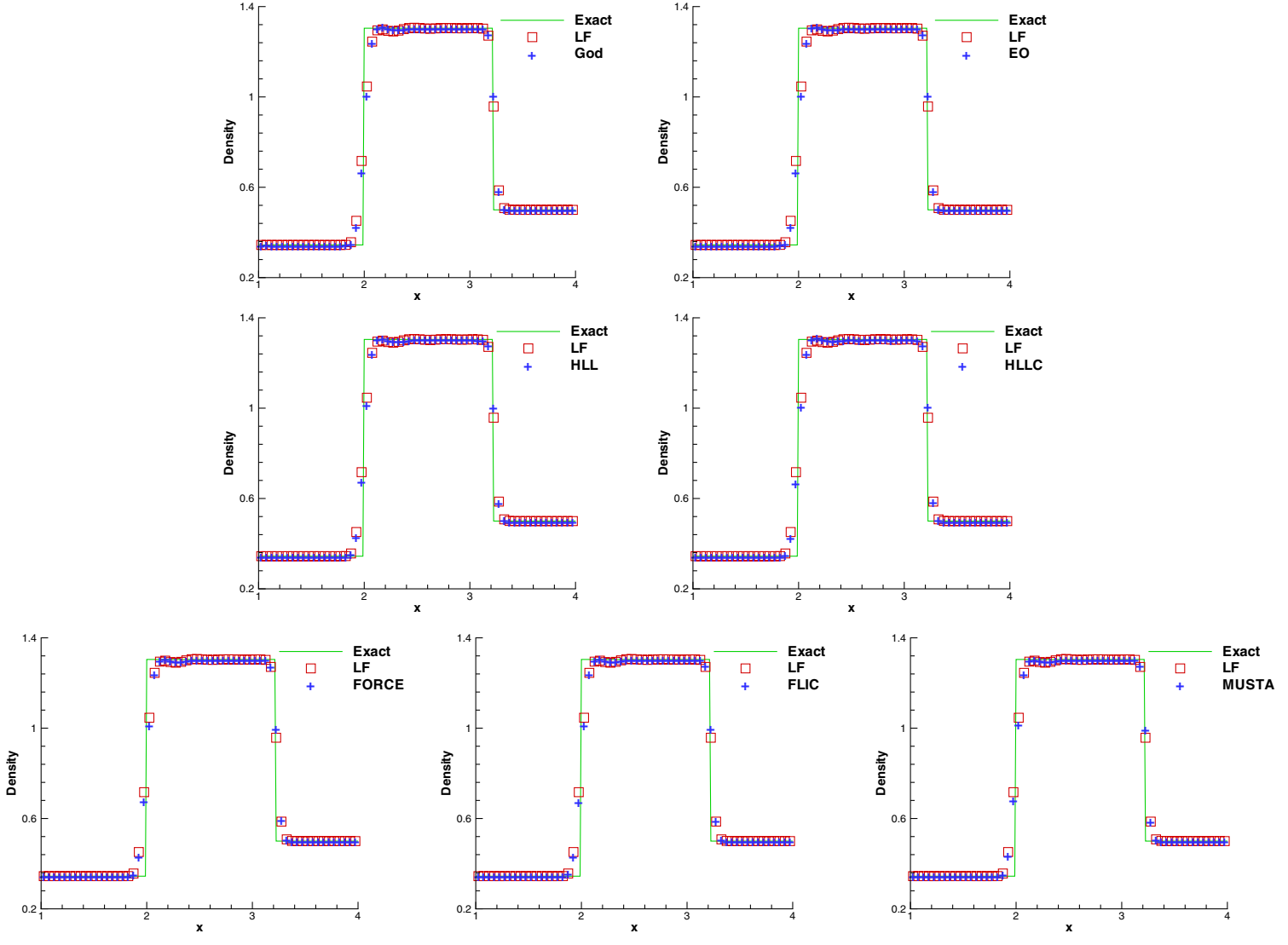


Fig. 3. Lax problem. $t = 1.3$. RKDG with different fluxes, $k = 3$, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

