Bull. Korean Math. Soc. **48** (2011), No. 4, pp. 873–896
DOI 10.4134/BKMS.2011.48.4.873

CENTRAL SCHEMES WITH LAX-WENDROFF TYPE TIME DISCRETIZATIONS

Suyeon Shin and Woonjae Hwang

Abstract. The semi-discrete central scheme and central upwind scheme use Runge-Kutta (RK) time discretization. We do the Lax-Wendroff (LW) type time discretization for both schemes. We perform numerical experiments for various problems including two dimensional Riemann problems for Burgers' equation and Euler equations. The results show that the LW time discretization is more efficient in CPU time than the RK time discretization while maintaining the same order of accuracy.

1. Introduction

The central scheme was introduced by Nessyahu and Tadmor in 1990 [9]. Since then, many works are done by using central scheme [3, 5, 6, 7]. The central scheme is very convenient because it doesn't require Riemann solvers. As an advanced version, the central upwind scheme was introduced by Kuganov et al. in 2001 [4].

Since Liu et al. introduced the Weighted Essentially Non-Oscillatory(WENO) scheme, WENO became very popular for high-order computations [8]. Original WENO uses Runge-Kutta (RK) time discretization. Some works for WENO with Lax-Wendroff (LW) type time discretization were done instead of RK time discretization [10, 11, 12].

The semi-discrete central scheme also uses the RK time discretization. In this paper, we do the LW type time discretization for central scheme and central upwind scheme. In Section 2, we show the Lax-Wendroff (LW) type time discretization for 1-D and 2-D scalar equations and systems. In Section 3, we briefly explain central scheme and central upwind scheme. In Section 4, we do numerical experiments for the various problems. We apply central scheme and central upwind scheme with LW and RK time discretizations for 1-D and 2-D linear advection equation, Burgers equation, Buckley-Leverett equation and Euler equations. We compare the errors and check the order of accuracy for

Received January 22, 2010.
2010 *Mathematics Subject Classification.* Primary 65M06.
Key words and phrases. central scheme, Lax-Wendroff type time discretization.
This work is financially supported by Korea University grant.

both RK and LW time discretizations with smooth initial conditions. We also consider two dimensional Riemann problems for scalar equations and systems. For two dimensional Riemann problems, see [1, 2, 7, 13] for examples. To check the efficiency, we compare the CPU time of two-dimensional Riemann problems for both time discretizations. The conclusion follows in Section 5.

2. Lax-Wendroff type time discretization

2.1. One-dimensional scalar equations

Consider the initial value problem for the one-dimensional scalar conservation laws:

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

We explain the procedures for the LW type time discretization [12]. By a temporal Taylor expansion we obtain

$$u(x,t+\triangle t) = u(x,t) + \triangle t u' + \frac{\triangle t^2}{2}u'' + \frac{\triangle t^3}{6}u''' + \frac{\triangle t^4}{24}u^{(4)} + \cdots . \tag{1}$$

In this paper, we use 3rd order accuracy in time. So we need to approximate the first 3 time derivatives u', u'' and u'''. This procedure can be extended to any higher orders as desired.

Step 1. The reconstruction of the first derivative $u' = -f(u)_x$ is obtained by the second order central scheme.

Step 2. The reconstruction of the second time derivative $u'' = -(f'(u)u')_x$ is obtained as follows. Let $g_i = f'(u_i)u_i'$, where u_i and u_i' are the point values of u and u' at the point (x_i, t^n) computed in Step 1 described above. We can use a simple central difference formula to approximate u'' at the point (x_i, t^n)(i.e. $u'' \approx -\frac{1}{2\Delta x}(g_{i+1} - g_{i-1})$).

Step 3. The reconstruction of the third time derivative $u''' = -(f'(u)u'' + f''(u)(u')^2)_x$ is obtained as follows. Let $g_i = f'(u_i)u_i'' + f''(u_i)(u_i')^2$; here u_i' and u_i'' are the point values of u' and u'' at the point (x_i, t^n) computed in Step 1 and Step 2 above. Then we can get the approximation of u''' by the same way as in Step 2.

2.2. One-dimensional systems

Consider the initial value problem for the one-dimensional system of conservation laws:

$$\begin{cases} \frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}f(u(x,t)) = 0, \\ u(x,0) = u_0(x), \end{cases}$$

where $u : \mathbb{R} \times [0, \infty) \to \mathbb{R}^m$ is an m-dimensional vector of conserved quantities, and $f : \mathbb{R}^m \to \mathbb{R}^m$ is the vector valued flux function. Then, we have

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_t + \begin{pmatrix} f_1(u) \\ f_2(u) \\ \vdots \\ f_m(u) \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

i.e.,

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_t + \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

We apply the temporal Taylor expansion (1) with an m-dimensional vector u.

Step 1. The reconstruction of the first derivative $u' = -f(u)_x$ is obtained by the second order central scheme.

Step 2. The reconstruction of the second time derivative $u'' = -(f'(u)u')_x$ is obtained as follows.
Let

$$\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_m \end{pmatrix},$$

where u_i and u'_i are the point values of u and u' at the point (x_i, t^n) computed in Step 1 described above. We can use a simple central difference formula to approximate u'' at the point (x_i, t^n).

Step 3. The reconstruction of the third time derivative $u''' = -(f'(u)u'' + f''(u)(u')^2)_x$ is obtained as follows. Let

$$\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u''_1 \\ u''_2 \\ \vdots \\ u''_m \end{pmatrix}$$
$$+ \begin{pmatrix} \frac{\partial^2 f_1}{\partial u_1^2} & \frac{\partial^2 f_1}{\partial u_2^2} & \cdots & \frac{\partial^2 f_1}{\partial u_m^2} \\ \frac{\partial^2 f_2}{\partial u_1^2} & \frac{\partial^2 f_2}{\partial u_2^2} & \cdots & \frac{\partial^2 f_2}{\partial u_m^2} \\ & & \vdots & \\ \frac{\partial^2 f_m}{\partial u_1^2} & \frac{\partial^2 f_m}{\partial u_2^2} & \cdots & \frac{\partial^2 f_m}{\partial u_m^2} \end{pmatrix} \begin{pmatrix} (u'_1)^2 \\ (u'_2)^2 \\ \vdots \\ (u'_m)^2 \end{pmatrix},$$

here u'_i and u''_i are the point values of u' and u'' at the point (x_i, t^n) computed in Step 1 and Step 2 above. Then we can get the approximation of u''' by the same way as in Step 2.

2.3. Two-dimensional scalar equations

Consider the initial value problem for two-dimensional conservation laws:

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

By a temporal Taylor expansion we obtain

$$u(x,y,t+\triangle t) = u(x,y,t) + \triangle t u' + \frac{\triangle t^2}{2}u'' + \frac{\triangle t^3}{6}u''' + \frac{\triangle t^4}{24}u^{(4)} + \cdots . \tag{2}$$

The first time derivative $u' = -f(u)_x - g(u)_y$ is approximated by the second order central scheme. The second order time derivative is $u'' = -(f'(u)u')_x - (g'(u)u')_y$, and the third order time derivative is $u''' = -(f''(u)(u')^2 + f'(u)u'')_x - (g''(u)(u')^2 + g'(u)u'')_y$.

Step 1. The reconstruction of the first derivative $u' = -f(u)_x - g(u)_y$ is obtained by the second order central scheme.

Step 2. The reconstruction of the second time derivative $u'' = -(f'(u)u')_x - (g'(u)u')_y$ is obtained as follows. Let $l_i = f'(u_i)u'_i$ and $n_i = g'(u_i)u'_i$, where u_i and u'_i are the point values of u and u' at the point (x_i, y_i, t^n) computed in Step 1 described above. We can use a simple central difference formula to approximate u'' at the point (x_i, y_i, t^n) (i.e., $u'' \approx -\frac{1}{2\Delta x}(l_{i+1} - l_{i-1}) - \frac{1}{2\Delta y}(n_{i+1} - n_{i-1})$).

