



Central random choice methods for hyperbolic conservation laws

Yousef Hashem Zahran¹ · Amr H. Abdalla¹

Received: 25 February 2022 / Revised: 1 July 2022 / Accepted: 2 November 2022
© The Author(s) 2022

Abstract

In this article, we briefly review the random choice method (RCM) and ADER methods for solving one and two-dimensional hyperbolic conservation laws. The main advantage of RCM is that it computes discontinuities with infinite resolution. In this method, the original problem is reduced to a set of local Riemann problems (RPs). The exact solutions of these RPs are used to form the solution of the original problem. However, RCM has the following disadvantages: (1) one should solve the RP exactly, however, the exact solutions are usually complex and unavailable for many problems. (2) The accuracy of the smooth region of the flow is poor. ADER methods are explicit, one-step schemes with a very high order of accuracy in time and space. They depend on the solution of the generalized RP (GRP) exactly. In Zahran (J Math Anal Appl 346:120–140, 2008), an improved version of ADER methods (central ADER) was introduced where the RPs were solved numerically and used central fluxes, instead of upwind fluxes. The improved central ADER schemes are more accurate, faster, simple to implement, RP solver free, and need less computer memory. To fade the drawbacks of the above schemes and keep their advantages, we propose, in this paper, an improved version of the RCM. We merge the central ADER technique with the RCM. The resulting scheme is called Central RCM (CRCM). The improvements are listed as follows: we use the WENO reconstruction for the initial data instead of constant reconstruction in RCM, we solve the RPs numerically by using central finite difference schemes and use random sampling to update the solution, as the original RCM. Here we use the staggered and non-staggered RCM. To enhance the accuracy of the new methods, we use a third-order TVD flux (Zahran in Bull Belg Math Soc Simon Stevin 14:259–275, 2007), instead of a first-order flux. Compared with the original RCM and the central ADER, the new methods combine the advantages of RCM, ADER, and central finite difference methods as follows: more accurate, very simple to implement, need less computer memory, and RP solver free. Moreover, the new

✉ Yousef Hashem Zahran
dryhhz@yahoo.com

¹ Physics and Mathematics Department, Port Said University, Faculty of Engineering, Port Said, Egypt

methods capture the discontinuities with infinite resolution and improve the accuracy of the smooth parts. The new methods have less CPU time than the central ADER methods, this is due to less flux evaluation in CRCM. An extension of the schemes to general systems of nonlinear hyperbolic conservation laws in one and two dimensions is presented. We present several numerical examples for one and two-dimensional problems. The results confirm that the presented schemes are superior to the original RCM, ADER, and central ADER schemes.

Keywords Hyperbolic conservation laws · WENO schemes · RCM · ADER methods · ℓ SSPRK Runge–Kutta · Euler equations · Burgers equation

Mathematics Subject Classification Primary 65M10; Secondary 65M05

1 Introduction

Hyperbolic conservation laws are very important and arise in many fields such as shallow water flow, gas dynamics, weather prediction, and many other fields. The exact solutions of such equations are only available in a few special cases. So, we should use numerical methods. Since the nonlinear hyperbolic conservation laws develop discontinuities in their solution even if the initial conditions are smooth. Therefore, if the solutions contain discontinuities as well as rich structures, the numerical schemes should avoid the spurious oscillations as well as be high order accuracy. It is well known that the central methods are more efficient than the upwind methods in the smooth regions of the solutions since the central methods give a less dissipative solution. However, near discontinuities, to avoid the numerical instability, the upwind scheme is better than the central scheme. Thus successful methods should be high order of accuracy in smooth regions and present the discontinuities with the correct position and without spurious oscillations.

In the past, six decades or more, there are many high order numerical high resolution methods have been developed, such as Total Variation Diminishing (TVD) [6], random choice method (RCM) [2, 4, 16], weighted essentially non-oscillatory schemes (WENO) [1, 9, 24, 25], ADER methods [11–15, 17, 18, 23], and so on. These methods produce satisfactory numerical results. There are a lot of studies on them that have been done to improve the computational robustness, accuracy, and efficiency and reduce dissipation and dispersion errors. TVD schemes, introduced by Harten [6], are at most first-order accurate near discontinuities and smooth extrema and thus they are not suitable for many applications areas especially, acoustics, compressible turbulence, and problems with long-time evolution wave propagation. In such applications, the extrema are clipped with time evolution and thus the numerical dissipation may become dominant. To fade this flaw, uniformly very high order methods are required for such applications. For example, the weighted essentially non-oscillatory (WENO) schemes and ADER methods. WENO schemes are essentially non-oscillatory. On the other side, RCM is a shock-capturing method. It is neither a finite element nor a finite difference method. The main advantage of the RCM is that it captures discontinuities (shocks and contact surfaces) with infinite resolution (zero width). The first step in the

RCM is to suppose that the initial data at a given time level t^n , of the original problem, is approximated by a piece-wise constant function. Thus, the original problem is reduced to a set of local Riemann problems (RPs). The exact solution of these local RPs is pieced together to form the solution at the next time level t^{n+1} . RCM takes the updated solution at the next time level t^{n+1} to be computed from these solutions by using a sequence of random numbers. RCM has the following flaws: 1- although RCM captures the discontinuities with zero width, this property comes at a cost, one should solve the RPs exactly. However, the exact solution of RP is usually complex and unavailable for many hyperbolic problems for practical interest. 2- the accuracy in smooth regions of the flow, such as rarefactions, is poor. ADER methods are explicit, one-step schemes with a very high order of accuracy in time and space. They depend on the solution of the generalized RP (GRP). In ADER schemes the initial data are approximated by piecewise polynomial functions of arbitrary order, unlike piece-wise constant in RCM. The ADER methods [12–15] were developed as an extension of the Godunov upwind methods [3]. By using Taylor expansion of the state (or flux) functions, the GRP is reduced to a set of m conventional RPs. The first RP is non-linear and the remaining ones are linear RPs of the k -th order spatial derivatives of the initial conditions, with $k = 0, 1, \dots, m$, where m is the order of accuracy (arbitrary) of the scheme. To compute the upwind flux, in ADER methods, the RPs are solved exactly. Again, the exact solutions of RPs are mostly unavailable and more complex. In [23], an improved version of ADER methods (central ADER) was introduced where the RPs were solved numerically and used central fluxes, instead of upwind fluxes. The improved central ADER schemes are more accurate, faster, simple to implement, RP solver free, and need less computer memory. To fade the drawbacks of the above schemes and keep their advantages, we propose, in this paper, an improved version of the RCM. We merge the central ADER technique with RCM. The resulting scheme is called Central RCM (CRCM). The improvements are fourfold. Firstly, we propose to reconstruct the initial data by piece-wise polynomial functions of arbitrary order. Here we use the WENO reconstruction. Secondly, we use Taylor expansion of the state functions as ADER methods, and therefore the original problem is reduced to a set of m conventional RPs. Thirdly, we solve these RPs numerically by using central finite difference schemes. Here we use two different finite difference schemes, fully discrete and semi-discrete schemes. Fourthly, we use random sampling, to update the solution, as the original RCM. Moreover, we use the staggered and non-staggered RCM. To enhance the accuracy of the new methods, we use a third-order TVD flux [22], instead of a first-order Lax–Friedrichs (LxF) flux. Compared with the original RCM and the central ADER, the new methods combine the advantages of RCM, ADER, and central finite difference methods as follows: more accurate, very simple to implement, and need less computer memory and RP solver free. Moreover, the new methods capture the discontinuities with infinite resolution (zero-width) and improve the accuracy of the smooth parts. This is due to using the central finite difference schemes for both the function and its derivatives. Moreover, the new methods have a faster computing time than the central ADER method, this is due to less flux evaluation in CRCM.

The paper consists of the following: In Sect. 2 we review the original RCM. In Sect. 3 we briefly review the ADER method. The central RCM on both staggered and non-staggered grids is described in Sect. 4. In Sect. 5 the accuracy of the new schemes is

discussed. In Sect. 6 the new methods are tested by solving some numerical examples for one and two-dimensional equations. Finally, in Sect. 7 we summarize the main findings of this work and give some outlook on future work.

2 Random choice method

In this section, we review the RCM on both staggered and non-staggered grids [16, 19].

2.1 RCM on non-staggered grids

Firstly, we consider the one-dimensional system of conservation laws, namely

$$u_t + [f(u)]_x = 0 \tag{2.1}$$

with initial and boundary conditions. Where $u(x,t)$ is the vector of conservative variables and $f(u)$ is the flux vector. Throughout this article, we consider regular grids. Let Δx and Δt be small spatial and time scales, $x_j = j \Delta x$, $x_{j \pm \frac{1}{2}} = x_j \pm \frac{1}{2} \Delta x$, $t^n = n \Delta t$, $u_j^n = u(x_j, t^n)$ and the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. Suppose that the solution u_j^n at the time $t = t^n$ is given. To compute the solution at the next time step t^{n+1} , the RCM assumes that the data at t^n for (2.1) can be approximated by a piecewise constant function over the cell I_j as

$$u(x, t) = u_j^n, \quad x_{j-\frac{1}{2}} \leq x \leq x_{j+\frac{1}{2}} \tag{2.2}$$

Therefore the problem (2.1) becomes a sequence of local Riemann Problems $\{RP(j, j + 1)\}$, that is the initial value problem (2.1) subject to the initial condition

$$u(x, t^n) = \begin{cases} u_j^n, & x < x_{j+\frac{1}{2}} \\ u_{j+1}^n, & x \geq x_{j+\frac{1}{2}} \end{cases} \tag{2.3}$$

The exact solutions to these Riemann Problems are pieced together to give the solution for the next time level t^{n+1} . Each local problem has an exact solution as shown in Fig. 1 (for Euler equations) [19].

The RCM updates the solution u_j^n in I_j at t^n to the u_j^{n+1} at t^{n+1} , for sufficiently small Δt , in two steps as follows:

Step 1 Solve both RP $(j - 1, j)$ and RP $(j, j + 1)$ exactly, to obtain the solutions at t^{n+1} .