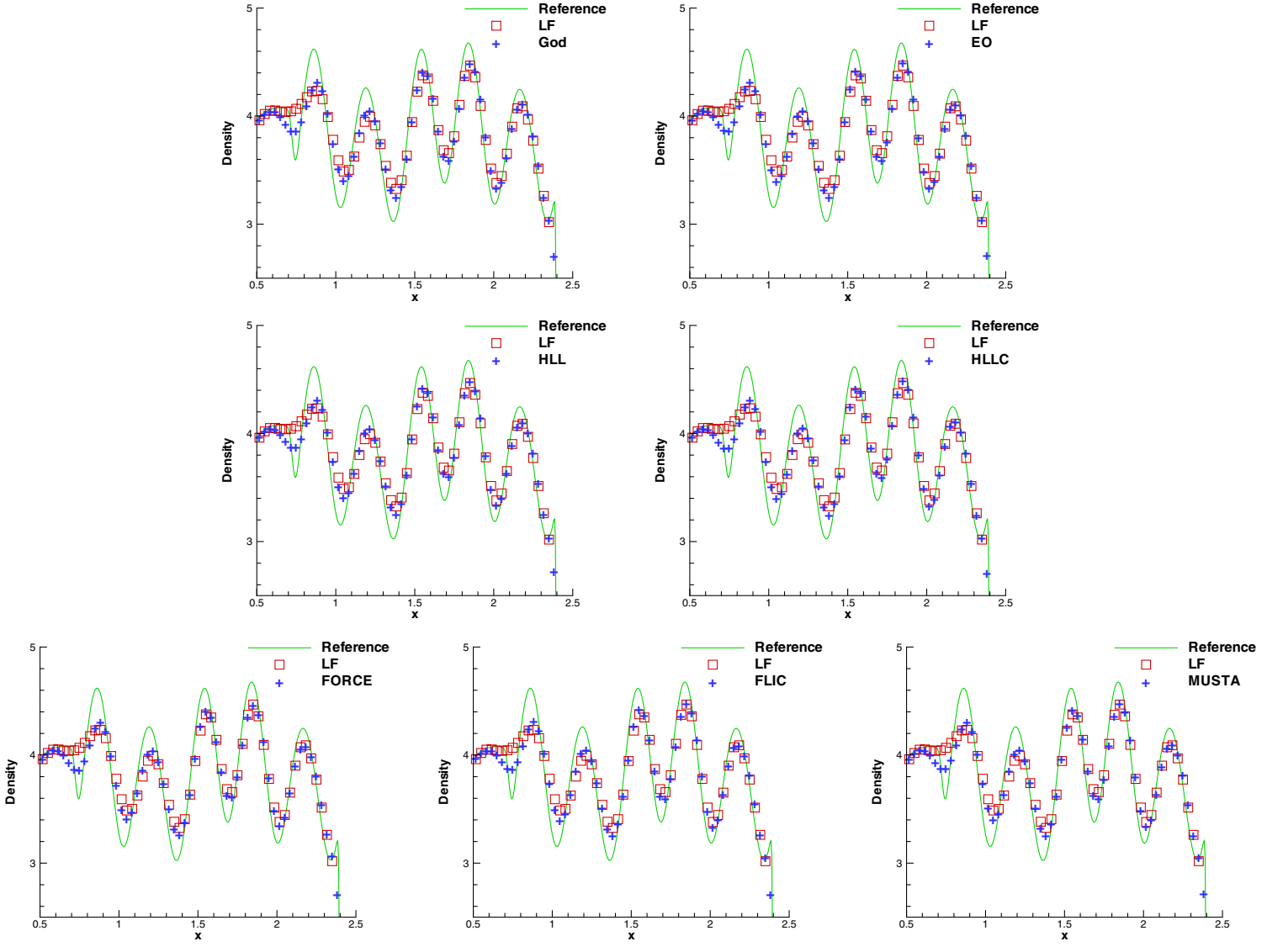


Fig. 4. The shock interaction with entropy waves problem. $t = 1.8$. RKDG with different fluxes, $k = 1$, 300 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

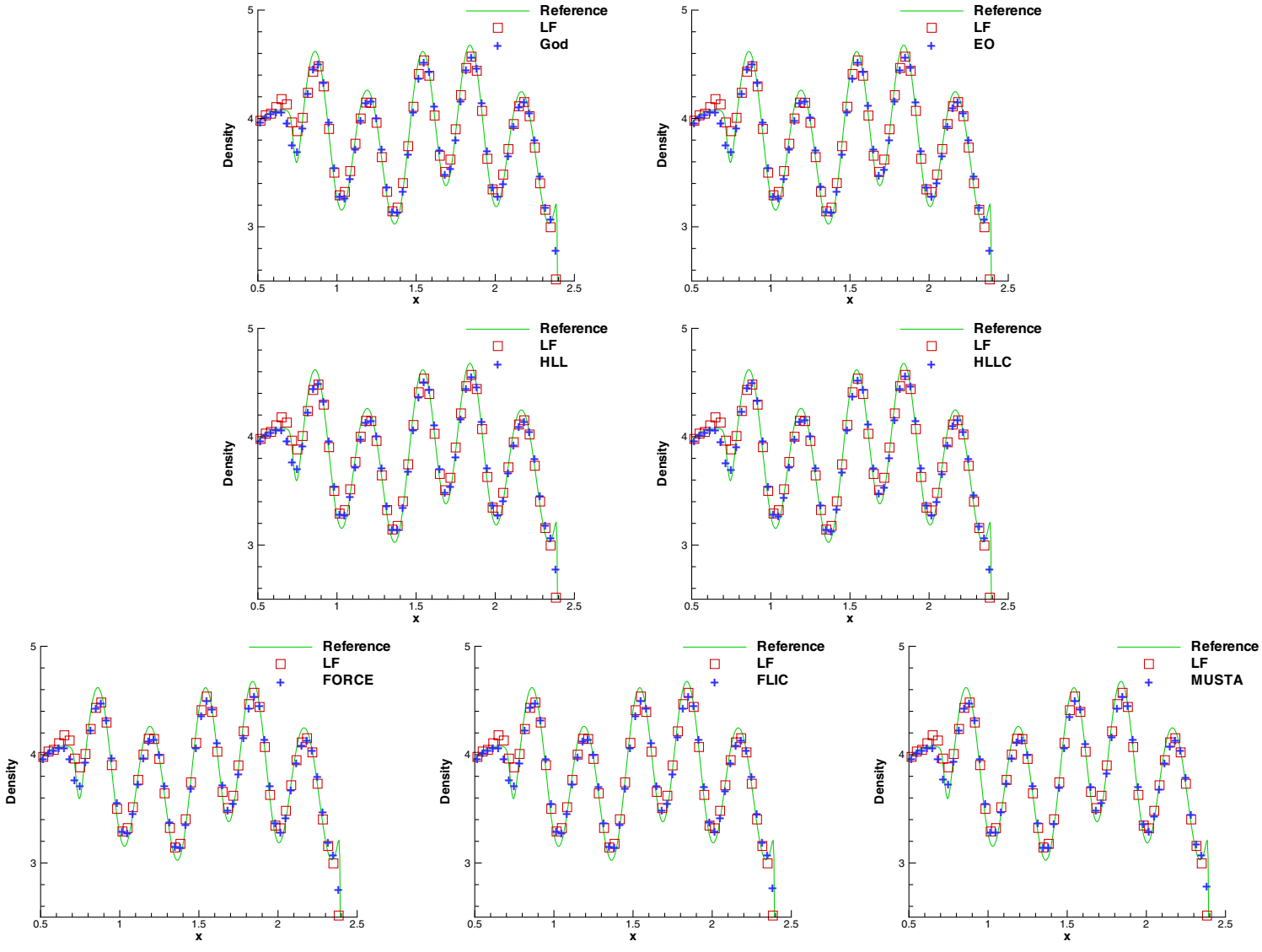


Fig. 5. The shock interaction with entropy waves problem. $t = 1.8$. RKDG with different fluxes, $k = 2$, 300 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