Step 3. The reconstruction of the third time derivative $u''' = -(f''(u)(u')^2 + f'(u)u'')_x - (g''(u)(u')^2 + g'(u)u'')_y$ is obtained as follows. Let $l_i = f'(u_i)u''_i + f''(u_i)(u'_i)^2$ and $n_i = g'(u_i)u''_i + g''(u_i)(u'_i)^2$; here u'_i and u''_i are the point values of u' and u'' at the point (x_i, y_i, t^n) computed in Step 1 and Step 2 above. Then we can get the approximation of u''' by the same way as in Step 2.

2.4. Two-dimensional systems

Consider the initial value problem for the two-dimensional system of conservation laws:

$$\begin{cases} \frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}f(u(x,y,t)) + \frac{\partial}{\partial y}g(u(x,y,t)) = 0, \\ u(x,y,0) = u_0(x,y), \end{cases}$$

where $u : \mathbb{R}^2 \times [0,\infty) \to \mathbb{R}^m$ is an m-dimensional vector of conserved quantities, and $f, g : \mathbb{R}^m \to \mathbb{R}^m$ are the vector valued flux function. Then, we have

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_t + \begin{pmatrix} f_1(u) \\ f_2(u) \\ \vdots \\ f_m(u) \end{pmatrix}_x + \begin{pmatrix} g_1(u) \\ g_2(u) \\ \vdots \\ g_m(u) \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

i.e.,

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_t + \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_x + \begin{pmatrix} \frac{\partial g_1}{\partial g_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial g_m}{\partial u_1} & \frac{\partial g_m}{\partial u_2} & \cdots & \frac{\partial g_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

We apply the temporal Taylor expansion (2) with an m-dimensional vector u.

Step 1. The reconstruction of the first derivative $u' = -f(u)_x - g(u)_y$ is obtained by the second order central scheme.

Step 2. The reconstruction of the second time derivative $u'' = -(f'(u)u')_x - (g'(u)u')_y$ is obtained as follows.

Let

$$\begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_m \end{pmatrix}$$

and

$$\begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial g_m}{\partial u_1} & \frac{\partial g_m}{\partial u_2} & \cdots & \frac{\partial g_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u'_1 \\ u'_2 \\ \vdots \\ u'_m \end{pmatrix},$$

where u_i and u'_i are the point values of u and u' at the point (x_i, y_i, t^n) computed in Step 1 described above. We can use a simple central difference formula to approximate u'' at the point (x_i, y_i, t^n).

Step 3. The reconstruction of the third time derivative $u''' = -(f''(u)(u')^2 + f'(u)u'')_x - (g''(u)(u')^2 + g'(u)u'')_y$ is obtained as follows.

Let

$$\begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & \cdots & \frac{\partial f_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u_1'' \\ u_2'' \\ \vdots \\ u_m'' \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 f_1}{\partial u_1^2} & \frac{\partial^2 f_1}{\partial u_2^2} & \cdots & \frac{\partial^2 f_1}{\partial u_m^2} \\ \frac{\partial^2 f_2}{\partial u_1^2} & \frac{\partial^2 f_2}{\partial u_2^2} & \cdots & \frac{\partial^2 f_2}{\partial u_m^2} \\ & & \vdots & \\ \frac{\partial^2 f_m}{\partial u_1^2} & \frac{\partial^2 f_m}{\partial u_2^2} & \cdots & \frac{\partial^2 f_m}{\partial u_m^2} \end{pmatrix} \begin{pmatrix} (u_1')^2 \\ (u_2')^2 \\ \vdots \\ (u_m')^2 \end{pmatrix}$$

and

$$\begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial g_m}{\partial u_1} & \frac{\partial g_m}{\partial u_2} & \cdots & \frac{\partial g_m}{\partial u_m} \end{pmatrix} \begin{pmatrix} u_1'' \\ u_2'' \\ \vdots \\ u_m'' \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 g_1}{\partial u_1^2} & \frac{\partial^2 g_1}{\partial u_2^2} & \cdots & \frac{\partial^2 g_1}{\partial u_m^2} \\ \frac{\partial^2 g_2}{\partial u_1^2} & \frac{\partial^2 g_2}{\partial u_2^2} & \cdots & \frac{\partial^2 g_2}{\partial u_m^2} \\ & & \vdots & \\ \frac{\partial^2 g_m}{\partial u_1^2} & \frac{\partial^2 g_m}{\partial u_2^2} & \cdots & \frac{\partial^2 g_m}{\partial u_m^2} \end{pmatrix} \begin{pmatrix} (u_1')^2 \\ (u_2')^2 \\ \vdots \\ (u_m')^2 \end{pmatrix},$$

here u_i' and u_i'' are the point values of u' and u'' at the point (x_i, y_i, t^n) computed in Step 1 and Step 2 above. Then we can get the approximation of u''' by the same way as in Step 2.

3. Central scheme and central upwind scheme

We give a brief overview of central schemes [6]. Assume that we have already computed the piecewise-linear solution at time level t^n, based on the cell averages u_j^n, and have reconstructed approximate derivatives $(u_x)_j^n = minmod(\frac{u_j^n - u_{j-1}^n}{\Delta x}, \frac{u_{j+1}^n - u_j^n}{\Delta x})$. We now turn to evolve it in time. To begin with, we estimate the local speed of propagation at the cell boundaries, $x_{j+\frac{1}{2}}$: the upper bound is denoted by $a_{j+\frac{1}{2}}^n$ and is given by

$$a_{j+\frac{1}{2}}^n := \max_{\omega \in C(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+)} \rho(\frac{\partial f}{\partial u}(\omega)), \tag{3}$$

where

$$
\begin{aligned}
u^+_{j+\frac{1}{2}} &:= u^n_{j+1} - \frac{\Delta x}{2}(u_x)^n_{j+1} \equiv p_{j+1}(x_{j+\frac{1}{2}}),\\
u^-_{j+\frac{1}{2}} &:= u^n_j + \frac{\Delta x}{2}(u_x)^n_j \equiv p_j(x_{j+\frac{1}{2}}),
\end{aligned}
\tag{4}
$$

are the correspondent left and right intermediate values of u at $x_{j+\frac{1}{2}}$, and $C(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}})$ is a curve in phase space connecting $u^-_{j+\frac{1}{2}}$ and $u^+_{j+\frac{1}{2}}$ via the Riemann fan.

We consider the domains

$$
[x^n_{j-\frac{1}{2},r}, x^n_{j+\frac{1}{2},l}] \times [t^n, t^{n+1}] \quad \text{and} \quad [x^n_{j+\frac{1}{2},l}, x^n_{j+\frac{1}{2},r}] \times [t^n, t^{n+1}],
\tag{5}
$$

with $x^n_{j+\frac{1}{2},l} := x_{j+\frac{1}{2}} - \triangle t a^n_{j+\frac{1}{2}}$ and $x^n_{j+\frac{1}{2},r} := x_{j+\frac{1}{2}} + \triangle t a^n_{j+\frac{1}{2}}$.
Given the reconstruction $\{p^n_j(x)\}$, we integrate over these domains and obtain the cell averages

$$
\begin{aligned}
\overline{w}^{n+1}_j = \frac{1}{x^n_{j+\frac{1}{2},l} - x^n_{j-\frac{1}{2},r}}[&\int_{x^n_{j-\frac{1}{2},r}}^{x^n_{j+\frac{1}{2},l}} p^n_j(x)dx\\
&- \int_{t^n}^{t^{n+1}} (f(u(x^n_{j+\frac{1}{2},l}, t)) - f(u(x^n_{j-\frac{1}{2},r}, t)))dt]
\end{aligned}
\tag{6}
$$

and

$$
\begin{aligned}
\overline{w}^{n+1}_{j+\frac{1}{2}} = \frac{1}{x^n_{j+\frac{1}{2},r} - x^n_{j+\frac{1}{2},l}}[&\int_{x^n_{j+\frac{1}{2},l}}^{x_{j+\frac{1}{2}}} p^n_j(x)dx + \int_{x_{j+\frac{1}{2}}}^{x^n_{j+\frac{1}{2},r}} p^n_{j+1}(x)dx\\
&- \int_{t^n}^{t^{n+1}} (f(u(x^n_{j+\frac{1}{2},r}, t)) - f(u(x^n_{j+\frac{1}{2},l}, t)))dt]
\end{aligned}
\tag{7}
$$

over the corresponding non-equal spatial cells.