Step 2 Sample these solutions at t^{n+1} in the cell I_j at a random position $u(Q_j, t^{n+1})$ where $Q_j = (x_{j-\frac{1}{2}} + \theta^n \Delta x, t^{n+1})$. This sampling depends on a quasi-random number $\theta^n \in [0, 1][0, 1]$, see [16]. The updated solution u_j^{n+1} can be written in the form

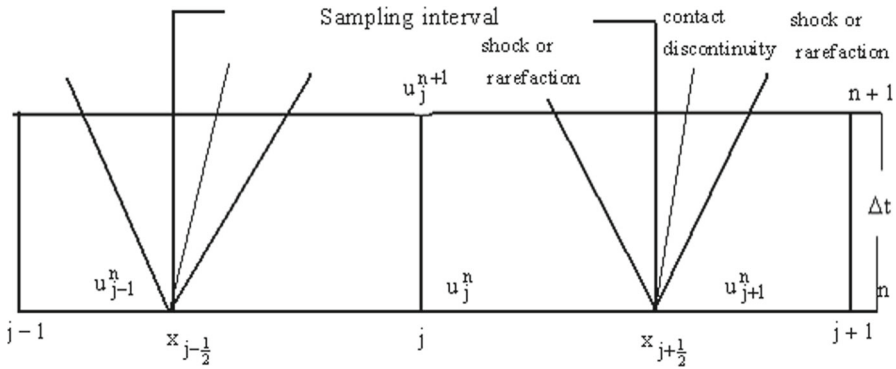
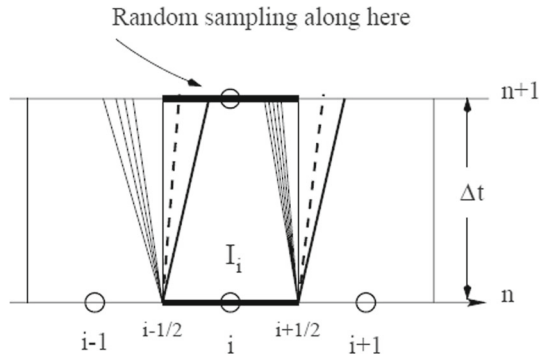


Fig. 1 Solution of local Riemann problems RP (j - 1, j) and RP(j, j + 1)

Fig. 2 RCM sampling on non-staggered grids



$$u_j^{n+1} = \begin{cases} u(x_{j-\frac{1}{2}} + \theta^n \Delta x) & 0 \leq \theta^n \leq \frac{1}{2} \\ u(x_{j+\frac{1}{2}} + (\theta^n - 1)\Delta x) & \frac{1}{2} \leq \theta^n \leq 1 \end{cases} \quad (2.4)$$

This sampling is shown in Fig. 2, [19]. It is clear that for $\theta^n = 0$, Q_j lies on the inter-cell boundary $x = x_{j-\frac{1}{2}}$ and therefore the solution u_j^{n+1} is the exact solution of RP (j - 1, j) at this position. For $\theta^n = 1$, Q_j lies on the right boundary and the solution $x = x_{j+\frac{1}{2}}$ and the solution u_j^{n+1} equals the exact solution of RP(j, j + 1). For $\theta^n = 0.5$, the solution $u_j^{n+1} = u_j^n$, i.e., the same old value. See [19] for more details.

2.2 RCM on staggered grids

In this section, we describe the RCM on a staggered grid to solve the problem (2.1). It updates u_j^n to the u_j^{n+1} in two steps as follows: [19]

Step 1 Solve the Riemann problems RP $(j - 1, j)$ and RP $(j, j + 1)$ to find the solutions $\bar{u}_{j \pm \frac{1}{2}}^{n+\frac{1}{2}}$ at $t^{n+\frac{1}{2}}$.

Step 2. Sample these solutions at $t^{n+\frac{1}{2}}$ as

$$\begin{aligned} u_{j-\frac{1}{2}}^{n+\frac{1}{2}} &= \bar{u}_{j-\frac{1}{2}}^{n+\frac{1}{2}}(x_{j-\frac{1}{2}} + (\theta^n - \frac{1}{2})\Delta x, t^{n+\frac{1}{2}}) & 0 \leq \theta^n \leq 1 \\ u_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= \bar{u}_{j+\frac{1}{2}}^{n+\frac{1}{2}}(x_{j+\frac{1}{2}} + (\theta^n - \frac{1}{2})\Delta x, t^{n+\frac{1}{2}}) & 0 \leq \theta^n \leq 1 \end{aligned} \tag{2.5}$$

Step 3. Solve the RP $(u_{j-\frac{1}{2}}^{n+\frac{1}{2}}, u_{j+\frac{1}{2}}^{n+\frac{1}{2}})$, exactly, to obtain the solution $\bar{u}_j^{n+1}(x, t)$ and then sample it, randomly, to get $u_j^{n+1}(x, t)$ as

$$u_j^{n+1} = \bar{u}_j^{n+1}(x_{j-\frac{1}{2}} + \theta^{n+1}\Delta x, t^{n+\frac{1}{2}}) \quad 0 \leq \theta^{n+1} \leq 1 \tag{2.6}$$

Note: we choose the time step Δt according to the CFL condition.

$$CFL \leq 0.25 \tag{2.7}$$

where $CFL = \max_j \left(S_j^n \frac{\Delta t}{\Delta x} \right)$, S_j^n is the maximum propagation speed in I_j at time level n .

3 ADER methods

Here we review the ADER methods for a system of hyperbolic conservation laws

$$u_t + f(u)_x = 0, \quad -\infty < x < \infty, \quad t \geq 0, \quad u(x, 0) = u_0(x) \tag{3.1}$$

This equation may be written in the non-conservative form

$$u_t + \lambda(u) u_x = 0, \tag{3.2}$$

where $\lambda(u) = \frac{df}{du}$ is the characteristic speed. We integrate (3.1) over the control volume in x - t space $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [t^n, t^{n+1}]$ to get the fully discrete scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left[F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}} \right] \tag{3.3}$$

Here u_j^n is the space average of the solution in the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ at t^n

$$u_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t^n) dx \tag{3.4a}$$

and the flux $F_{j+\frac{1}{2}}$ is the time average of the flux at the cell interface $x_{j+\frac{1}{2}}$:

$$F_{j+\frac{1}{2}} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(u(x_{j+\frac{1}{2}})) dt, \tag{3.4b}$$

The ADER methods define the numerical fluxes $F_{j+\frac{1}{2}}$ in such a way that the scheme (3.3) calculates the numerical solutions of (3.1) to arbitrary high order in both space and time. They rely on the solution of GRP (exactly) with the initial condition consisting of polynomial functions of arbitrary order, unlike piece-wise constant construction in RCM. We will mention the details in the next subsections.

3.1 Original (upwind) ADER method

Let the solution at time t^n , u_j^n , be known, the ADER methods [11–13] update the solution at the next time t^{n+1} (u_j^{n+1}) in the following steps:

Step 1: Reconstruction

The first step is to generate a piece-wise polynomial reconstruction from the cell averages i.e.,

$$u(x, t) = P_j^n(x), \quad x_{j-\frac{1}{2}} \leq x \leq x_{j+\frac{1}{2}} \tag{3.5}$$

where the polynomial $P_j^n(x)$, defined on I_j , must be conservative, high order accurate, and non-oscillatory. Thus, we obtain at each cell interface $x_{j+\frac{1}{2}}$ the following local GRP($j, j + 1$):

$$u_t + f(u)_x = 0, \quad \left. \begin{aligned} u_L(x) &= P_j(x, t), & x < x_{j+\frac{1}{2}} \\ u_R(x) &= P_{j+1}(x, t), & x > x_{j+\frac{1}{2}} \end{aligned} \right\} \tag{3.6}$$

It is well known that WENO reconstruction produces more accurate, non-oscillatory results and therefore we used here the WENO reconstruction in the design of our schemes.

Step 2 State expansion

Firstly, we compute the approximate solution $u(x_{j+\frac{1}{2}}, \tau)$, where τ is local time $\tau = t - t^n$ which is sufficiently small. Using Taylor expansion of the interface state in time, we get:

$$u(x_{j+\frac{1}{2}}, \tau) = u(x_{j+\frac{1}{2}}, 0+) + \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial t^k} u(x_{j+\frac{1}{2}}, 0+) \right\} \frac{\tau^k}{k!} \tag{3.7}$$

where $0+ = \lim_{t \rightarrow 0+} t$. Secondly, apply Cauchy-Kowaleski [11] procedure to replace all time derivatives in (3.7) by space derivatives. Equation (3.7) becomes

$$u(x_{j+\frac{1}{2}}, \tau) = u(x_{j+\frac{1}{2}}, 0+) + \sum_{k=1}^{m-1} \left\{ A^{(k)}(u_x^{(0)}(x_{j+\frac{1}{2}}, 0+), \dots, u_x^{(k)}(x_{j+\frac{1}{2}}, 0+)) \right\} \frac{\tau^k}{k!} \tag{3.8}$$

The first term $u(x_{j+\frac{1}{2}}, 0+)$ is obtained by solving the conventional (piece-wise constant) RP:

$$\left. \begin{aligned} u_t + f(u)_x &= 0, \\ u(x, 0) &= \begin{cases} u_L(x_{j+\frac{1}{2}}), & x < x_{j+\frac{1}{2}} \\ u_R(x_{j+\frac{1}{2}}), & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \right\} \tag{3.9}$$

This is done by using the upwind Godunov method. The remaining terms $u_x^{(k)}(x_{j+\frac{1}{2}}, 0+)$, $(k = 1, 2, \dots, m-1)$ are obtained by solving the k -th order derivative RPs which are locally linearised around the leading term $u_x^{(0)}(x_{j+\frac{1}{2}}, 0+)$ of (3.8):

$$\left. \begin{aligned} v_t + \lambda(u(x_{j+\frac{1}{2}}, 0+))v_x &= 0, \\ v(x, 0) &= \begin{cases} u_L^{(k)}(x_{j+\frac{1}{2}}), & x < x_{j+\frac{1}{2}} \\ u_R^{(k)}(x_{j+\frac{1}{2}}), & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \right\} \tag{3.10}$$

where $v = u_x^{(k)}$. The numerical flux $F_{j+\frac{1}{2}}$ of Eq. (3.3) is evaluated by using the m -th order accurate Gaussian rule

$$F_{j+\frac{1}{2}} = \sum_{\alpha=0}^N F(u(x_{j+\frac{1}{2}}, \gamma_\alpha \Delta t)) K_\alpha \tag{3.11}$$

where γ_j and K_j are scaled nodes and weights of the rule and N is the number of nodes [11].

3.2 Central ADER method

However, the high order accuracy of the upwind ADER schemes comes at a cost, RPs must be solved exactly. The exact solutions of the RPs are mostly unavailable and more complex. To overcome this problem, central ADER schemes were presented in [23]. In these methods, central fluxes are used instead of upwind fluxes as the building block. These fluxes are the central first-order monotone LxF flux and the third-order TVD flux [22]. Now, we give a brief review of the central ADER methods.

To evaluate the terms in (3.8), we proceed as follows:

For the first term $u(x_{j+\frac{1}{2}}, 0+)$, we use the staggered form of the central LxF scheme to solve the RP (3.9):

$$u_{j+\frac{1}{2}}^{0+} = \frac{1}{2}[u_L(x_{j+\frac{1}{2}}, 0) + u_R(x_{j+\frac{1}{2}}, 0)] - \frac{\alpha_0 \Delta t}{\Delta x}[f(u_R(x_{j+\frac{1}{2}}, 0)) - f(u_L(x_{j+\frac{1}{2}}, 0))] \tag{3.12}$$

For $u_x^{(k)}(x_{j+\frac{1}{2}}, 0+)$, ($k = 1, 2, \dots, m-1$), are obtained by solving the k -th order derivative RPs which are locally linearized around the leading term $u_x^{(0)}(x_{j+\frac{1}{2}}, 0+)$ of (3.8) and they are computed by staggered LxF scheme:

$$v_{j+\frac{1}{2}}^{0+} = \frac{1}{2}[v_L(x_{j+\frac{1}{2}}, 0) + v_R(x_{j+\frac{1}{2}}, 0)] - \lambda \frac{\alpha_k \Delta t}{\Delta x}[v_R(x_{j+\frac{1}{2}}, 0) - v_L(x_{j+\frac{1}{2}}, 0)] \tag{3.13}$$

here the time weight $\alpha_k = \frac{1}{(k+1)^k}$, $k \geq 1$ see [11, 23]. It is easy to see that α_k lie in the interval $[0.5, 1]$ and we take $\alpha_0 = 0.36788$. To get the flux $F_{j+\frac{1}{2}}$ of (3.2), there are two options. The first option is the m -th order accurate Gaussian rule. The second option is the flux expansion in which Taylor expansion of the flux directly is used [23]. To improve the accuracy of the central ADER method, the third-order TVD flux [22] is used rather than the first-order flux as the building block. For details see [23].