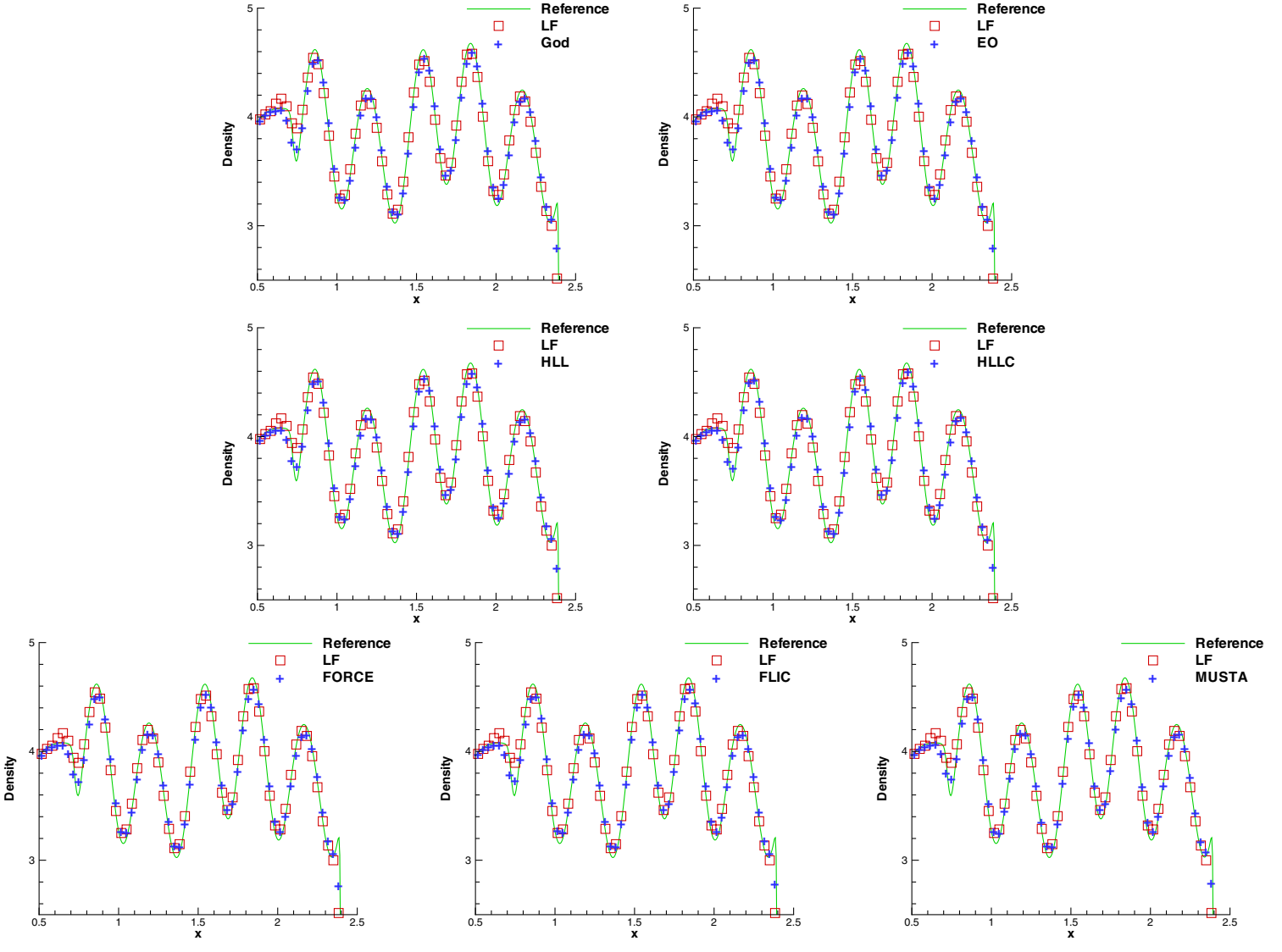


Fig. 6. The shock interaction with entropy waves problem. $t = 1.8$. RKDG with different fluxes, $k = 3$, 300 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