To obtain the cell averages over the original grid of uniform, $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, we consider the piecewise-linear construction over the nonuniform cells at $t = t^n$,

$$
\begin{aligned}
\tilde{w}^{n+1}(x) := \sum_j \{&[\overline{w}^{n+1}_{j+\frac{1}{2}} + (u_x)^{n+1}_{j+\frac{1}{2}}(x - x_{j+\frac{1}{2}})]1_{[x^n_{j+\frac{1}{2},l}, x^n_{j+\frac{1}{2},r}]}\\
&+ \overline{w}^{n+1}_j 1_{[x^n_{j-\frac{1}{2},r}, x^n_{j+\frac{1}{2},l}]}\}.
\end{aligned}
\tag{8}
$$

The construction of this scheme is completed by projection $\tilde{w}^{n+1}$ back onto the original grid, i.e., we compute the cell averages

$$
\overline{u}^{n+1}_j = \frac{1}{\triangle x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tilde{w}^{n+1}(x)dx
\tag{9}
$$

at next time level. This leads to a fully discrete central scheme. We omit the details of messy computations and continue semidiscrete framework.

The time derivative of $\overline{u}_j(t)$ is expressed with the help of (9) as

$$\frac{d}{dt}\overline{u}_j(t) = \lim_{\Delta t \to 0} \frac{\overline{u}_j^{n+1} - \overline{u}_j^n}{\Delta t} = \lim_{\Delta t \to 0} \frac{1}{\triangle t}\Big[\frac{1}{\triangle x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tilde{w}^{n+1}(x)dx - \overline{u}_j(t)\Big]. \tag{10}$$

We derive the semi-discrete scheme

$$\frac{d}{dt}\overline{u}_j(t) = -\frac{H_{j+\frac{1}{2}}(t) - H_{j-\frac{1}{2}}(t)}{\Delta x}, \tag{11}$$

with the numerical flux

$$H_{j+\frac{1}{2}}(t) := \frac{f(u^+_{j+\frac{1}{2}}(t)) + f(u^-_{j+\frac{1}{2}}(t))}{2} - \frac{a_{j+\frac{1}{2}}(t)}{2}[u^+_{j+\frac{1}{2}}(t) - u^-_{j+\frac{1}{2}}(t)]. \tag{12}$$

The intermediate values $u^{\pm}_{j+\frac{1}{2}}$ are given by (4).

In two-dimensional case, the corresponding semi-discrete scheme for the system

$$u_t + f(u)_x + g(u)_y = 0 \tag{13}$$

is

$$\frac{d}{dt}\overline{u}_{j,k}(t) := -\frac{H^x_{j+\frac{1}{2},k}(t) - H^x_{j-\frac{1}{2},k}(t)}{\Delta x} - \frac{H^y_{j,k+\frac{1}{2}}(t) - H^y_{j,k-\frac{1}{2}}(t)}{\Delta y}. \tag{14}$$

Here, the numerical fluxes are a straightforward generalization of the one dimensional numerical flux,

$$H^x_{j+\frac{1}{2},k}(t) := \frac{f(u^+_{j+\frac{1}{2},k}(t)) + f(u^-_{j+\frac{1}{2},k}(t))}{2} - \frac{a^x_{j+\frac{1}{2},k}(t)}{2}[u^+_{j+\frac{1}{2},k}(t) - u^-_{j+\frac{1}{2},k}(t)], \tag{15}$$

and

$$H^y_{j,k+\frac{1}{2}}(t) := \frac{g(u^+_{j,k+\frac{1}{2}}(t)) + g(u^-_{j,k+\frac{1}{2}}(t))}{2} - \frac{a^y_{j,k+\frac{1}{2}}(t)}{2}[u^+_{j,k+\frac{1}{2}}(t) - u^-_{j,k+\frac{1}{2}}(t)], \tag{16}$$

which are expressed in terms of the intermediate values

$$\begin{aligned} u^{\pm}_{j+\frac{1}{2},k}(t) &:= u_{j+1,k}(t) \mp \frac{\Delta x}{2}(u_x)_{j+\frac{1}{2}\pm\frac{1}{2},k}(t), \\ u^{\pm}_{j,k+\frac{1}{2}}(t) &:= u_{j,k+1}(t) \mp \frac{\Delta y}{2}(u_y)_{j,k+\frac{1}{2}\pm\frac{1}{2}}(t), \end{aligned} \tag{17}$$

and the local speeds, $a^x_{j+\frac{1}{2},k}(t)$ and $a^y_{j,k+\frac{1}{2}}(t)$, are computed, e.g., by

$$a^x_{j+\frac{1}{2},k}(t) := \max_{\pm} \rho(\frac{\partial f}{\partial u}(u^{\pm}_{j+\frac{1}{2},k}(t))), \quad a^y_{j,k+\frac{1}{2}}(t) := \max_{\pm} \rho(\frac{\partial g}{\partial u}(u^{\pm}_{j,k+\frac{1}{2}}(t))). \tag{18}$$

We introduce the central upwind scheme [4]. We require a piecewise linear reconstruction of the form

$$p^n_{j,k}(x,y) = \bar{u}^n_{j,k} + (u_x)^n_{j,k}(x - x_j) + (u_y)^n_{j,k}(y - y_k). \tag{19}$$

Here, $(u_x)^n_{j,k}$ and $(u_y)^n_{j,k}$ stand for an approximation to the derivatives $u_x(x_j, y_k, t^n)$ and $u_y(x_j, y_k, t^n)$, respectively. To ensure a nonoscillatory nature of the reconstruction, one needs to use a nonlinear limiter in the computation of these slopes. This can be done in many different ways. In this article, we have used van Leer's one-parameter family of the minmod limiters

$$\begin{aligned}(u_x)_{j,k} &= \mathit{minmod}(\theta\frac{\bar{u}_{j+1,k}-\bar{u}_{j,k}}{\Delta x}, \frac{\bar{u}_{j+1,k}-\bar{u}_{j-1,k}}{2\Delta x}, \theta\frac{\bar{u}_{j,k}-\bar{u}_{j-1,k}}{\Delta x}),\\ (u_y)_{j,k} &= \mathit{minmod}(\theta\frac{\bar{u}_{j,k+1}-\bar{u}_{j,k}}{\Delta x}, \frac{\bar{u}_{j,k+1}-\bar{u}_{j,k-1}}{2\Delta x}, \theta\frac{\bar{u}_{j,k}-\bar{u}_{j,k-1}}{\Delta x}),\end{aligned} \tag{20}$$

where $\theta \in [1,2]$, and the multivariable minmod function is defined by

$$\mathit{minmod}(x_1, x_2, \ldots) := \begin{cases} \min_j\{x_j\}, & \text{if } x_j > 0 \quad \forall j \\ \max_j\{x_j\}, & \text{if } x_j < 0 \quad \forall j \\ 0, & \text{otherwise.} \end{cases}$$

Given the piecewise linear polynomial we can compute the reconstructed values at the interfaces

$$\begin{aligned} u^N_{j,k} &= p^n_{j,k}(x_j, y_{k+\frac{1}{2}}), \quad & u^S_{j,k} &= p^n_{j,k}(x_j, y_{k-\frac{1}{2}}),\\ u^E_{j,k} &= p^n_{j,k}(x_{j+\frac{1}{2}}, y_k), \quad & u^W_{j,k} &= p^n_{j,k}(x_{j-\frac{1}{2}}, y_k).\end{aligned} \tag{21}$$