4 Central random choice methods

To fade the drawbacks of the above schemes and maintain their advantages, we merge the central ADER technique into RCM to obtain an improved version of RCM. We call these methods Central RCM (CRCM). We consider the system of hyperbolic conservation laws

$$u_t + f(u)_x = 0, \quad -\infty < x < \infty, \quad t \geq 0, \quad u(x, 0) = u_0(x) \tag{4.1}$$

This equation may be written in the non-conservative form

$$u_t + \lambda(u)u_x = 0, \tag{4.2}$$

where $\lambda(u) = \frac{df}{du}$ is the characteristic speed. We integrate (4.1) over the control volume in x - t space $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [t^n, t^{n+1}]$ to get the fully discrete finite difference scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left[F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}} \right] \tag{4.3}$$

u_j^n is the space average of the solution in the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ at t^n and the flux $F_{j+\frac{1}{2}}$ is the time average of the flux at the cell interface $x_{j+\frac{1}{2}}$:

$$u_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t^n) dx \quad F_{j+\frac{1}{2}} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(u(x_{j+\frac{1}{2}})) dt, \tag{4.4}$$

There are two types of finite difference schemes to compute the numerical solutions; one-step and two-step finite difference schemes.

4.1 One-step RCM on non-staggered grids

Assuming that the solution at time t^n , u_j^n is known, our task is to get the solution at the next time $t^{n+1}(u_j^{n+1})$. The implementation of this method is described in the following steps:

Step 1:

The first step is to generate a piece-wise polynomial reconstruction from the cell averages i.e.

$$u(x, t) = P_j^n(x), \quad x_{j-\frac{1}{2}} \leq x \leq x_{j+\frac{1}{2}} \tag{4.5}$$

where the polynomial $P_j^n(x)$, defined on I_j , must be conservative, high order accurate, and non-oscillatory. We used here the WENO reconstruction in the design of our schemes. Thus, we obtain at each cell interface $x_{j+\frac{1}{2}}$ the following local RP($j, j + 1$):

$$u_t + f(u)_x = 0, \quad u(x, 0) = \left\{ \begin{array}{ll} u_L(x) = P_j(x, t), & x < x_{j+\frac{1}{2}} \\ u_R(x) = P_{j+1}(x, t), & x > x_{j+\frac{1}{2}} \end{array} \right\} \tag{4.6}$$

Step 2

We solve the local RPs $RP(j - 1, j)$ and $RP(j, j + 1)$, numerically, by using the staggered LxF scheme of (4.1)

$$u_{j-\frac{1}{2}}^{n+1} = \frac{1}{2}[u_L(x_{j-\frac{1}{2}}, t^n) + u_R(x_{j-\frac{1}{2}}, t^n)] - \frac{\alpha_0 \Delta t}{\Delta x}[f(u_R(x_{j-\frac{1}{2}}, t^n)) - f(u_L(x_{j-\frac{1}{2}}, t^n))] \tag{4.7a}$$

$$u_{j+\frac{1}{2}}^{n+1} = \frac{1}{2}[u_L(x_{j+\frac{1}{2}}, t^n) + u_R(x_{j+\frac{1}{2}}, t^n)] - \frac{\Delta t}{\Delta x}[f(u_R(x_{j+\frac{1}{2}}, t^n)) - f(u_L(x_{j+\frac{1}{2}}, t^n))] \tag{4.7b}$$

For the case of the piece-wise constant data $u_L(x_{j-\frac{1}{2}}) = u_{j-1}^n$ and $u_R(x_{j-\frac{1}{2}}) = u_j^n$, the scheme (4.7a) becomes a monotone first order LxF scheme in the staggered form [10]

$$u_{j-\frac{1}{2}}^{n+1} = \frac{1}{2}[u_{j-1}^n + u_j^n] - \frac{\alpha_0 \Delta t}{\Delta x}[f(u_j^n) - f(u_{j-1}^n)] \tag{4.8a}$$

Similarly, the scheme (4.7b) leads to

$$u_{j+\frac{1}{2}}^{n+1} = \frac{1}{2}[u_j^n + u_{j+1}^n] - \frac{\alpha_0 \Delta t}{\Delta x}[f(u_{j+1}^n) - f(u_j^n)] \tag{4.8b}$$

Then we use a Taylor expansion to compute the updated solution at a random position $Q_j = (x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+1})$ in the x-t plane:

$$u(x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+1}) = u(x_{j+\frac{1}{2}}, t^{n+1}) + \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial x^k} u(x_{j+\frac{1}{2}}, t^{n+1}) \right\} \frac{(\theta^n \Delta x)^k}{k!} + o(\Delta x)^m \tag{4.9}$$

The first term $u(x_{j+\frac{1}{2}}, t^{n+1})$ is the solution of conventional (piece-wise constant) RP:

$$u_t + f(u)_x = 0, \quad \left. \begin{aligned} u(x, 0) &= \begin{cases} u_L(x_{j+\frac{1}{2}}), & x < x_{j+\frac{1}{2}} \\ u_R(x_{j+\frac{1}{2}}), & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \right\} \tag{4.10}$$

The first term should be calculated by a monotone scheme in order to fade entropy violation. It is well known that the LxF scheme is the best central monotone scheme. Here we use the staggered LxF schemes (4.7–4.8). The terms $u_x^{(k)}(x_{j+\frac{1}{2}}, t^{n+1})$ ($k = 1, 2, \dots, m-1$) are the solutions of the k-th order derivative RPs linearized locally around the first term $u(x_{j+\frac{1}{2}}, t^{n+1}) \equiv u_{j+\frac{1}{2}}^{n+1}$ of (4.9):

$$\begin{aligned}
 &v_t + \lambda(u(x_{j+\frac{1}{2}}, t^{n+1}))v_x = 0, \\
 &v(x, 0) = \begin{cases} u_L^{(k)}(x_{j+\frac{1}{2}}), & x < x_{j+\frac{1}{2}} \\ u_R^{(k)}(x_{j+\frac{1}{2}}), & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \tag{4.11}$$

Here $v = u_x^{(k)}$ are evaluated by the staggered LxF scheme [10]:

$$v_{j+\frac{1}{2}}^{n+1} = \frac{1}{2}[v_L^n(x_{j+\frac{1}{2}}) + v_R^n(x_{j+\frac{1}{2}})] - \lambda \frac{\alpha_k \Delta t}{\Delta x} [v_R^n(x_{j+\frac{1}{2}}) - v_L^n(x_{j+\frac{1}{2}})] \tag{4.12a}$$

Again, in the case of piece-wise constant, the scheme (4.12a) becomes

$$v_{j+\frac{1}{2}}^{n+1} = \frac{1}{2}[v_j^n + v_{j+1}^n] - \lambda \frac{\alpha_k \Delta t}{\Delta x} [v_{j+1}^n - v_j^n] \tag{4.12b}$$

Once we obtain all the spatial derivatives, we calculate the Taylor expansion (4.9) and therefore compute the updated solution $u(x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+1})$.

Now, we evaluate the updated solution u_j^{n+1} as follows:

$$u_j^{n+1} = \begin{cases} u(x_{j-\frac{1}{2}} + \theta^n \Delta x, t^{n+1}), & 0 \leq \theta^n \leq \frac{1}{2} \\ u(x_{j+\frac{1}{2}} + (\theta^n - 1) \Delta x, t^{n+1}), & \frac{1}{2} \leq \theta^n \leq 1 \end{cases} \tag{4.13}$$

4.2 Two steps RCM on staggered grids

In this section, we propose to use the two steps RCM on staggered grids. This method consists of the following steps:

Step 1

Firstly, we solve the local RPs: RP(j - 1, j) and RP(j, j + 1) by using the staggered LxF scheme of (4.1) at time $t^{n+\frac{1}{2}} = (n + \frac{1}{2})\Delta t$

$$u_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[u_L(x_{j-\frac{1}{2}}, t^n) + u_R(x_{j-\frac{1}{2}}, t^n)] - \frac{\alpha_0 \Delta t}{2\Delta x} [f(u_R(x_{j-\frac{1}{2}}, t^n)) - f(u_L(x_{j-\frac{1}{2}}, t^n))] \tag{4.14a}$$

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[u_L(x_{j+\frac{1}{2}}, t^n) + u_R(x_{j+\frac{1}{2}}, t^n)] - \frac{\alpha_0 \Delta t}{2\Delta x} [f(u_R(x_{j+\frac{1}{2}}, t^n)) - f(u_L(x_{j+\frac{1}{2}}, t^n))] \tag{4.14b}$$

For the case of piece-wise constant data $u_L(x_{j-\frac{1}{2}}) = u_{j-1}^n$ and $u_R(x_{j-\frac{1}{2}}) = u_j^n$, these schemes become monotone first order LxF schemes in the staggered form

$$\bar{u}_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[u_{j-1}^n + u_j^n] - \frac{\alpha_0 \Delta t}{2\Delta x}[f(u_j^n) - f(u_{j-1}^n)] \tag{4.15a}$$

$$\bar{u}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[u_j^n + u_{j+1}^n] - \frac{\alpha_0 \Delta t}{2\Delta x}[f(u_{j+1}^n) - f(u_j^n)] \tag{4.15b}$$

Step 2

The updated solution to be determined by the numerical solutions of the local RPs: RP(j - 1, j) and RP(j, j + 1), evaluated at a random position $Q_j = (x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+\frac{1}{2}})$ and computed by Taylor expansion

$$\bar{u}(x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+\frac{1}{2}}) = \bar{u}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}}) + \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial x^k} \bar{u}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}}) \right\} \frac{(\theta^n \Delta x)^k}{k!} + o(\Delta x)^m \tag{4.16}$$

The first term is the solution of conventional RP (4.10) by using the scheme (4.15).