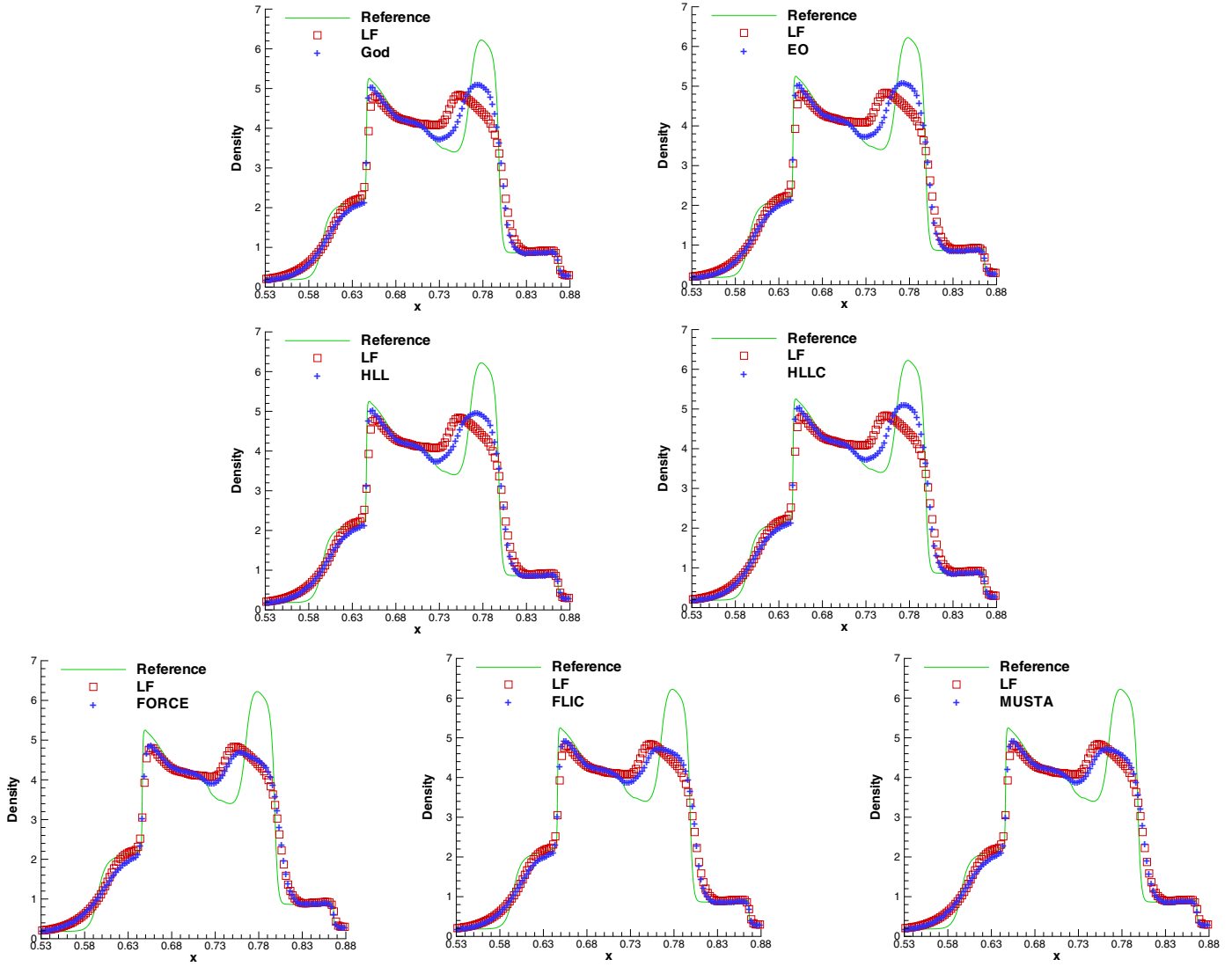


Fig. 7. Blast wave problem. $t = 0.038$. RKDG with different fluxes, $k = 1$, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

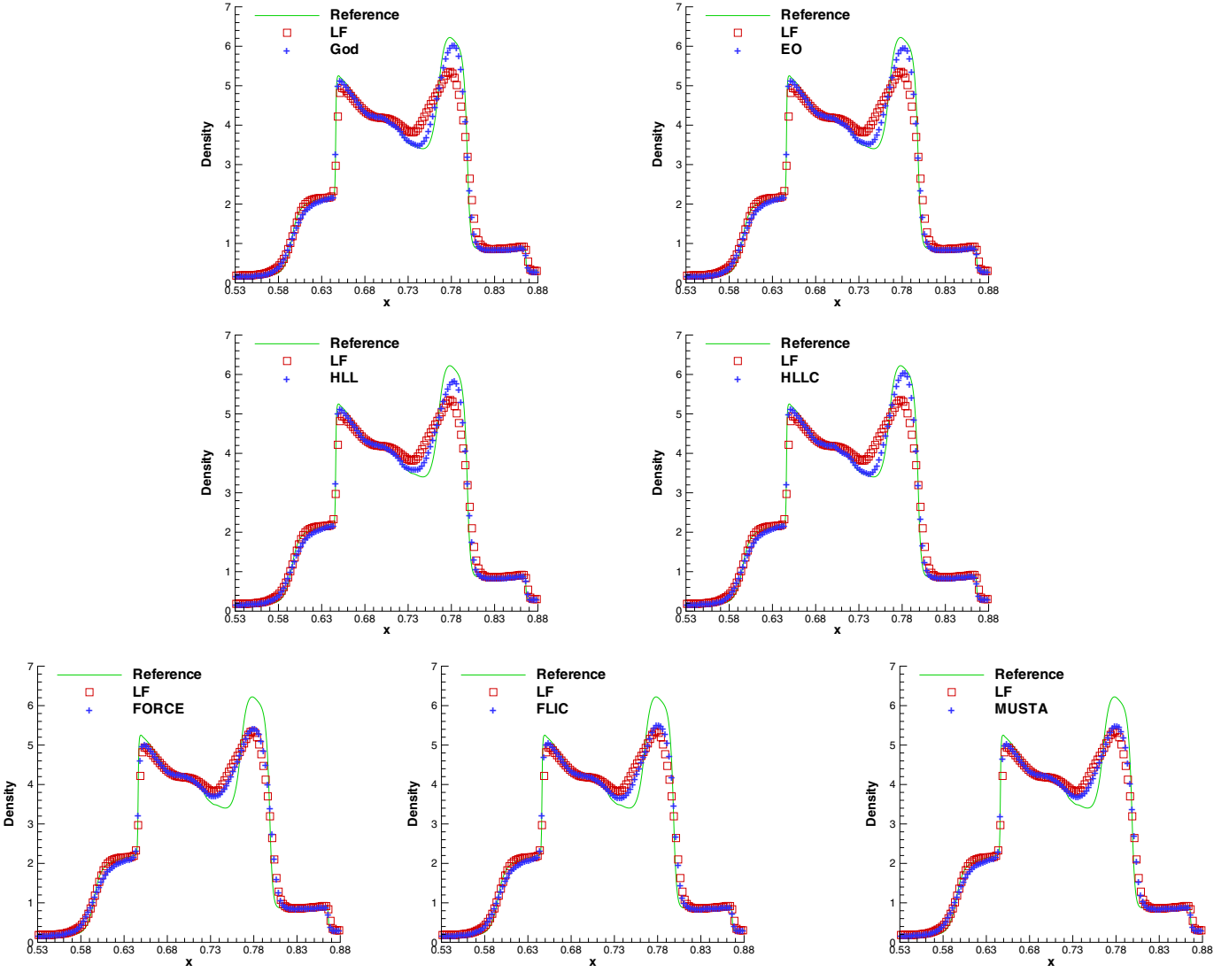


Fig. 8. Blast wave problem. $t = 0.038$. RKDG with different fluxes, $k = 2$, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