These interfaces are moving with the corresponding speeds

$$\begin{aligned} a^+_{j+\frac{1}{2},k} &:= \max\{\lambda_N(\frac{\partial f}{\partial u}(u^W_{j+1,k})), \lambda_N(\frac{\partial f}{\partial u}(u^E_{j,k})), 0\},\\ b^+_{j,k+\frac{1}{2}} &:= \max\{\lambda_N(\frac{\partial g}{\partial u}(u^S_{j,k+1})), \lambda_N(\frac{\partial g}{\partial u}(u^N_{j,k})), 0\},\\ a^-_{j+\frac{1}{2},k} &:= \min\{\lambda_1(\frac{\partial f}{\partial u}(u^W_{j+1,k})), \lambda_1(\frac{\partial f}{\partial u}(u^E_{j,k})), 0\},\\ b^-_{j,k+\frac{1}{2}} &:= \min\{\lambda_1(\frac{\partial g}{\partial u}(u^S_{j,k+1})), \lambda_1(\frac{\partial g}{\partial u}(u^N_{j,k})), 0\},\end{aligned} \tag{22}$$

where λ_N and λ_1 denote the largest and the smallest eigenvalues of the Jacobians $\frac{\partial f}{\partial u}$ and $\frac{\partial g}{\partial u}$, respectively. Using second-order midpoint rule to approximate the spatial integrals along the faces of side cells results in the second-order numerical fluxes

$$H^x_{j+\frac{1}{2},k} = \frac{a^+_{j+\frac{1}{2},k} f(u^E_{j,k}) - a^-_{j+\frac{1}{2},k} f(u^W_{j+1,k})}{a^+_{j+\frac{1}{2},k} - a^-_{j+\frac{1}{2},k}} + \frac{a^+_{j+\frac{1}{2},k} a^-_{j+\frac{1}{2},k}}{a^+_{j+\frac{1}{2},k} - a^-_{j+\frac{1}{2},k}}[u^W_{j+1,k} - u^E_{j,k}], \tag{23}$$

and

$$H^y_{j,k+\frac{1}{2}} = \frac{b^+_{j,k+\frac{1}{2}} g(u^N_{j,k}) - b^-_{j,k+\frac{1}{2}} g(u^S_{j,k+1})}{b^+_{j,k+\frac{1}{2}} - b^-_{j,k+\frac{1}{2}}} + \frac{b^+_{j,k+\frac{1}{2}} b^-_{j,k+\frac{1}{2}}}{b^+_{j,k+\frac{1}{2}} - b^-_{j,k+\frac{1}{2}}}[u^S_{j,k+1} - u^N_{j,k}]. \tag{24}$$

TABLE 1. L_1 errors and orders of central scheme with RK and LW time discretizations for 1-D linear advection equation (Example 1)

	RK		LW	
N	L_1 error	order	L_1 error	order
10	2.961E-01		3.036E-01	
20	7.688E-02	1.9454	7.974E-02	1.9286
40	1.957E-02	1.9738	2.169E-02	1.8783
80	3.724E-03	2.3938	6.054E-03	1.8383
160	8.920E-04	2.0617	1.043E-03	2.5398
320	1.902E-04	2.2295	1.664E-04	2.6479

TABLE 2. L_1 errors and orders of central scheme with RK and LW time discretizations for 1-D Burgers' equation (Example 2)

	RK		LW	
N	L_1 error	order	L_1 error	order
10	3.600E-03		3.598E-03	
20	9.545E-04	1.9152	9.555E-04	1.9128
40	2.106E-04	2.1804	2.114E-04	2.1761
80	5.573E-05	1.9178	5.591E-05	1.9189
160	1.257E-05	2.1489	1.260E-05	2.1501
320	3.076E-06	2.0306	3.079E-06	2.0327

4. Numerical experiments

Both central and central upwind schemes are second order accuracy in space. We apply 3rd order accuracy in time for both RK and LW time discretizations.

4.1. Numerical experiments in 1D

Example 1. 1-D scalar (linear advection) equation: Consider the initial value problem

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = \sin x, \quad 0 \le x \le 2\pi. \end{cases}$$

To compute the accuracy, we solve the linear advection equation with the smooth initial condition $\sin x$. The solution is computed at time $T = 1$ and the CFL is 0.008. We use the L_1 norm to compute the errors. Table 1 shows the L_1 errors and orders of the central scheme with RK and LW time discretizations. The results demonstrate that the order of both methods are 2nd and their accuracies are comparable.

TABLE 3. L_1 errors and orders of central upwind scheme with RK and LW time discretizations for 1-D Burgers' equation (Example 2)

	RK		LW	
N	L_1 error	order	L_1 error	order
10	3.607E-03		3.609E-03	
20	9.572E-04	1.9139	9.585E-04	1.9126
40	2.107E-04	2.1839	2.115E-04	2.1799
80	5.573E-05	1.9185	5.591E-05	1.9196
160	1.257E-05	2.1489	1.260E-05	2.1501
320	3.076E-06	2.0306	3.079E-06	2.0327

Example 2. 1-D scalar (Burgers') equation: Consider the initial value problem

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0 \\ u(x,0) = 0.5 + \sin \pi x, \quad 0 \le x \le 2. \end{cases}$$

Since Burgers' equation is nonlinear equation, the shock forms even though the initial condition is smooth. To check the accuracy, we compute the solution at $T = 0.5/\pi$ before shock is formed. The numerical solutions by central scheme with RK and LW time discretizations are shown in Figure 1 (a). Tables 2 and 3 show the errors and orders of central scheme and central upwind scheme with RK and LW time discretizations, respectively. As we see in linear advection equation case (Example 1), the results show that the order of both methods are 2nd and their accuracies are comparable. We also compute the solution by central scheme with RK and LW time discretizations at $T = 1.5/\pi$ after shock is formed. This solution is shown in Figure 1 (b). In this example, CFL is 0.00305 and the number of grids N is 80.

Example 3. 1-D scalar (Buckley-Leverett) equation: Consider the Riemann problem

$$u_t + \left(\frac{u^2}{u^2 + a(1-u)^2}\right)_x = 0, \quad a = 0.5$$

$$u(x,0) = \begin{cases} 1, & -0.5 \le x \le 0 \\ 0, & \text{otherwise.} \end{cases}$$

The numerical solutions are computed at $T = 0.4$ by central scheme with RK and LW time discretizations and they are shown in Figure 2. For this example, CFL is 0.05 and N is 160.

Example 4. 1-D system (Euler equations): One-dimensional Euler equations can be written as

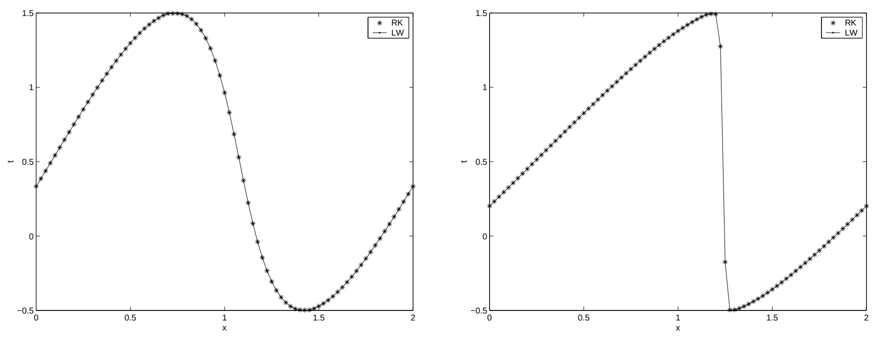
$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}_x = 0$$

where ρ, u, p, and E are density, velocity, pressure, and total energy, respectively. The above system is closed by the equation of state, $p = (\gamma - 1)(E - \rho u^2/2)$ and $\gamma = 1.4$.

Example 4.1. We consider the 1-D Riemann problem for Euler equations with the initial conditions

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & x \leq 0, \\ (0.5, 0, 0.571), & x > 0. \end{cases}$$

For this example, CFL is 0.05 and N is 200. Figure 3 (a) and (b) show the numerical solutions at time $T = 0.16$ by central scheme and central upwind with RK and LW time discretizations, respectively.