The terms $\bar{u}_x^{(k)}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}})$ ($k = 1, 2, \dots, m-1$) are the solutions of the k-th order derivative RPs linearized locally around the first term $u(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}})$ of (4.16):

$$\left. \begin{aligned} v_t + \lambda(\bar{u}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}}))v_x &= 0, \\ v(x, 0) &= \begin{cases} u_L^{(k)}(x_{j+\frac{1}{2}}), & x < x_{j+\frac{1}{2}} \\ u_R^{(k)}(x_{j+\frac{1}{2}}), & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \right\} \tag{4.17}$$

Here $v = \bar{u}_x^{(k)}$ are computed by the staggered LxF scheme:

$$\bar{v}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[v_L^n(x_{j+\frac{1}{2}}) + v_R^n(x_{j+\frac{1}{2}})] - \lambda \frac{\alpha_k \Delta t}{2\Delta x}[v_R^n(x_{j+\frac{1}{2}}) - v_L^n(x_{j+\frac{1}{2}})] \tag{4.18a}$$

In the case of piece = wise constant, the scheme (4.18a) becomes

$$\bar{v}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[v_j^n + v_{j+1}^n] - \lambda \frac{\alpha_k \Delta t}{2\Delta x}[v_{j+1}^n - v_j^n] \tag{4.18b}$$

Having computed all the derivatives in (4.16), we compute the updated solutions $u_{j-\frac{1}{2}}^{n+\frac{1}{2}}$ and $u_{j+\frac{1}{2}}^{n+\frac{1}{2}}$ as follows:

$$u_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \bar{u}(x_{j-\frac{1}{2}} + (\theta^n - \frac{1}{2})\Delta x, t^{n+\frac{1}{2}}) \quad \text{and} \quad u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \bar{u}(x_{j+\frac{1}{2}} + (\theta^n - \frac{1}{2})\Delta x, t^{n+\frac{1}{2}}) \tag{4.19}$$

Step 3

Solve the RP($j - \frac{1}{2}, j + \frac{1}{2}$) with initial data (4.19) to find u_j^{n+1} and then find Taylor expansion

$$u(x_j + \theta^{n+1} \Delta x, t^{n+1}) = \bar{u}(x_j, t^{n+1}) + \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial x^k} \bar{u}^{(k)}(x_j, t^{n+1}) \right\} \frac{(\theta^{n+1} \Delta x)^k}{k!} + o(\Delta x)^m \tag{4.20}$$

where the first term is the solution of conventional RP($j - \frac{1}{2}, j + \frac{1}{2}$) by using the LxF scheme

$$\bar{u}_j^{n+1} = \frac{1}{2} [u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n+\frac{1}{2}}] - \frac{\alpha_0 \Delta t}{2 \Delta x} [f(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}) - f(u_{j-\frac{1}{2}}^{n+\frac{1}{2}})] \tag{4.21}$$

The terms $\bar{u}_x^{(k)}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}})$ ($k = 1, 2, \dots, m-1$) are the solutions of the k -th order derivative RPs linearized locally around the first term of (4.20):

$$v_t + \lambda(\bar{u}(x_j, t^{n+1}))v_x = 0, \quad \left. \begin{aligned} v(x, 0) &= \begin{cases} u_L^{(k)}(x_j), & x < x_j \\ u_R^{(k)}(x_j), & x > x_j \end{cases} \end{aligned} \right\} \tag{4.22}$$

$v = u_x^{(k)}$ are evaluated by the LxF scheme:

$$\bar{v}_j^{n+\frac{1}{2}} = \frac{1}{2} [v_L^{n+\frac{1}{2}}(x_j) + v_R^{n+\frac{1}{2}}(x_j)] - \lambda \frac{\alpha_k \Delta t}{2 \Delta x} [v_R^{n+\frac{1}{2}}(x_j) - v_L^{n+\frac{1}{2}}(x_j)] \tag{4.23a}$$

In the case of piece-wise constant, this scheme becomes

$$\bar{v}_j^{n+1} = \frac{1}{2} [v_j^{n+\frac{1}{2}} + v_{j+1}^{n+\frac{1}{2}}] - \lambda \frac{\alpha_k \Delta t}{2 \Delta x} [v_{j+1}^{n+\frac{1}{2}} - v_j^{n+\frac{1}{2}}] \tag{4.23b}$$

After computing all the derivatives in (4.20), we sample the updated solutions u_j^{n+1} as follows:

$$u_j^{n+1} = \bar{u}(x_j + (\theta^{n+1} - \frac{1}{2})\Delta x, t^{n+1}) \quad 0 \leq \theta^{n+1} \leq 1 \tag{4.24}$$

4.3 Central RCM with third-order Flux

The main target of this paper is to enhance the accuracy of the CRCM. For this purpose, we use the central third-order TVD flux [22] instead of the first-order flux as the building block of the presented methods. The new method is denoted by the CRCM-r-TVD method. It consists of the following steps:

Step 1

Firstly, we solve the $RP(j - \frac{1}{2}, j + \frac{1}{2})$ to obtain the solution \bar{u}_j^{n+1} as

$$\bar{u}_j^{n+1} = \frac{1}{2}[u_{j-\frac{1}{2}}^n + u_{j+\frac{1}{2}}^{n+\frac{1}{2}}] - \frac{\alpha_0 \Delta t}{\Delta x} [F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n] \tag{4.25}$$

Here we use the third-order TVD flux presented in [22]. For linear Eq. (4.1–4.2)

$$F_{j+1/2} = \frac{1}{2}(\lambda u_j + \lambda u_{j+1}) - \frac{1}{2}|\lambda| \Delta_{j+1/2} u + |\lambda| \left\{ A_0 \Delta_{j+1/2} u + A_1 \Delta_{j+L+1/2} u \right\} \phi_j + |\lambda| A_2 \Delta_{j+M+1/2} u \phi_{j+M} \tag{4.26}$$

where $L = -1, M = 1$ for $c > 0$ and $L = 1, M = -1$ for $c < 0$.

Here $c = \lambda \frac{\Delta t}{\Delta x}$ is the Courant number and $\Delta_{j+1/2} u = u_{j+1} - u_j$,

$$A_0 = \frac{1}{2} - \frac{|c|}{4}, A_1 = -\frac{|c|}{8} - \frac{c^2}{8}, A_2 = -\frac{|c|}{8} + \frac{c^2}{8} \tag{4.27}$$

Here ϕ_j and ϕ_{j+M} are flux limiter functions see [22].

For nonlinear scalar problems $\lambda = \lambda(u)$, we define the wave speed

$$\lambda_{j+\frac{1}{2}} = \begin{cases} \frac{\Delta_{j+\frac{1}{2}} f}{\Delta_{j+\frac{1}{2}} u} & \Delta_{j+\frac{1}{2}} u \neq 0 \\ \frac{\partial f}{\partial u} \Big|_{u_j} & \Delta_{j+\frac{1}{2}} u = 0 \end{cases} \tag{4.28}$$

The numerical flux (4.2) takes the form

$$F_{j+1/2} = \frac{1}{2}(f_j + f_{j+1}) - \frac{1}{2} \left| \lambda_{j+\frac{1}{2}} \right| \Delta_{j+1/2} u + \left| \lambda_{j+\frac{1}{2}} \right| \left\{ A_0 \Delta_{j+1/2} u + A_1 \Delta_{j+L+1/2} u \right\} \phi_j + \left| \lambda_{j+\frac{1}{2}} \right| A_2 \Delta_{j+M+1/2} u \phi_{j+M} \tag{4.29}$$

The coefficients A_j are the same (4.27) with replacing c by $c_{j+\frac{1}{2}} = \frac{\Delta t}{\Delta x} \lambda_{j+\frac{1}{2}}$.

Then we use Taylor expansion

$$u(x_j + \theta^{n+1} \Delta x, t^{n+1}) = \bar{u}(x_j, t^{n+1}) + \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial x^k} \bar{u}^{(k)}(x_j, t^{n+1}) \right\} \frac{(\theta^{n+1} \Delta x)^k}{k!} + o(\Delta x)^m \tag{4.30}$$

where \bar{u}_j^{n+1} is given by (4.25). The terms $\bar{u}_x^{(k)}(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}})$ ($k = 1, 2, \dots, m-1$) are the solutions of the k -th order derivative RPs linearized locally around the first term of (4.30):

$$v_t + \lambda(\bar{u}(x_j, t^{n+1}))v_x = 0, \quad \left. \begin{aligned} v(x, 0) = \begin{cases} u_L^{(k)}(x_j), & x < x_j \\ u_R^{(k)}(x_j), & x > x_j \end{cases} \end{aligned} \right\} \quad (4.31)$$

$v = u_x^{(k)}$ are evaluated by the finite difference scheme:

$$v_j^{n+1} = \frac{1}{2} [v_{j-\frac{1}{2}}^n + v_{j+\frac{1}{2}}^{n+\frac{1}{2}}] - \frac{\alpha_k \Delta t}{\Delta x} [F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n] \quad (4.32)$$

with the third-order TVD (4.25).

After computing all the derivatives in (4.30), we sample the updated solutions u_j^{n+1} as follows:

$$u_j^{n+1} = \bar{u}(x_j + (\theta^{n+1} - \frac{1}{2})\Delta x, t^{n+1}) \quad 0 \leq \theta^{n+1} \leq 1$$

4.4 semi-discrete finite volume scheme

There is another approach for establishing numerical methods, this is the semi-discrete method. In this approach, we consider the discretization in space, while leaving the problem continuous in time. This produces an Ordinary Differential Equation (ODE) in time. This approach gives more flexibility and is well suitable for constructing very high-order schemes.

By integrating (4.1) with respect to x and keeping the time variable continuous, we obtain the semi-discrete finite volume scheme

$$\frac{d}{dt} u_j(t) = -\frac{1}{\Delta x} \left\{ F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}} \right\} \quad (4.33)$$

where $u_j(t)$ and $F_{j+\frac{1}{2}}$ are

$$u_j(t) = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t) dx, \quad F_{j+\frac{1}{2}} = F(u_{j+\frac{1}{2}}(t)) \quad (4.34)$$

Equation (4.33) is a system of time-dependent ODEs that should be solved by using a stable ODE solver that keeps the spatial accuracy of the scheme. In this study, we

use the linear strong-stability-preserving Runge–Kutta algorithm $\ell SSPRK(s,s-1)$ i.e., s -stage and $(s-1)$ -th order method, presented by Gottlieb [5].

For our central RCM, we use the semi-discrete scheme (4.33) instead of fully discrete schemes (4.3).

5 Accuracy

In this section, we discuss the accuracy of the central CRCM based on a semi-discrete scheme on non-staggered grids. We will discuss the accuracy of the assumption of a smooth solution.

Firstly, we reconstruct, at t^n , the cell averages u_j^n by a piece-wise polynomial $P_j(x)$. To fade the spurious oscillations near discontinuities, we use the WENO polynomial reconstruction. For r stencils, we construct the polynomial $P_j(x)$ of degree $(r-1)$ and therefore the spatial accuracy is $(2r-1)$ -th order i.e.,

$$P_j(x) = u(x, t^n) + o(\Delta x)^{2r-1} \tag{5.1}$$

and the spatial accuracy for the k -th order derivative for u is $(r-k)$ -th order

$$\partial_x^{(k)} P_j(x) = u_x^{(k)}(x, t^n) + o(\Delta x)^{r-k} \tag{5.2}$$

Also, the function $g_j(x) = f(P_j(x))$ has the order of accuracy [11, 23]

$$g_j(x) = f(x, t^n) + o(\Delta x)^{2r-1} \tag{5.3a}$$

$$\partial_x^{(k)} g_j(x) = f_x^{(k)}(x, t^n) + o(\Delta x)^{r-k} \tag{5.3b}$$

The numerical solution defined in (4.13) with (4.9) can be written as

$$\begin{aligned} u_j^{n+1} &= u(x_{j+\frac{1}{2}} + \theta^n \Delta x, t^{n+1}) = u(x_{j+\frac{1}{2}}, t^{n+1}) \\ &+ \sum_{k=1}^{m-1} \left\{ \frac{\partial^k}{\partial x^k} u(x_{j+\frac{1}{2}}, t^{n+1}) \right\} \frac{(\theta^n \Delta x)^k}{k!} + o(\Delta x)^m \end{aligned} \tag{5.4}$$

where the first term is the solution of (4.10) using the semi-discrete scheme (4.33) with the truncation error

$$\begin{aligned} T_0(x_j, t^n) &= \frac{du_j}{dt} + \frac{1}{\Delta x} \left\{ F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}} \right\} = [u_t + o(\Delta t)^s] + [f_x + o(\Delta x)^{2r-1}] \\ &= o(\Delta t)^s + o(\Delta x)^{2r-1} \end{aligned} \tag{5.5}$$

Since $u_t + f_x = 0$. Here s is the order of the time discretization method.