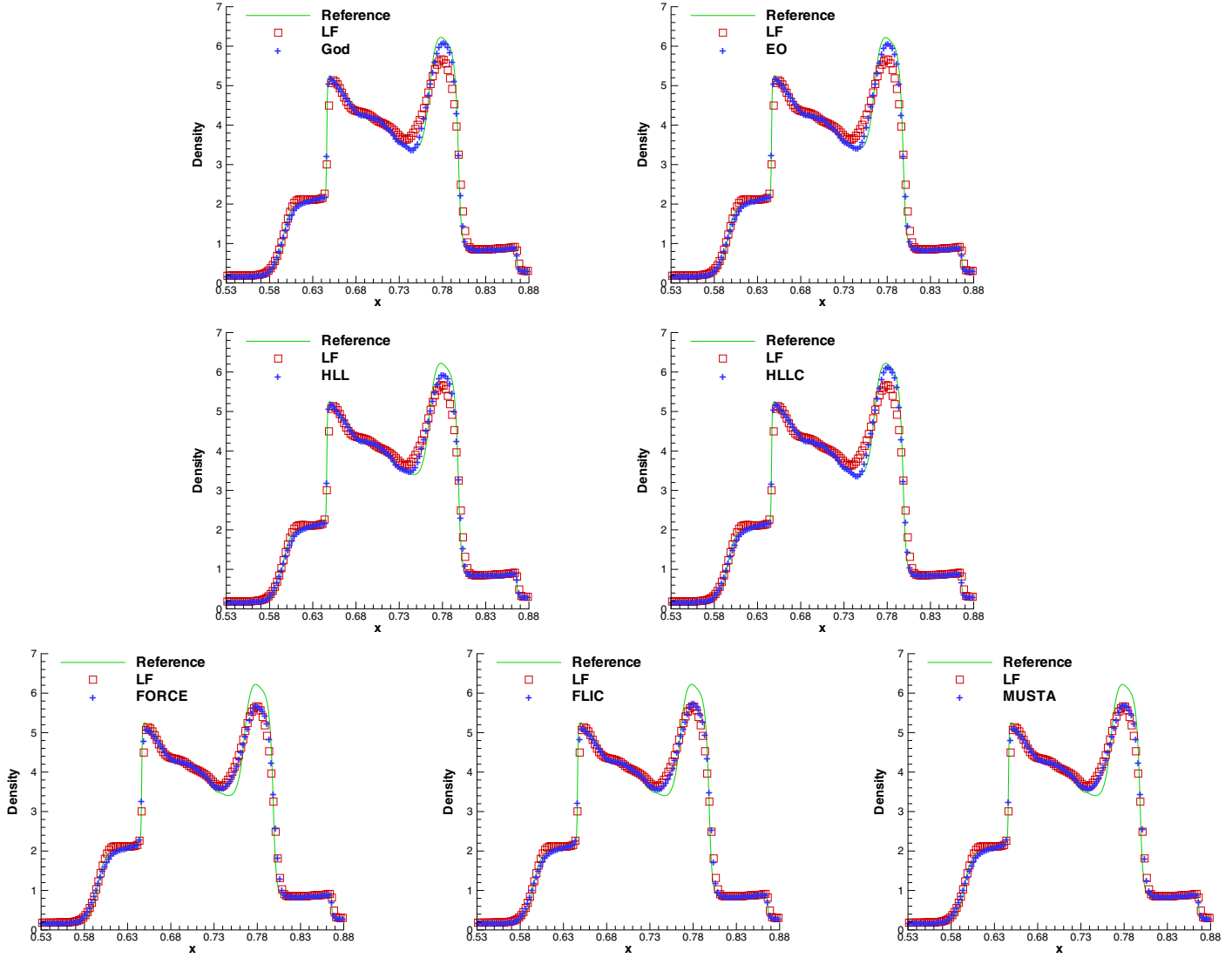


Fig. 9. Blast wave problem. $t = 0.038$. RKDG with different fluxes, $k = 3$, 400 cells. Density. Solid lines: the "exact" reference solution; hollow squares: the results computed by the RKDG-LF scheme; plus symbols: results computed by the RKDG-G (top left), RKDG-EO (top right), RKDG-HLL (middle left), RKDG-HLLC (middle right), RKDG-FORCE (bottom left), RKDG-FLIC (bottom middle) and RKDG-MUSTA (bottom right) schemes.