(a) Smooth solution at $T = 0.5/\pi$ (b) Shock solution at $T = 1.5/\pi$

FIGURE 1. Numerical solutions of cental scheme with RK and LW time discretizations for 1-D Burgers' equation (Example 2)

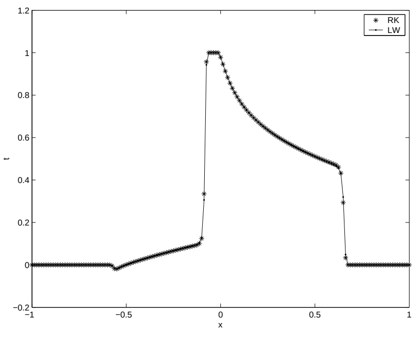


FIGURE 2. Numerical solutions of cental scheme with RK and LW time discretizations for 1-D Buckley-Leverett equation at time $T = 0.4$ (Example 3)

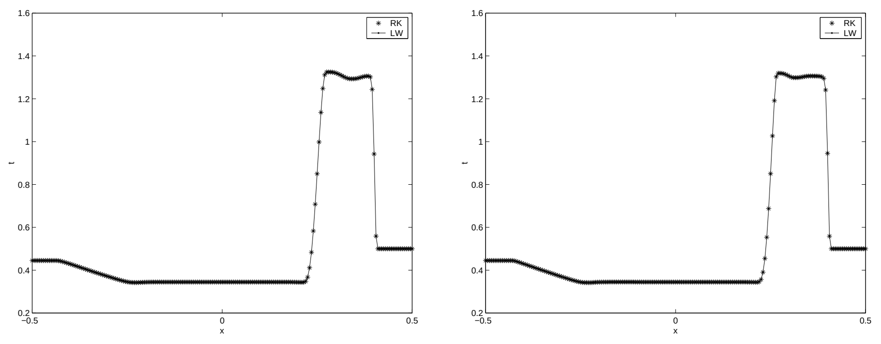
Example 4.2. We consider the 1-D Riemann problem for Euler equations with the initial conditions

$$(\rho, u, p) = \begin{cases} (1.0, 0, 1.0), & x \leq 0, \\ (0.125, 0, 0.1), & x > 0. \end{cases}$$

For this example, CFL is 0.05 and grid N is 200. Figure 4 (a) and (b) show the numerical solutions at time $T = 0.1644$ by central scheme and central upwind with RK and LW time discretizations, respectively.

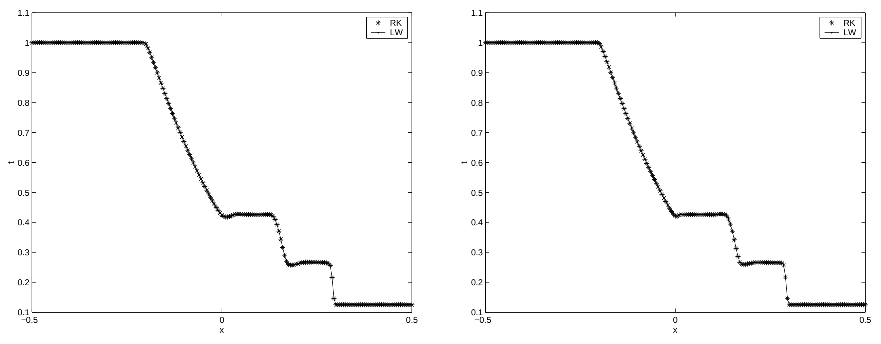
Example 4.3. We consider the 1-D Riemann problem for Euler equations with the initial conditions

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & x \leq -4, \\ (1 + 0.2 * \sin(5x), 0, 1), x > -4. \end{cases}$$



(a) central scheme (b) central upwind scheme

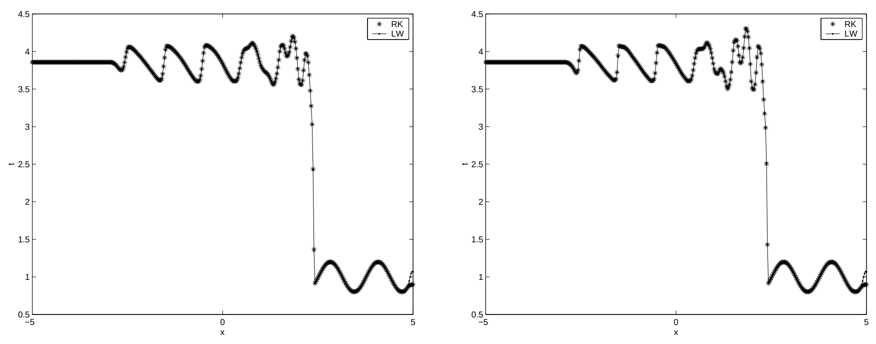
FIGURE 3. Numerical solutions of the density for Euler equations (Example 4.1)



(a) central scheme (b) central upwind scheme

FIGURE 4. Numerical solutions of the density for Euler equations (Example 4.2)

For this example, CFL is 0.05 and N is 400. Figure 5 (a) and (b) show the numerical solutions at time $T = 1.8$ by central scheme and central upwind with RK and LW time discretizations, respectively.



(a) central scheme (b) central upwind scheme

FIGURE 5. Numerical solutions of the density for Euler equations (Example 4.3)

Example 4.4. We consider the Riemann problem for Euler equations with the initial conditions

$$(\rho, u, p) = \begin{cases} (1, 0, 2500), & 0 \leq x \leq 0.1, \\ (1, 0, 0.025), & 0.1 \leq x \leq 0.9, \\ (1, 0, 250), & 0.9 \leq x \leq 1. \end{cases}$$

For this example, CFL is 0.475 and N is 400. Figure 6 (a), (b) and (c) show the numerical solutions at time $T = 0.01, T = 0.03$ and $T = 0.038$ by central scheme with RK and LW time discretizations, respectively. Figure 6 (a′), (b′) and (c′) show the numerical solutions at time $T = 0.01, T = 0.03$ and $T = 0.038$ by central upwind scheme with RK and LW time discretizations, respectively.

4.2. Numerical experiments in 2D

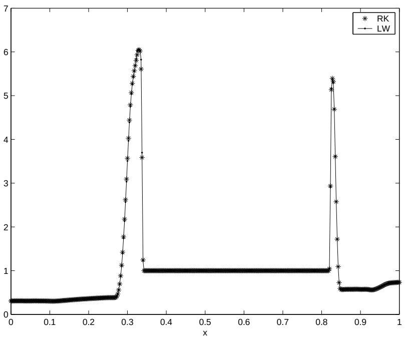
Example 5. 2-D scalar (linear advection) equation: Consider the initial value problem

$$\begin{cases} u_t + u_x + u_y = 0 \\ u(x, y, 0) = \sin(0.5 * (x + y)), \quad 0 \leq x, y \leq 2\pi. \end{cases}$$

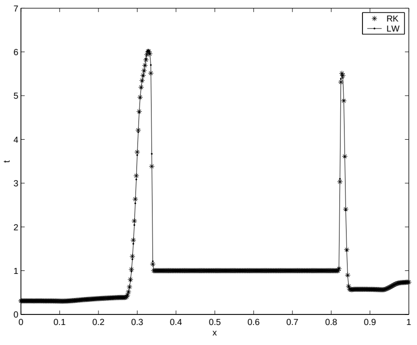
To check the accuracy in 2-D, we solve the 2-D linear advection equation with the smooth initial condition $\sin(0.5 * (x + y))$. The solution is computed at time $T = 1$ and the CFL is 0.006. Table 4 shows the L_1 errors and orders of the central scheme with RK and LW time discretizations. The results show that the order of both methods are 2nd and their accuracies are comparable.