The remaining terms of (5.4) are the solutions of (4.11) with (4.33) with the error

$$T_k(x_j, t^n) = o(\Delta t)^s + o(\Delta x)^{r-k} \tag{5.6}$$

Therefore from (5.4–5.6) we get the solution u_j^{n+1} with the total truncation error

$$\begin{aligned} T(x_j, t^{n+1}) &= [o(\Delta t)^s + o(\Delta x)^{2r-1}] + \sum_{k=1}^{m-1} [o(\Delta t)^s + o(\Delta x)^{r-k}] \frac{(\theta \Delta x)^k}{k!} + o(\Delta x)^m \\ &= o(\Delta t)^s + o(\Delta x)^{2r-1} + o(\Delta t)^s \sum_{k=1}^{m-1} o(\Delta x)^k \frac{\theta^k}{k!} + o(\Delta x)^r \cdot \sum_{k=1}^{m-1} \frac{\theta^k}{k!} + o(\Delta x)^m \end{aligned} \tag{5.7}$$

If the ratio $\frac{\Delta t}{\Delta x}$ is constant, we get

$$T(x_j, t^{n+1}) = o(\Delta t)^s + o(\Delta x)^{2r-1} + o(\Delta x)^s \sum_{k=1}^{m-1} o(\Delta x)^k \frac{\theta^k}{k!} + o(\Delta x)^r \sum_{k=1}^{m-1} \frac{\theta^k}{k!} + o(\Delta x)^m$$

Since $\theta \leq 1$ i.e., $\frac{\theta^k}{k!}$ is bounded, therefore

$$|T(x_j, t^{n+1})| \leq o(\Delta t)^s + o(\Delta x)^{2r-1} + o(\Delta x)^s \sum_{k=1}^{m-1} A_1 o(\Delta x)^k + A_2 o(\Delta x)^r + o(\Delta x)^m$$

where A_1 and A_2 are constants.

Therefore

$$|T(x_j, t^n)| \leq C_1(\Delta x)^d + C_2(\Delta t)^s \tag{5.8}$$

where C_1 and C_2 are constants and $d = \min\{s + 1, r, m\}$. Using the WENO reconstruction of order r , the time discretization of order s and Taylor expansion (4.9) of order m , we get the scheme of order s in time and d in space.

For the central RCM on staggered grids and RCM with TVD fluxes, we will see the accuracy empirically.

6 Numerical experiments

In this section, we study, numerically, the efficiency and convergence properties of our schemes presented here. To show the efficiency of the schemes, we compare their numerical results with those of other state-of-the-art high-order shock-capturing schemes, e.g., RCM [16] and central ADER schemes [22] and others. For all schemes we use CFL = 0.45. For all examples, we use a uniform mesh with N as the number of cells. In all graphs, the exact solution is shown by the full line while the numerical solution is shown by symbols.

Here, we compare the following schemes:

1. RCM is the original RCM [16]
2. NONS-CRCM-r is the non-staggered central RCM of the r-th order.
3. STAG-CRCM-r is the staggered central RCM of r-th order.
4. ADER-r is the r-th order central ADER scheme [23] with LxF flux.
5. ADER-r-TVD is the r-th order central ADER scheme [23] with third-order TVD flux [23].
6. CRCM-r- TVD is the CRCM of order r with third-order TVD flux [22].

We use the *ℓSSPRK*(s, s-1), s = 6 method for time integration for the linear problems while for the nonlinear problems we replace the linear SSPRK schemes with SSPRK(5,4) method. [5]

6.1 Example 1 (Convergence tests)

Firstly, we test the accuracy of the new schemes on the linear scalar problems.

$$u_t + u_x = 0, \quad x \in [-1, 1] \tag{6.1}$$

with periodic initial conditions

$$u(x, 0) = \sin\left(\pi x - \frac{\sin \pi x}{\pi}\right) \tag{6.2}$$

defined on $[-1, 1]$. Table 1 presents the convergence rates and errors in L^1 norm at time $t = 20$. We notice, from the table, that ADER5, NONS-CRCM-5, STAG-CRCM-5, ADER-5-TVD, and CRCM-5-TVD methods reach the designed order of accuracy. Also, we observe that the central RCM schemes, presented here, are more accurate in both error sizes and order of accuracy while ADER-5-TVD and CRCM-5-TVD are more accurate (approximately 6.5). Moreover, CRCM-5-TVD is comparable with ADER-5-TVD. From Table 1 we have the following notes:

1- the CPU time used by the NONS-CRCM-5 scheme is about 66% of that by the ADER5 scheme on a given mesh. While the CPU time used by the Stag-CRCM-5 scheme is about 75% of that by the ADER5 scheme on a given mesh. This is due to a less flux evaluation in CRCM than in the ADER methods.

2- As expected, ADER-5-TVD and CRCM-5-TVD are the most accurate in both error sizes and order of accuracy. This is due to utilizing high-order flux (third-order TVD) instead of first-order flux. However, the CPU time used by CRCM-5-TVD is about 35% of ADER-5-TVD on given meshes. Moreover, the CPU time used by CRCM5-TVD is about 52% of that by NONS-CRCM-5 and is about 57% of STAG-CRCM. This is due to the evaluations of the updated solution of NONS.CRCM-5 and STAG-CRCM methods are computed by using the information of the adjacent two RPs, while for CRCM-5-TVD, we use only one RP. Figure 3 shows the comparison of the efficiency of L^1 errors versus CPU time. We observe that the scheme with the least CPU time to achieve the same error would have the best efficiency, therefore CRCM-5-TVD is the most efficient. It is beneficial to see how the numerical solution

Table 1 Convergence study for example 1

Method	Mesh	L^1 error	L^1 order	CPU time
ADER-5-TVD	10	2.17E-6		3.67E-2
	20	2.38E-8	6.51	1.27E-1
	40	2.45E-10	6.60	5.13E-1
	80	2.36E-12	6.69	1.91E+0
	160	2.53E-14	6.54	7.61E+0
ADER-5	10	8.01E-5		2.91E-2
	20	2.43E-6	5.04	1.08E-1
	40	7.51E-8	5.02	4.01E-1
	80	2.42E-9	4.95	1.53E+0
	160	7.11E-11	5.09	5.75E+0
NONS-CRCM-5	10	6.13E-5		1.92E-2
	20	1.53E-6	5.32	7.34E-2
	40	3.82E-8	5.33	2.80E-1
	80	9.74E-10	5.30	1.06E+0
	160	2.44E-11	5.32	4.05E+0
STAG-CRCM-5	10	7.21E-5		2.16E-2
	20	1.97E-6	5.19	6.24E-2
	40	5.31E-8	5.21	3.12E-1
	80	1.43E-9	5.21	1.19E+0
	160	3.83E-11	5.22	4.54E+0
CRCM-5-TVD	10	2.37E-6		1.23E-2
	20	2.46E-8	6.59	4.76E-2
	40	2.63E-10	6.55	1.81E-1
	80	2.57E-12	6.68	6.89E-1
	160	3.11E-14	6.37	2.65E+0

depends on the number of grid points N . For this target the errors of the numerical solutions in L^1 norms are shown in Fig. 4 as a function of the number of grid points N . It is clear that NONS-CRCM-5, STAG-CRCM-5, and ADER-5 schemes achieve the fifth-order accuracy even on coarse grids. The ADER-5-TVD and CRCM-5-TVD schemes achieve approximately seventh order. For reference targets, we plot the lines with N^{-5} and N^{-7} slopes, which are consistent with the formal order of accuracy of the schemes.

6.2 Example 2. (Example with discontinuities)

We consider the linear Eq. (6.1) with the following initial data

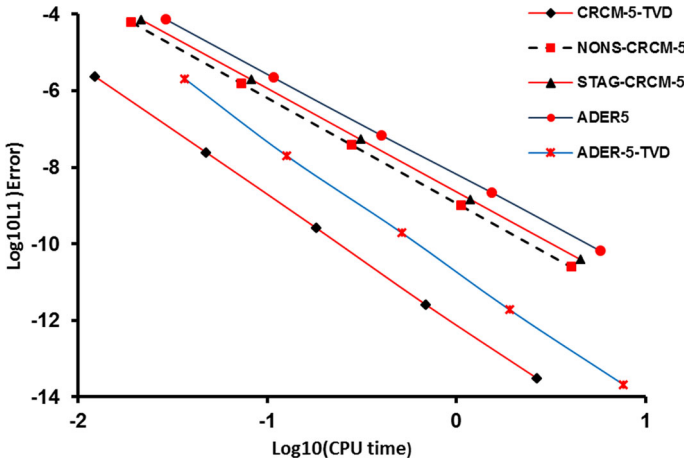


Fig. 3 CPU time and L^1 error curve for Example 1

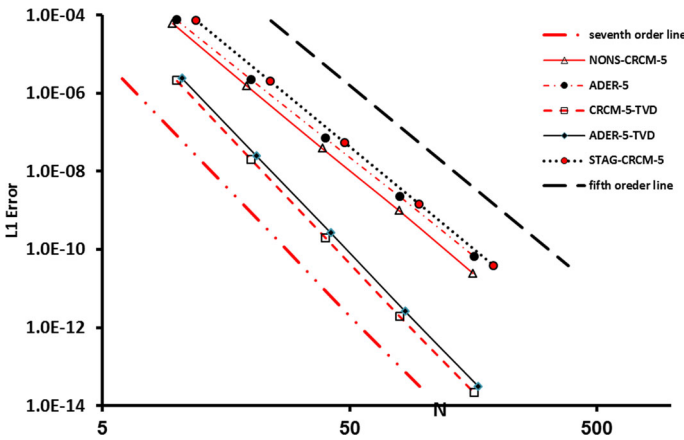


Fig. 4 Rate of convergence in terms of L^1 error for example 1

$$u(x, 0) = \begin{cases} \frac{1}{6}[G(x, z - \delta) + G(x, z + \delta) + 4G(x, z)], & -0.8 \leq x \leq -0.6 \\ 1, & -0.4 \leq x \leq -0.2 \\ 1 - |10(x - 0.1)| & 0 \leq x \leq 0.2 \\ \frac{1}{6}[F(x, a - \delta) + F(x, a + \delta) + 4F(x, a)], & 0.4 \leq x \leq 0.6 \\ 0, & \text{otherwise} \end{cases} \quad (6.3)$$

Here $G(x, z) = \exp(-\beta(x - z)^2)$, $F(x, a) = \{\max(1 - \alpha^2(x - a)^2, 0)\}^{1/2}$ and periodic boundary conditions on $[-1, 1]$. Where the constants $a = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$ and $\beta = (\log 2)/36\delta^2$. It is clear that the initial condition consists of different shapes which are difficult for numerical schemes to resolve correctly. Some of them are not smooth and the others are smooth but very sharp. We compute the

solution to this problem at time $t = 20$ with 200 cells. Figures 5, 6, 7 and 8 show the results of RCM, NONS-CRCM-5, Stag-CRCM-5 and CRCM-5-TVD schemes respectively. Comparing these results with the ADER5 and ADER5-TVD schemes (Figures 3–4 in [23]), we observe that the RCM produces sharp discontinuities while the accuracy in the smooth regions is poor, the numerical solution obtained by the ADER5 is quite satisfied. NONS-CRCM-5 and STAG-CRCM-5 schemes are better results while the numerical solutions obtained by CRCM-5-TVD and ADER-5-TVD schemes are almost indistinguishable from the exact solution. Comparing the results here with the sixth-order WENO schemes presented in [7, 8, 20] (figure 5 in [8] with $N = 400$, figure 4 in [20] with $N = 400$, and figure 6 in [7] with $N = 200$) we observed that our schemes are comparable to them but they are much better in capturing the shocks and approximating top of the semi-ellipse. Also, we notice that the CRCM-5-TVD is superior. In Table 2, the errors and the CPU time are presented. It is clear that the