Table 5
CPU time (in seconds) for the RKDG methods to compute the double Mach reflection problem for the two meshes of 120×30 and 240×60 cells

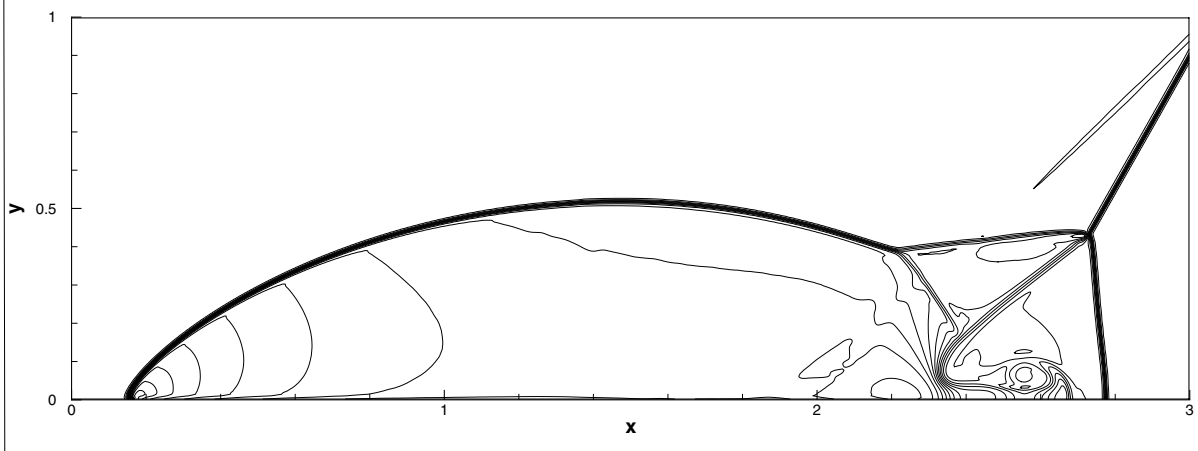
Flux	$N_x \times N_y$: 120×30			$N_x \times N_y$: 240×60		
	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$
LF	114.27	989.05	2778.06	1084.62	8177.00	21456.46
HLL	147.75	1149.49	3236.46	1388.65	9370.68	26396.87
FORCE	126.51	1120.47	2686.68	1211.80	9172.67	20737.88
MUSTA	132.77	1095.01	2926.06	1243.07	8788.72	22505.88

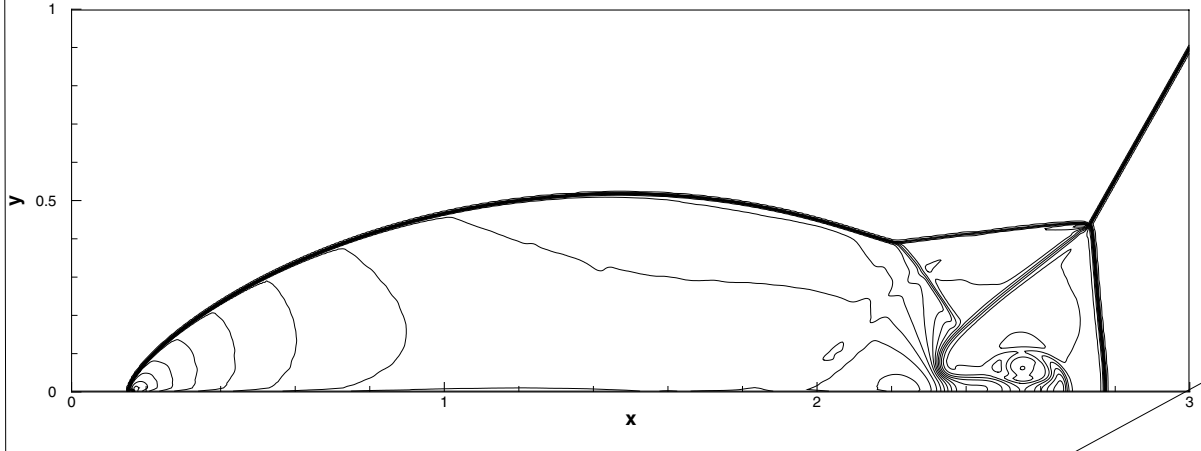
Similar to the previous two examples, the resolution of the RKDG-LF scheme is the worst among all schemes. The resolution of the RKDG-G, RKDG-EO and RKDG-HLLC schemes is the best, followed closely by that of the RKDG-HLL scheme, and the resolution of these four schemes is much better than that of the other four schemes. The resolution of the RKDG-FORCE, RKDG-FLIC and RKDG-MUSTA schemes are only slightly better than that of the RKDG-LF scheme.

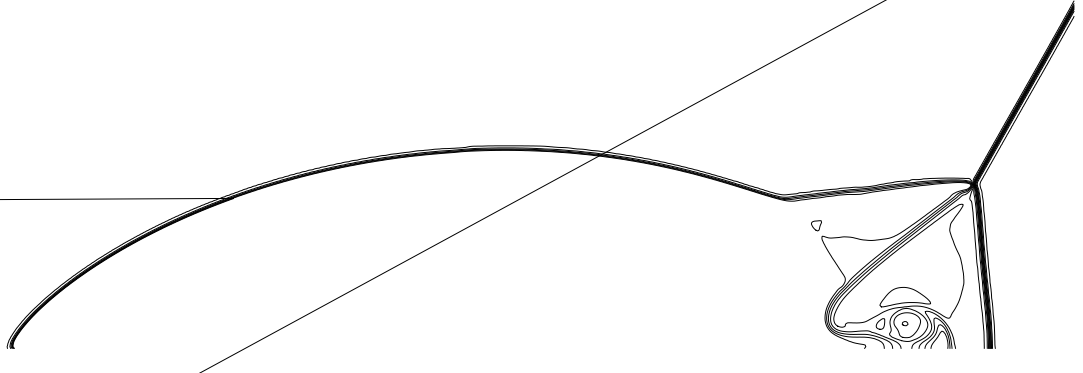
Example 3.5. Double Mach reflection. This problem is originally from [27]. The computational domain for this problem is $[0,4] \times [0,1]$. The reflecting wall lies at the bottom, starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$ and makes 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 $t = 0.2$. Based on our numerical experimental results for the one dimensional case, we test only the four relatively better performing schemes, namely the RKDG-LF, RKDG-HLL, RKDG-HLLC and RKDG-MUSTA schemes. The results of the RKDG-HLLC scheme are almost the same as that of the RKDG-HLL scheme, hence we only report the results of the other three schemes. In Table 5, we document the CPU time by the RKDG-LF, RKDG-HLL and RKDG-MUSTA schemes. We can see that the RKDG-HLL scheme costs about 15–30% more CPU time than the RKDG-LF scheme for the same order of accuracy and same mesh, and the RKDG-MUSTA scheme costs about 5–15% more than the RKDG-LF scheme. The RKDG methods with WENO limiters, for four uniform meshes, with 120×30, 240×60, 480×120 and 960×240 cells, and three different orders of accuracy (from $k = 1$ to $k = 3$, second to fourth order), are used in the numerical experiments. To save space, we plot only the simulation results on the most refined mesh with 960×240 cells by the RKDG-LF, RKDG-HLL and RKDG-MUSTA schemes in Figs. 10–12. All the figures are showing 30 equally spaced density contours from 1.5 to 22.7. It seems that all schemes perform similarly well for this test case.