Example 6. 2-D scalar (Burgers') equation: Consider the 2-D Burgers' equation

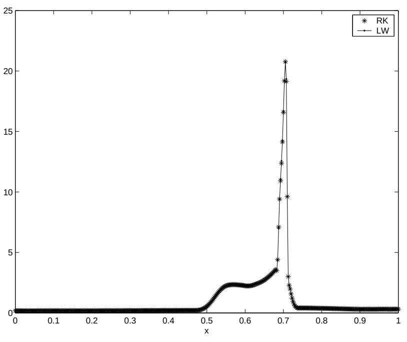
$$u_t + \left(\frac{1}{2}u^2\right)_x + \left(\frac{1}{2}u^2\right)_y = 0.$$



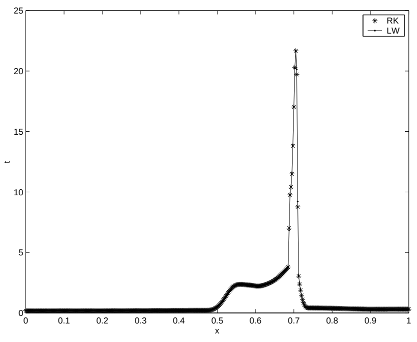
(a) central scheme(T=0.01)



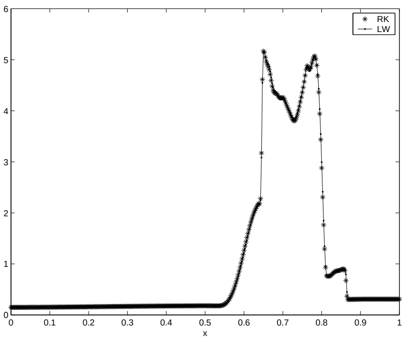
(a′) central upwind scheme(T=0.01)



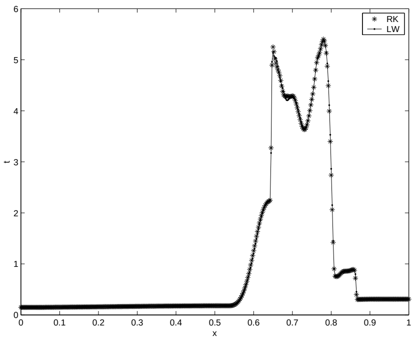
(b) central scheme(T=0.03)



(b′) central upwind scheme(T=0.03)



(c) central scheme(T=0.038)



(c′) central upwind scheme(T=0.038)

FIGURE 6. Numerical solutions of the density for Euler equations (Example 4.4)

TABLE 4. L_1 errors and orders of central scheme with RK and LW time discretizations for 2-D linear advection equation (Example 5)

	RK		LW	
$N_x * N_y$	L_1 error	order	L_1 error	order
10*10	8.243E+00		8.300E+00	
20*20	2.082E+00	1.9851	2.095E+00	1.9861
40*40	4.452E-01	2.2250	4.433E-01	2.2407
80*80	1.248E-01	1.8351	1.217E-01	1.8651
160*160	2.638E-02	2.2420	2.653E-02	2.1977
320*320	4.171E-03	2.6611	5.765E-03	2.2021

TABLE 5. L_1 errors and orders of central scheme with RK and LW time discretizations for 2-D Burgers' equation (Example 6.1)

	RK		LW	
$N_x * N_y$	L_1 error	order	L_1 error	order
10*10	4.445E-01		4.444E-01	
20*20	1.056E-01	2.0743	1.054E-01	2.0759
40*40	2.141E-02	2.3018	2.153E-02	2.2916
80*80	5.686E-03	1.9126	5.706E-03	1.9160
160*160	1.305E-03	2.1232	1.322E-03	2.1096
320*320	3.411E-04	1.9360	3.486E-04	1.9233

Example 6.1. Consider the initial value problem for 2-D Burgers' equation with the initial condition

$$u(x, y, 0) = 0.5 + \sin(0.5 * \pi(x + y)), \quad 0 \leq x, y \leq 4.$$

To check the accuracy, we compute the solution for 2-D nonlinear Burgers' equation at $T = 0.5/\pi$ before shock is formed. Tables 5 and 6 show the errors and orders of central scheme and central upwind scheme with RK and LW time discretizations, respectively. The results show that the order of both methods are 2nd and their accuracies are comparable just as in the case of linear advection equation (Example 5). In this example, CFL is 0.003.

Example 6.2. We consider the 2-D Riemann problem for Burgers' equation with the initial conditions

$$u(x, y, 0) = \begin{cases} 1 & 0 \leq x, 0 \leq y \\ 2 & 0 \geq x, 0 \leq y \\ 3 & 0 \geq x, 0 \geq y \\ 4 & 0 \leq x, 0 \geq y. \end{cases}$$

TABLE 6. L_1 errors and orders of central upwind scheme with RK and LW time discretizations for 2-D Burgers' equation (Example 6.1)

	RK		LW	
$N_x * N_y$	L_1 error	order	error	L_1 order
10*10	6.432E-01		6.435E-01	
20*20	1.630E-01	1.9807	1.629E-01	1.9818
40*40	3.649E-02	2.1590	3.657E-02	2.1555
80*80	9.113E-03	2.0016	9.142E-03	2.0001
160*160	2.375E-03	1.9399	2.393E-03	1.9337
320*320	5.157E-04	2.2034	5.262E-04	2.1852

TABLE 7. Comparisons of CPU time for 2-D Burgers' equation (Example 6.2)

	RK	LW
Central	187.89	87.84
Central Upwind	297.45	123.27

Figure 7 (a) and (a′) show the numerical solutions at time $T = 0.1$ by central scheme with RK and LW time discretizations, respectively. Figure 7 (b) and (b′) show the numerical solutions at time $T = 0.1$ by central upwind scheme with RK and LW time discretizations, respectively. As we see in Figure 7, one rarefaction and three shocks are formed in this example. For this example, the CFL is 0.05 and $N \times N$ is 200×200. To compare the efficiency of RK and LW time discretizations, we check the CPU time for central scheme and central upwind scheme with RK and LW time discretizations. The results are shown in Table 7. For central scheme, LW time discretization shows 53.2% savings against RK time discretization. For central upwind scheme, the LW time discretization shows 58.6% savings against the RK time discretization.

Example 6.3. We consider the 2-D Riemann problem for Burgers' equation with the initial conditions

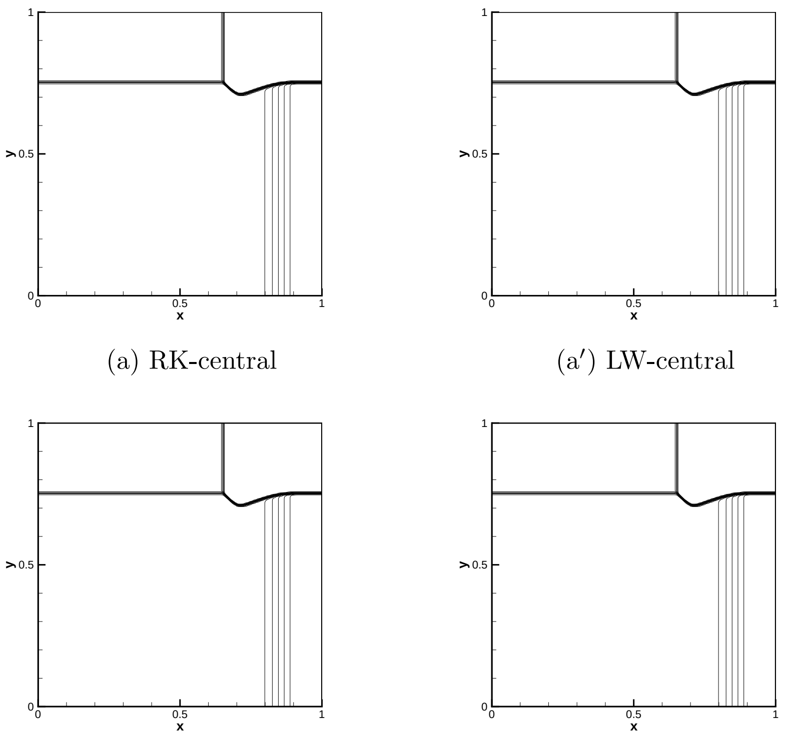
$$u(x,y,0) = \begin{cases} 4 & 0 \le x, 0 \le y \\ 2 & 0 \ge x, 0 \le y \\ 1 & 0 \ge x, 0 \ge y \\ 3 & 0 \le x, 0 \ge y. \end{cases}$$

Figure 8 (a) and (a′) show the numerical solutions at time $T = 0.1$ by central scheme with RK and LW time discretizations, respectively. Figure 8 (b) and (b′) show the numerical solutions at time $T = 0.1$ by central upwind scheme with RK and LW time discretizations, respectively. Figure 8 shows that four rarefactions are formed in this example. The CPU time for central scheme and

Table 8. Comparisons of CPU time for 2-D Burgers' equation (Example 6.3)

	RK	LW
Central	187.23	87.53
Central Upwind	268.13	122.88

central upwind scheme with RK and LW time discretizations are shown in Table 8. The LW time discretization shows 53.3% and 54.2% savings against the RK time discretization for central scheme and central upwind scheme, respectively. For this example, the CFL is 0.05 and $N \times N$ is 200×200.