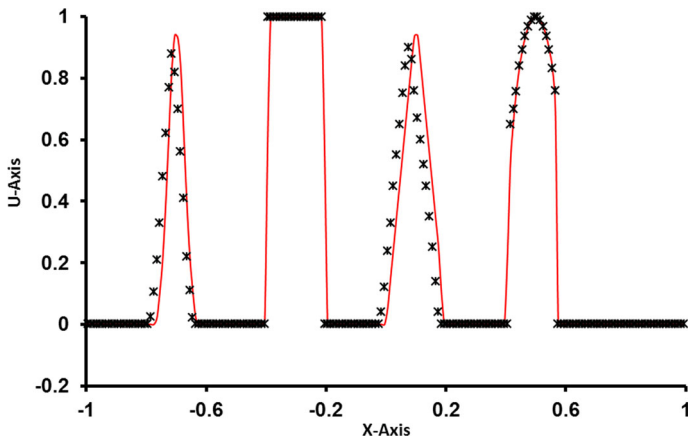


Fig. 5 Solution of Example 2 using RCM scheme at $t = 20$

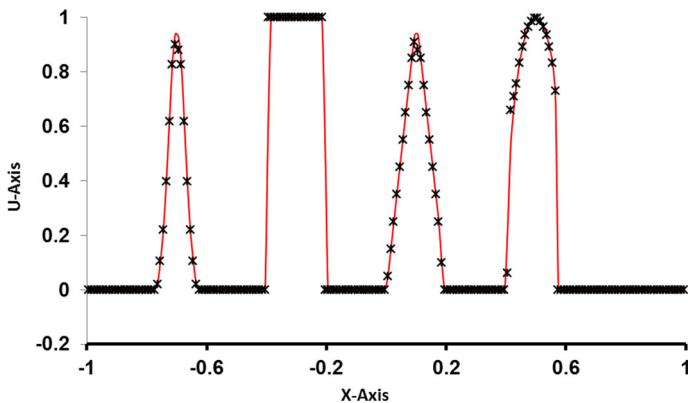


Fig. 6 Solution of Example 2 using NONS-CRCM-5 scheme at $t = 20$

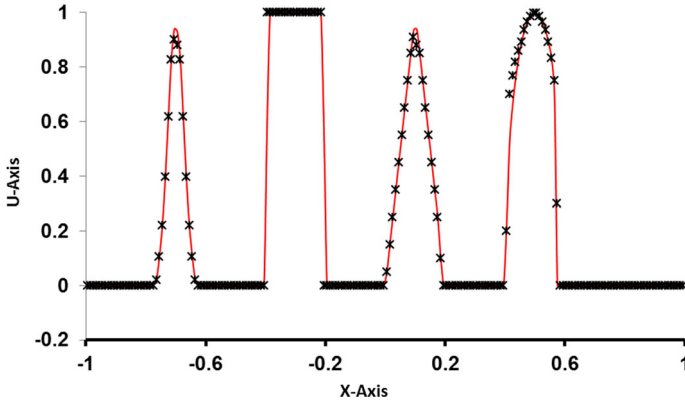


Fig. 7 Solution of Example 2 using Stag-CRCM-5 scheme at $t = 20$

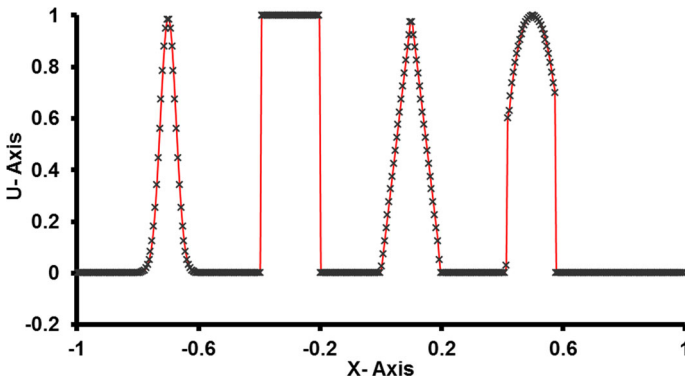
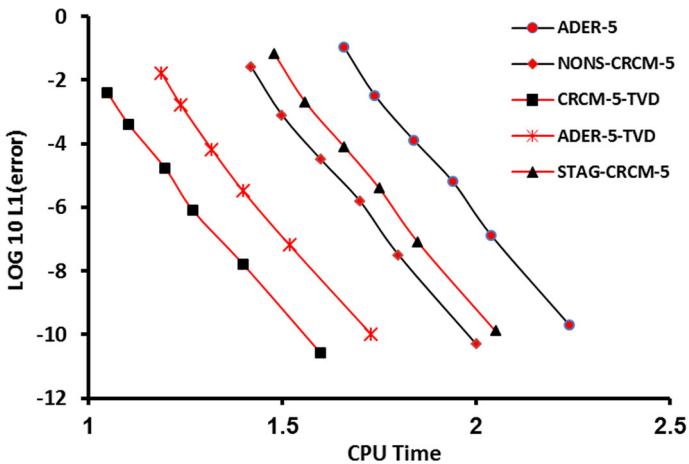


Fig. 8 Solution of Example 2 using CRCM-5-TVD scheme at $t = 20$

CPU time used by the NONS-CRCM-5 scheme is about 68% of that by the ADER5 scheme on a given mesh. While the CPU time used by the Stag-CRCM-5 scheme is about 75% of that by the ADER5 scheme on a given mesh. However, the CPU time used by CRCM-5-TVD is about 39% of ADER-5-TVD on a given mesh. Moreover, the CPU time used by CRCM5-TVD is about 75% of that by NONS-CRCM-5 and is about 69% of STAG-CRCM. Figure 9 shows the comparison of the efficiency of L^1 errors versus CPU time. We note that the CRCM-5-TVD scheme is the most efficient. Figure 10 shows the distribution of the L^1 norm of the solution error as a function of N . It is clear that all schemes achieve first-order accuracy, since the convergence rate is reduced to N^{-1} approximately, which is to be expected when discontinuities are involved.

Table 2 Convergence study for example 2

Method	Mesh	L^1 error	L^1 order	CPU time
ADER-5-TVD	200	3.34E-4		9.25E+0
	400	1.45E-4	1.20	3.49E+1
	800	6.26E-5	1.21	1.33E+2
	1600	2.72E-5	1.20	5.05E+2
ADER5	200	7.68E-3		7.04E+0
	400	3.52E-3	1.13	2.72E+1
	800	1.62E-3	1.12	1.06E+2
	1600	7.53E-4	1.11	4.11E+2
NONS-CRCM-5	200	2.68E-3		4.81E+0
	400	1.19E-3	1.17	1.82E+1
	800	5.36E-4	1.15	7.05E+1
	1600	2.37E-4	1.17	2.66E+2
Stag-CRCM-5	200	2.77E-3		5.26E+0
	400	1.26E-3	1.14	2.02E+1
	800	5.81E-4	1.12	7.64E+1
	1600	2.62E-4	1.15	2.91E+2
CRCM-5-TVD	200	3.94E-4		3.63E+0
	400	1.79E-4	1.16	1.36E+1
	800	7.86E-5	1.19	5.17E+1
	1600	3.61E-5	1.12	1.93E+2

**Fig. 9** CPU time and L^1 error curve for Example 2

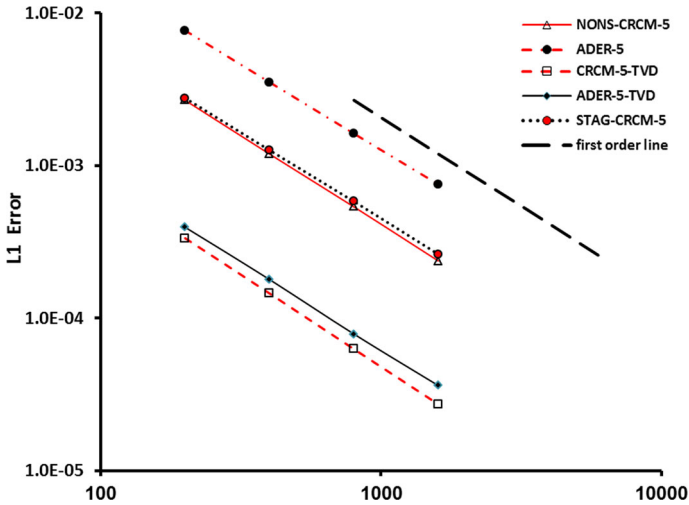


Fig. 10 Rate of convergence in terms of L^1 error for example 2

6.3 Example 3 (Burgers' equation)

It is well known that the Burger problem has simple acoustic waves and therefore allows shocks. It has the form

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \tag{6.4}$$

with initial condition

$$u(x, 0) = \begin{cases} -1 & |x| \geq 0.5 \\ 2 & |x| < 0.5 \end{cases} \tag{6.5}$$

The breakdown of the initial discontinuity gives a shock wave with a speed of 0.5 and a simple centered expansion fan with a sonic point. At $t = 2/3$ the rarefaction hits the shock and therefore the solution has a rarefaction wave only. Figures 11, 12, 13 and 14 show the results at $t = 0.4$ (before the collision of the head of the rarefaction with the shock) and $t = 1$ (after collision), with 80 grid points, obtained by RCM, NONS-CRCM-5, STAG-CRCM-5, and CRCM-5-TVD schemes respectively. We notice from the figures that the RCM produces sharp discontinuities while the accuracy in the smooth regions is poor. The numerical solution obtained by the ADER5 is quite satisfied. NONS-CRCM-5 and STAG-CRCM-5 schemes are better results while and the numerical solutions obtained by CRCM-5-TVD and ADER-5-TVD schemes are almost indistinguishable from the exact solution.