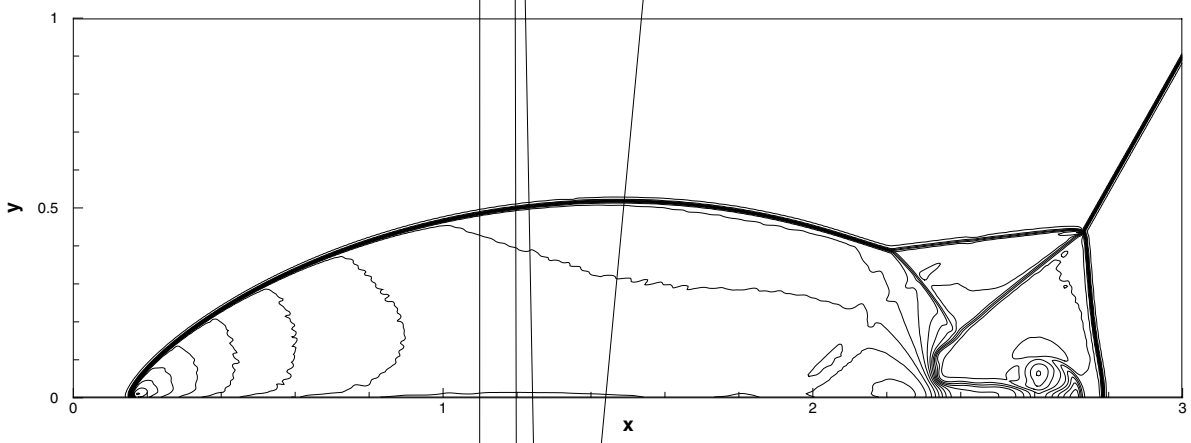
4. Concluding remarks

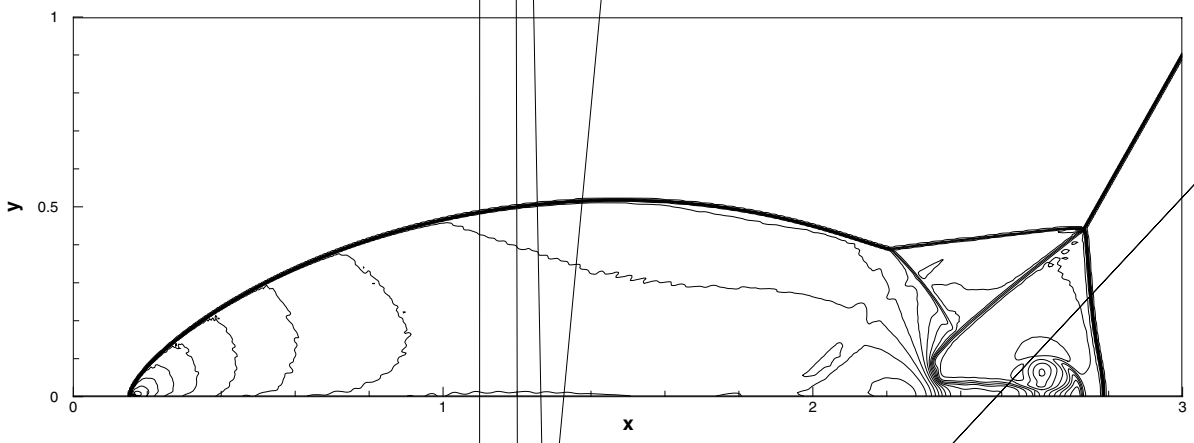
In this paper, we have systematically studied and compared a few different fluxes for the RKDG methods. Extensive one and two dimensional simulations on the hyperbolic systems of Euler equations indicate that RKDG methods with the LF flux cost the least CPU time among all, but the numerical errors and resolution of solutions on the discontinuities are also the worst among all. The RKDG methods with the Godunov or EO fluxes seem to cost significantly more CPU time than the RKDG-LF method. The HLL, HLLC and MUSTA fluxes might be good choices as fluxes for the RKDG method when all factors such as the cost of CPU time, numerical errors and resolution of discontinuities in the solution are considered.