(a) RK-central (a′) LW-central

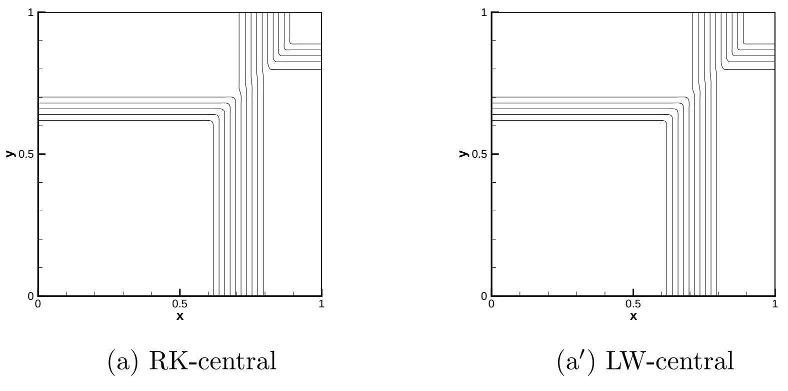
(b) RK-central upwind (b′) LW-central upwind

Figure 7. Numerical solutions for 2-D Burgers' equation (Example 6.2)

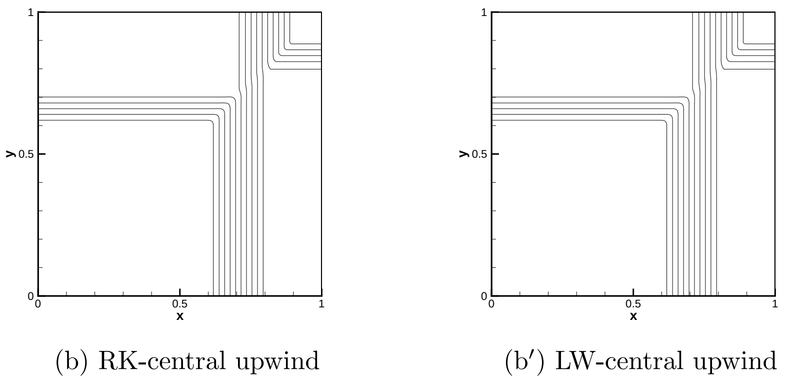
Example 7. 2-D system (Euler equations): Two-dimensional Euler equations can be written as

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}_y = 0,$$

where ρ, u, v, p, and E are density, x velocity, y velocity, pressure, and total energy, respectively. The above system is closed by the equation of state, $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2+v^2), \gamma = 1.4$. For this example, $CFL = 0.1$ and $N \times N = 200 \times 200$.



(a) RK-central (a′) LW-central



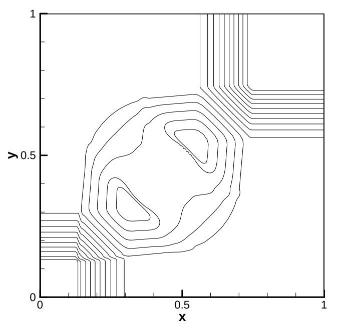
(b) RK-central upwind (b′) LW-central upwind

FIGURE 8. Numerical solutions for 2-D Burgers' equation (Example 6.3)

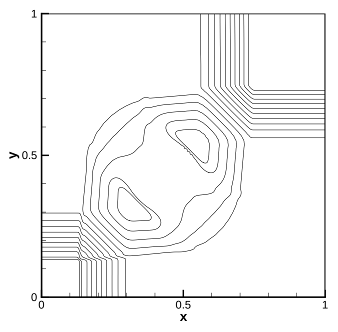
Example 7.1. We consider the 2-D Riemann problem for Euler equations with the initial conditions

$$(p,\rho,u,v)(x,y,0)=\begin{cases}(1,1,0,0) & \text{if } x>0.5 \text{ and } y>0.5\\ (0.4,0.5197,-0.7259,0) & \text{if } x<0.5 \text{ and } y>0.5\\ (1,1,-0.7259,-0.7259) & \text{if } x<0.5 \text{ and } y<0.5\\ (0.4,0.5197,0,-0.7259) & \text{if } x>0.5 \text{ and } y<0.5.\end{cases}$$

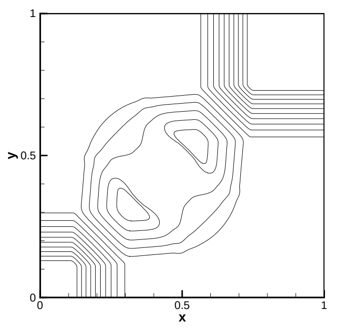
Figure 9 (a), (a′) and (b), (b′) show the numerical solutions at time $T = 0.2$ by central scheme and central upwind scheme with RK and LW time discretizations, respectively. The CPU time for central scheme and central upwind scheme with RK and LW time discretizations are shown in Table 9. The LW time discretization shows 48.6% and 53.7% savings against the RK time discretization for central scheme and central upwind scheme, respectively.



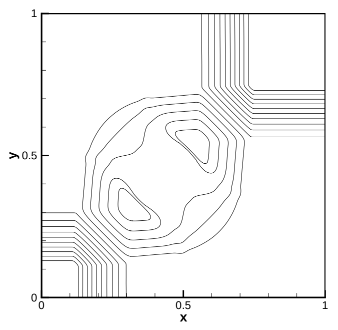
(a) RK-central



(a′) LW-central



(b) RK-central upwind



(b′) LW-central upwind

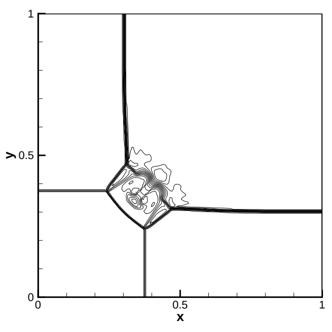
FIGURE 9. Numerical solutions of the density for 2-D Euler equations (Example 7.1)

TABLE 9. Comparisons of CPU time for 2-D Euler equations (Example 7.1)

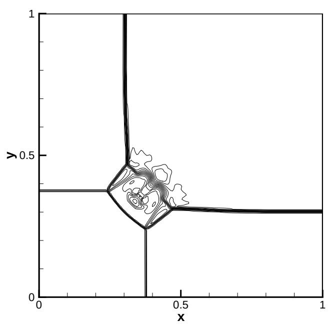
	RK	LW
Central	397.23	203.98
Central Upwind	607.44	281.22

Example 7.2. We consider the 2-D Riemann problem for Euler equations with the initial conditions

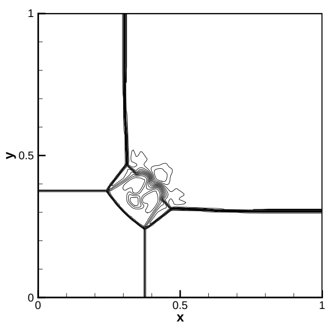
$$(p, \rho, u, v)(x, y, 0) = \begin{cases} (1.5, 1.5, 0, 0) & \text{if } x > 0.5 \text{ and } y > 0.5 \\ (0.3, 0.5323, 1.206, 0) & \text{if } x < 0.5 \text{ and } y > 0.5 \\ (0.029, 0.138, 1.206, 1.206) & \text{if } x < 0.5 \text{ and } y < 0.5 \\ (0.3, 0.5323, 0, 1.206) & \text{if } x > 0.5 \text{ and } y < 0.5. \end{cases}$$