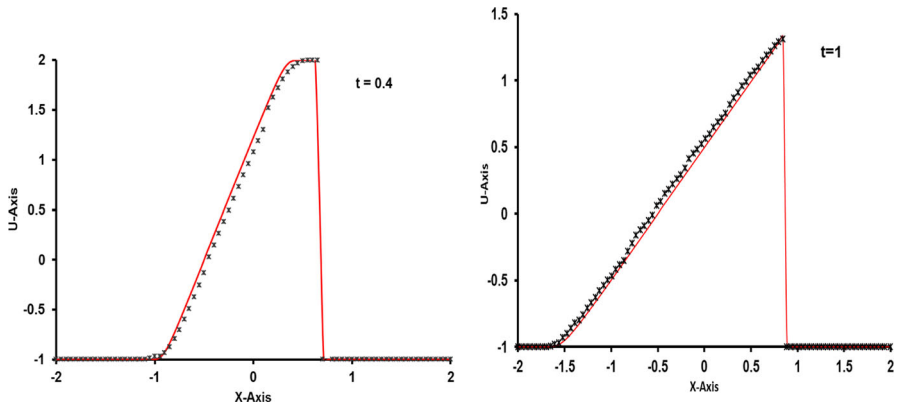


Fig. 11 Solution of Example 3 using RCM scheme

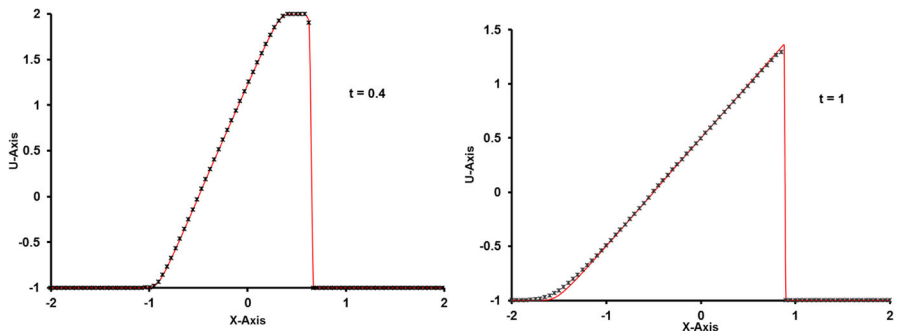


Fig. 12 Solution of Example 3 using NONS-CRCM-5 scheme

6.4 Example 4 (Buckley-Leverett Equation)

To show the performance of our schemes, we solve the Buckley-Leverett problem with non-convex fluxes

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2} \right)_x = 0, \tag{6.6}$$

with initial condition

$$u(x, 0) = \begin{cases} 1 & -0.5 \leq x \leq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{6.7}$$

The exact solution to this problem is a shock-rarefaction-contact discontinuity mixture. It is observed that, for this problem, some high-order schemes fail to converge to the correct entropy solution. Figures 15, 16, 17 and 18 show the numerical results at $t = 0.4$ obtained by RCM, NONS-CRCM-5, Stag-CRCM-5 and CRCM-5TVD

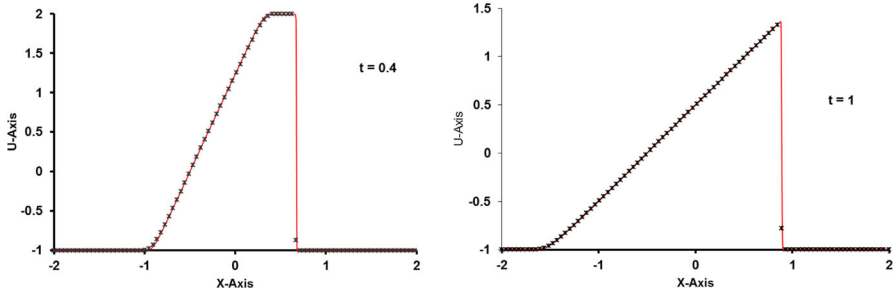


Fig. 13 Solution of Example 3 using STAG-CRCM-5 scheme

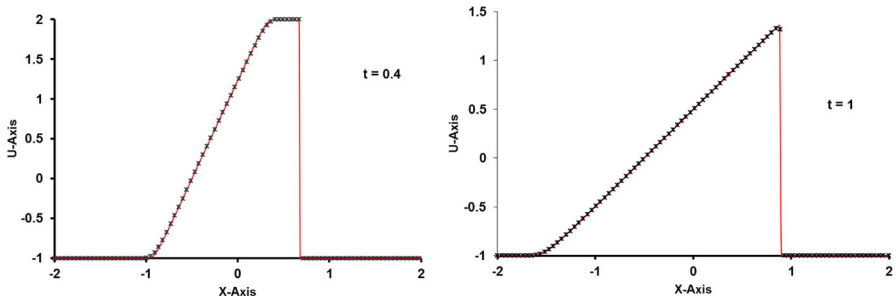


Fig. 14 Solution of Example 3 using CRCM-5-TVD scheme

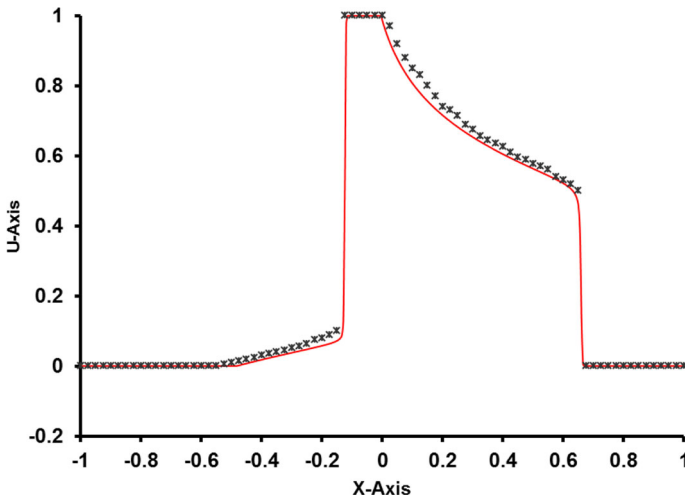


Fig. 15 Solution of Example 4 using RCM

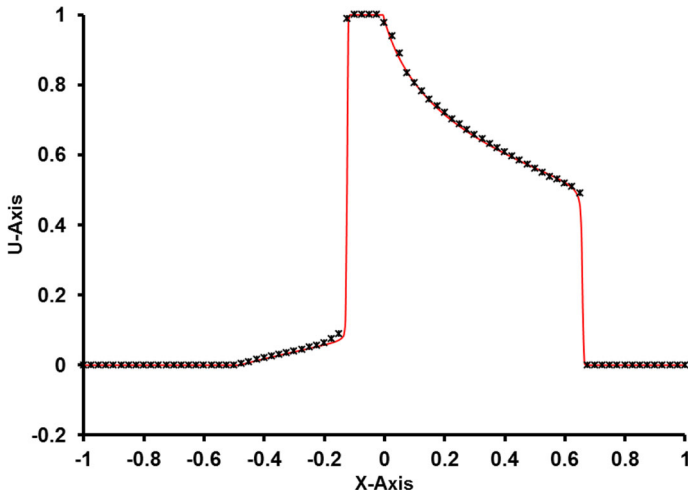


Fig. 16 Solution of Example 4 using NONS-CRCM-5 scheme

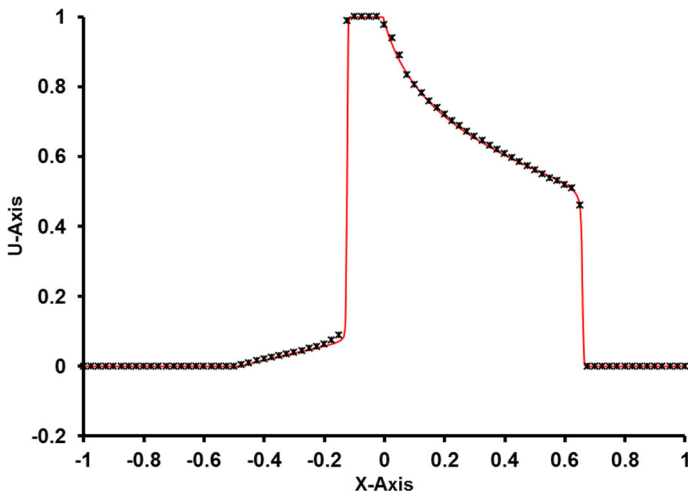


Fig. 17 Solution of Example 4 using STAG-CRCM-5 scheme

schemes respectively with 80 cells. We notice that NONS-CRCM-5 and Stag-CRCM-5 schemes capture the correct entropy solution well, with good resolutions for all the major features in the solution. While the RCM scheme produces the smooth regions of the solution with poor accuracy. Comparing these figures with the solution obtained by ADER5 and ADER-5-TVD (Figures. 13–14 in [23]), we observe that our schemes are more accurate.

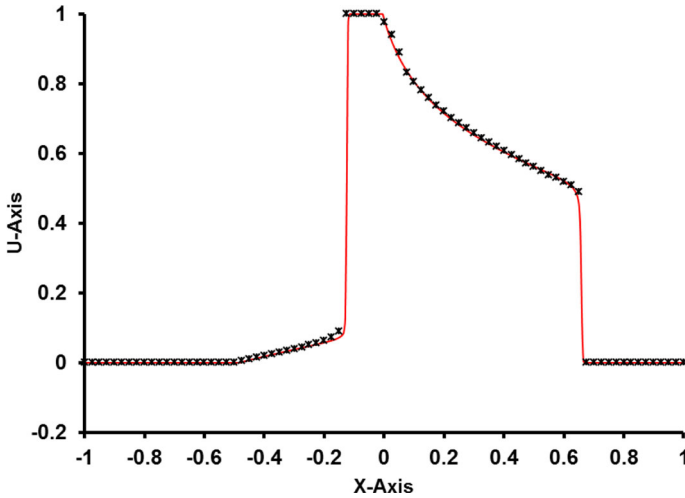


Fig. 18 Solution of Example 4 using CRCM-5-TVD scheme

6.5 Systems of conservation laws

Here we extend our schemes to solve the system of Euler equations of gas dynamics as hyperbolic systems of conservation laws of the form

$$U_t + F(U)_x = 0, \tag{6.8}$$

where $U = (\rho, \rho u, E)^T$, $F(U) = (\rho u, \rho u^2 + P, u(E + P))^T$, where ρ , u , P , E denote density, velocity, pressure, and the total energy $E = \frac{1}{2}\rho u^2 + \frac{P}{(\gamma-1)}$ respectively. The parameter γ denotes the ratio of specific heats, taken as 1.4 here.

6.5.1 Example 5. Lax’s Problem

This problem is the Eq. (6.8) with the initial condition consisting of two states, left (L) and right (R)

$$(\rho_L, u_L, E_L) = (0.445, 0.698, 8.928) \text{ and } (\rho_R, u_R, E_R) = (0.5, 0.0, 0.571) \tag{6.9}$$

separated by a discontinuity at $x = 0.5$. The computational domain is taken as $[0, 1]$. Figures 19, 20, 21 and 22 show the results obtained by RCM, NONS-CRCM-5, STAG-CRCM-5 and CRCM-5TVD schemes respectively at $t = 0.16$ with 100 cells. Comparing these results in the figures and Fig. (16–17) in [22] and figure 9 in [8], we note that our schemes are still better than the other schemes.