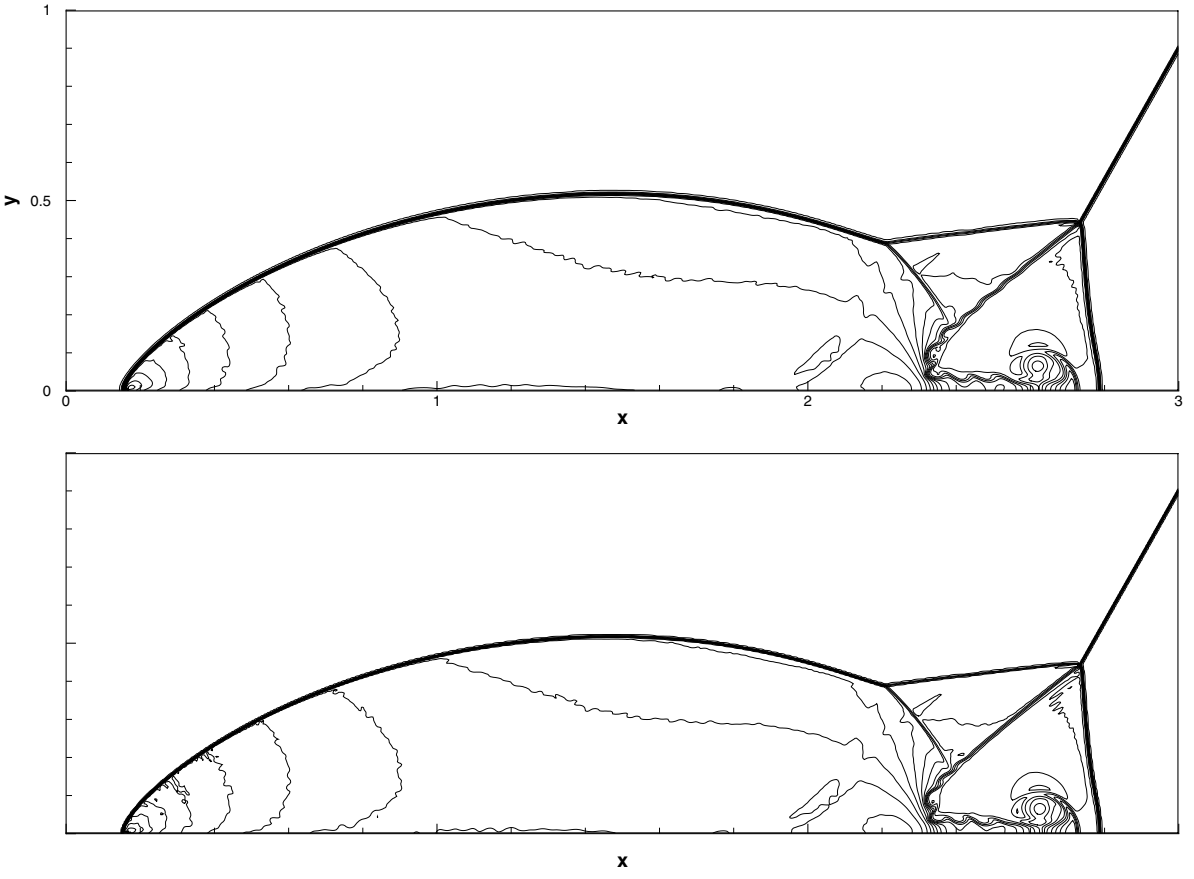

1
0.5
0
y
0
1
2
3
x


1
0.5
0
y
0
1
2
3
x




1
0.5
0
y
0
1
2
3
x


1
0.5
0
y
0
1
2
3
x


1
0.5
0
y
0
1
2
3
x
x

We have also tested the RKDG methods based on a few other numerical fluxes, such as the second-order Lax–Wendroff (LW) flux, the Warming-Beam (WB) flux, and the WAF flux [24]. Our numerical tests indicate that spurious oscillations appear for the Lax shock tube problem for the RKDG-LW and RKDG-WB schemes, and the codes are unstable (they blow up) for the blast wave test case. Because the WAF flux is based on the average of Godunov and Lax–Wendroff fluxes, it is more costly than the Godunov flux and hence is not comparable in CPU time cost with schemes such as RKDG-HLL and RKDG-MUSTA.

References

[1] B. Cockburn, S. Hou, 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.
[2] B. Cockburn, S.-Y. Lin, 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.
[3] B. Cockburn, 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.
[4] B. Cockburn, 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.
[5] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[6] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin method for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.
[7] B. Cockburn, C.-W. Shu, Foreword (for the special issue on discontinuous Galerkin methods), J. Sci. Comput. 22–23 (2005) 1–2.
[8] B. Engquist, S. Osher, One sided difference approximation for nonlinear conservation laws, Math. Comp. 36 (1981) 321–351.
[9] S.K. Godunov, Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics, Math. Sbornik 47 (1959) 271–306.
[10] A. Harten, B. Engquist, S. Osher, S. Chakravathy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 71 (1987) 231–303.
[11] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35–61.
[12] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[13] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Appl. Numer. Math. 48 (2004) 323–338.
[14] S. Osher, F. Solomon, Upwind difference schemes for hyperbolic conservation laws, Math. Comp. 38 (1982) 339–374.
[15] J. Qiu, C.-W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput. 26 (2005) 907–929.
[16] J. Qiu, C.-W. Shu, A comparison of troubled-cell indicators for Runge–Kutta discontinuous Galerkin methods using WENO limiters, SIAM J. Sci. Comput., in press.
[17] W.H. Reed, T.R. Hill, Triangular mesh methods for neutron transport equation, Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[18] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comp. 49 (1987) 105–121.
[19] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor (Eds.), in: A. Quarteroni (Ed.), Lecture Notes in Mathematics, vol. 1697, Springer, 1998, pp. 325–432.
[20] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[21] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.
[22] V.A. Titarev, E.F. Toro, Finite-volume WENO schemes for three-dimensional conservation laws, J. Comput. Phys. 201 (2004) 238–260.
[23] V.A. Titarev, E.F. Toro, WENO schemes based on upwind and centred TVD fluxes, Comput. Fluids 34 (2005) 705–720.
[24] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, a Practical Introduction, Springer, Berlin, 1997.
[25] E.F. Toro, Multi-stage predictor-corrector fluxes for hyperbolic equations, Preprint NI03037-NPA, Isaac Newton Institute for Mathematical Sciences, University of Cambridge, UK.

[26] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the Harten–Lax–van Leer Riemann solver, J. Shock Waves 4 (1994) 25–34.
[27] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.