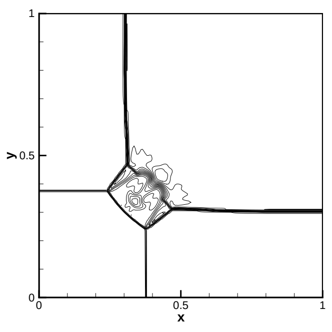
(a) RK-central



(a′) LW-central



(b) RK-central upwind



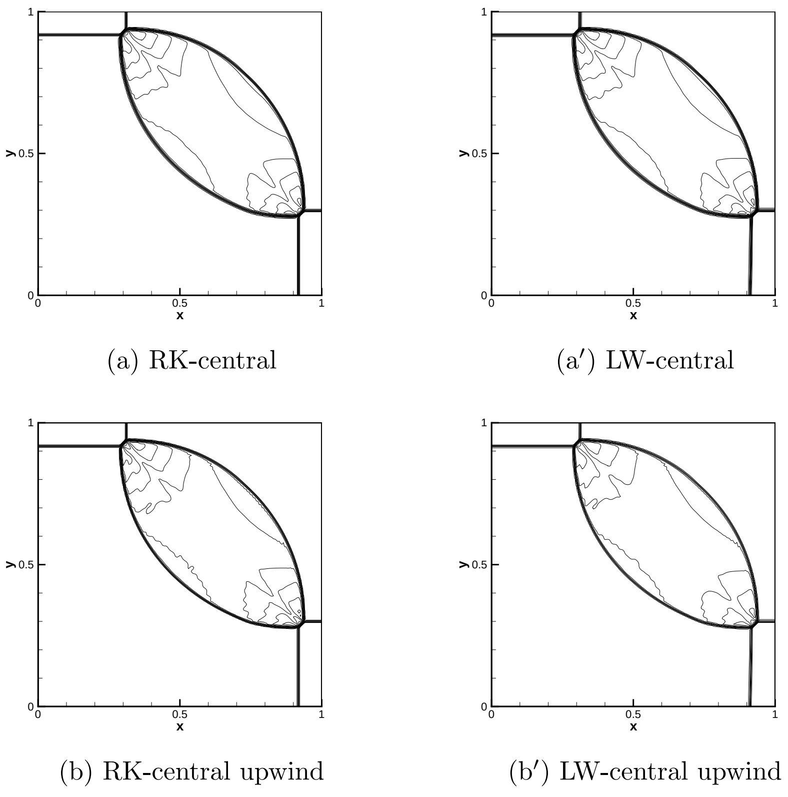
(b′) LW-central upwind

FIGURE 10. Numerical solutions of the density for 2-D Euler equations (Example 7.2)

Table 10. Comparisons of CPU time for 2-D Euler equations (Example 7.2)

	RK	LW
Central	693.64	354.22
Central Upwind	1004.00	460.67

Figure 10 (a), (a′) and (b), (b′) show the numerical solutions at time $T = 0.3$ by central scheme and central upwind scheme with RK and LW time discretizations, respectively. The CPU time for central scheme and central upwind scheme with RK and LW time discretizations are shown in Table 10. The LW time discretization shows 48.9% and 54.1% savings against the RK time discretization for central scheme and central upwind scheme, respectively.



(a) RK-central (a′) LW-central

(b) RK-central upwind (b′) LW-central upwind

Figure 11. Numerical solutions of the density for 2-D Euler equations (Example 7.3)

TABLE 11. Comparisons of CPU time for 2-D Euler equation (Example 7.3)

	RK	LW
Central	632.36	323.44
Central Upwind	986.39	447.56

Example 7.3. We consider the Riemann problem for Euler equations with the initial conditions

$$(p, \rho, u, v)(x, y, 0) = \begin{cases} (1.1, 1.1, 0, 0) & \text{if } x > 0.5 \text{ and } y > 0.5 \\ (0.35, 0.5065, 0.8939, 0) & \text{if } x < 0.5 \text{ and } y > 0.5 \\ (1.1, 1.1, 0.8939, 0.8939) & \text{if } x < 0.5 \text{ and } y < 0.5 \\ (0.35, 0.5065, 0, 0.8939) & \text{if } x > 0.5 \text{ and } y < 0.5. \end{cases}$$

Figure 11 (a), (a′) and (b), (b′) show the numerical solutions at time $T = 0.25$ by central scheme and central upwind scheme with RK and LW time discretizations, respectively. The CPU time for central scheme and central upwind scheme with RK and LW time discretizations are shown in Table 11. The LW time discretization shows 48.9% and 54.6% savings against the RK time discretization for central scheme and central upwind scheme, respectively.

5. Conclusions

We apply the Lax-Wendroff type (LW) time discretization for central scheme and central upwind scheme. We test for various examples including 1-D and 2-D linear and nonlinear scalar equations and nonlinear systems such as Euler equations. The results show that the LW time discretization maintains the same order in accuracy as the Runge-Kutta (RK) time discretization, but it saves much in CPU time. For central scheme, the LW time discretization shows 48.6−53.3% savings against the RK time discretization. For central upwind scheme, the LW time discretization shows 53.7 − 58.6% savings against the RK time discretization. The LW time discretization is much more efficient in CPU time than the RK time discretization for central scheme and especially for central upwind scheme. We conclude that even though the LW time discretization is a little bit messier than RK, LW greatly reduces the CPU time for both central and central upwind scheme.

References

[1] W. Hwang and W. B. Lindquist, *The 2-dimensional Riemann problem for a* 2×2 *hyperbolic conservation law. I. Isotropic media*, SIAM J. Math. Anal. **34** (2002), no. 2, 341–358.

[2] ———, *The 2-dimensional problem for a* 2×2 *hyperbolic conservation law. II. Anisotropic media*, SIAM J. Math. Anal. **34** (2002), no. 2, 359–384.

[3] G. Jiang and E. Tadmor, *Nonoscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J. Sci. Comput. **19** (1998), no. 6, 1892–1917.

[4] A. Kurganov, S. Noelle, and G. Petrova, *Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations*, SIAM J. Sci. Comput. **23** (2001), no. 3, 707–740.

[5] A. Kurganov and G. Petrova, *A third-order semi-discrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems*, Numer. Math. **88** (2001), no. 4, 683–729.

[6] A. Kurganov and E. Tadmor, *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys. **160** (2000), no. 1, 241–282.

[7] ______, *Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers*, Numer. Methods Partial Differential Equations **18** (2002), no. 5, 584–608.

[8] X. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys. **115** (1994), no. 1, 200–212.

[9] H. Nessyahu and E. Tadmor, *Nonoscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys. **87** (1990), no. 2, 408–463.

[10] J. Qiu, *Hermite WENO schemes with Lax-Wendroff type time discretizations for Hamilton-Jacobi equations*, J. Comput. Math. **25** (2007), no. 2, 131–144.

[11] ______, *WENO schemes with Lax-Wendroff type time discretizations for Hamilton-Jacobi equations*, J. Comput. Appl. Math. **200** (2007), no. 2, 591–605.

[12] J. Qiu and C. Shu, *Finite difference WENO schemes with Lax-Wendroff-Type time discretizations*, SIAM J. Sci. Comput. **24** (2003), no. 6, 2185–2198.

[13] D. Yoon and W. Hwang, *Two-dimensional Riemann problem for Burger's equation*, Bull. Korean Math. Soc. **45** (2008), no. 1, 191–205.

Suyeon Shin
Department of Mathematics
Korea University
Seoul 136-701, Korea
E-mail address: angelic52@korea.ac.kr

Woonjae Hwang
Department of Information and Mathematics
Korea University
Jochiwon 739-700, Korea
E-mail address: woonjae@korea.ac.kr