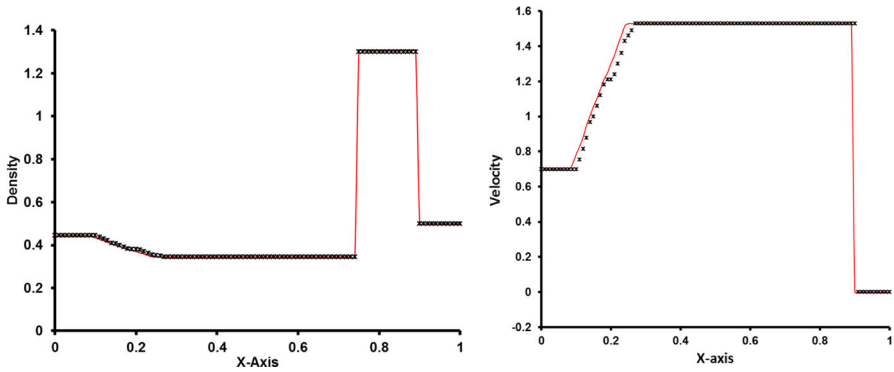


Fig. 19 Solution of Example 5 using RCM

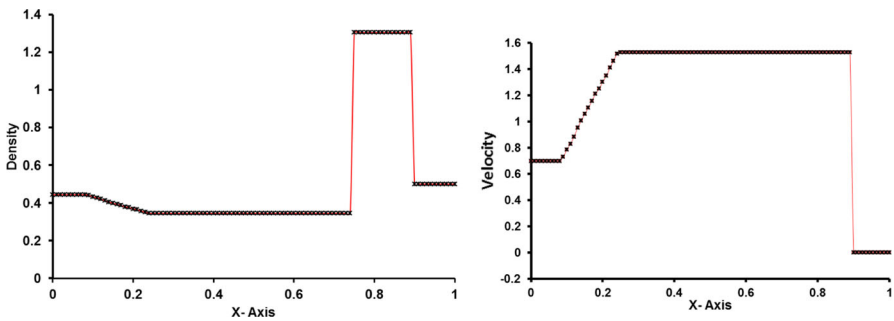


Fig. 20 Solution of Example 5 using the NONS-CRCM-5 scheme

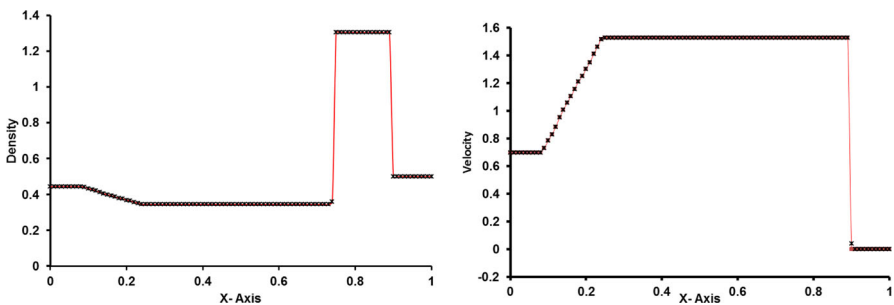


Fig. 21 Solution of Example 5 using STAG-CRCM-5 scheme

6.5.2 Example 6. Blast wave problem

In order to examine the robustness of the numerical schemes presented here, we solve the blast problem introduced by Woodward and Colella [21] which is a severe test problem and is very difficult to be solved numerically. Its solution contains the collision

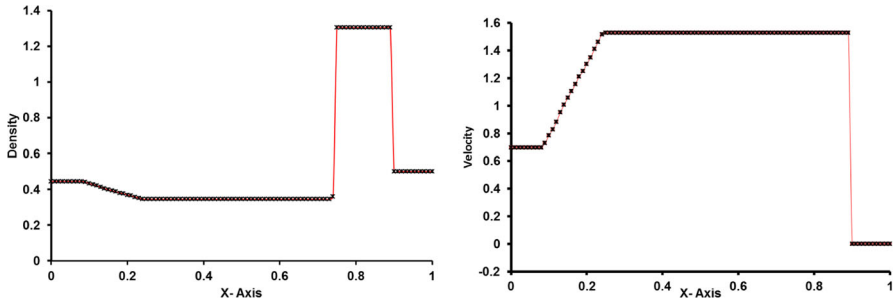


Fig. 22 Solution of Example 5 using CRCM-5-TVD scheme

of strong shock waves and interaction of shock waves and rarefactions, the propagation of strong shock waves into low-pressure regions, and is, therefore, a good test of the schemes. The initial condition consists of three states [21]

$$U(x, 0) = \begin{cases} (\rho_L, u_L, P_L) = (1, 0, 1000), & x < 0.1 \\ (\rho_M, u_M, P_M) = (1, 0, 0.01), & 0.1 < x < 0.9 \\ (\rho_R, u_R, P_R) = (1, 0, 100), & x > 0.9 \end{cases} \quad (6.10)$$

with $\gamma = 1.4$. The boundary conditions are reflective at both boundaries. We present the numerical results of the density and velocity of this complex problem in Fig. 23 obtained by the CRCM-5-TVD scheme with 200 cells at time $t = 0.028$ and $t = 0.038$.

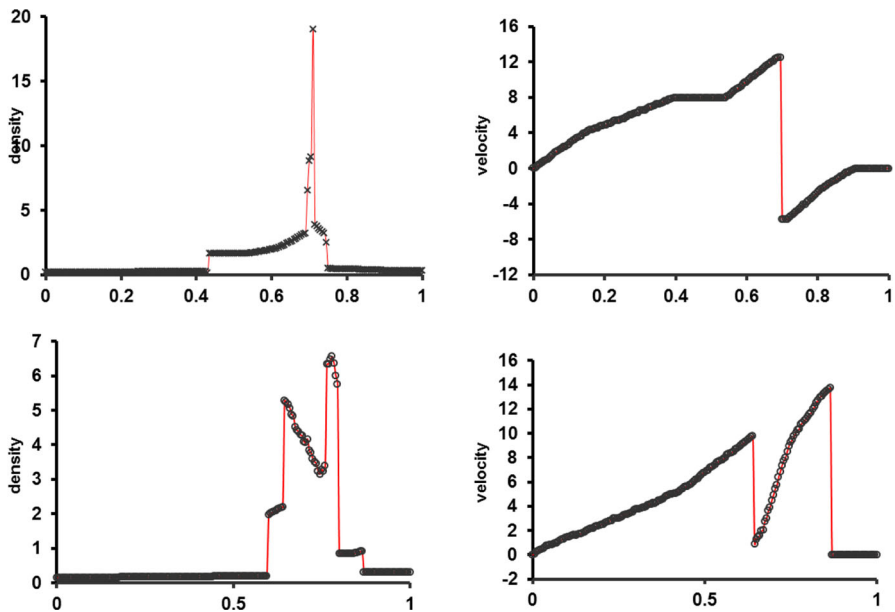


Fig. 23 Solution of Example 6 using NONS-CRCM-5 scheme at $t = .028$ (up) and $t = .038$ (down)

The exact solution is computed by the fifth-order WENO [1] scheme with 4000 cells. It is clear that our scheme can compute such sharp resolution of the complex double-blast problem, especially, since the density peaks have almost the correct value. Comparing the figures here with the results in [7, 8, 20], we conclude that our schemes are more accurate and more efficient.

Note: For the results obtained by NONS-CRCM-5 and STAG-CRCM-5 (not presented here), they are similar to Fig. 23.

6.5.3 Example 7. Shock/turbulence interaction problem

In order to show the advantages of the schemes presented here, we solve an example with a rich smooth structure and a shock wave. This example is the problem of shock interaction with entropy waves.

We take Eqs. (6.8) with a moving Mach = 3 shock interacting with sine waves in density; i.e., the initial condition is

$$\begin{aligned}(\rho_L, u_L, P_L) &= (3.857143, 2.629369, 10.3333), & \text{for } x \leq 0.1 \\(\rho_R, u_R, P_R) &= (1 + 0.2 \sin 50x, 0, 1), & \text{for } x > 0.1\end{aligned}\quad (6.11)$$

The solution contains physical oscillations which should be resolved by the numerical method. We compute the solution at $t = 0.18$ with 200 grid points. Figures 24, 25 and 26 show the computed density by NONS-CRCM-5 STAG-CRCM-5 and CRCM-5TVD schemes against the reference solution, which is computed by the fifth-order WENO scheme [1] with 4000 grid points. Comparing the results with ADER 5 and ADER-5-TVD schemes (figure 19–20 in [23]) and the results in [7, 8, 20], we observe that all the schemes presented here produce the most accurate solution and are more efficient.

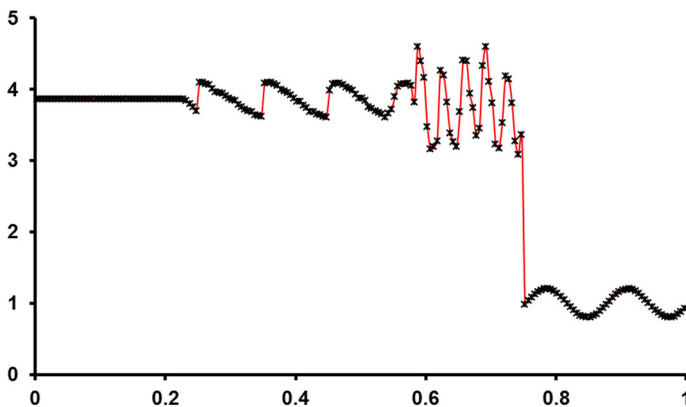


Figure 24 Solution of Example 7 using the NONS-CRCM-5 scheme

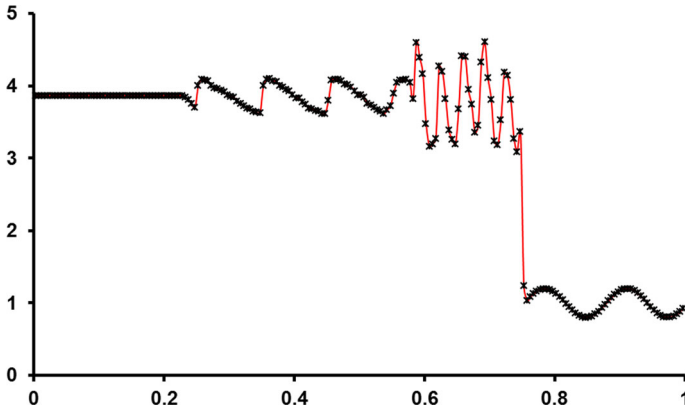


Fig. 25 Solution of Example 7 using STAG-CRCM-5 scheme

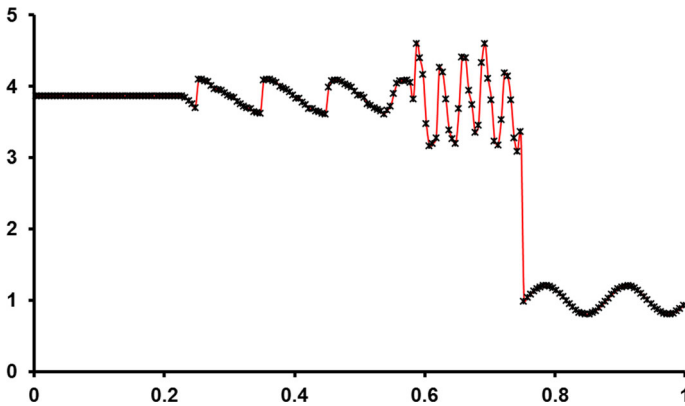


Fig. 26 Solution of Example 7 using CRCM-5-TVD scheme

6.6 Extension to two-dimensional problems

The schemes presented here can be applied to two-dimensional problems by using the space operator splitting technique. For example, we consider the two dimensional, Euler equations

$$U_t + [F(U)]_x + [G(U)]_y = 0 \tag{6.12}$$

where $U = (\rho, \rho u, \rho v, E)^T$, $F(U) = (\rho u, P + \rho u^2, \rho uv, u(P + E))^T$, $G(U) = (\rho v, \rho uv, P + \rho v^2, v(P + E))^T$, $P = (\gamma - 1)[e - 0.5\rho(u^2 + v^2)]$, $S = \frac{P}{\rho\gamma}$, $\gamma = 1.4$. here ρ, P, S, e, u, v are density, pressure, entropy, the specific total energy of the fluid, x-component, and y-component of the velocity respectively. There are several forms of space splitting. We take here the simplest one, where the two-dimensional problem

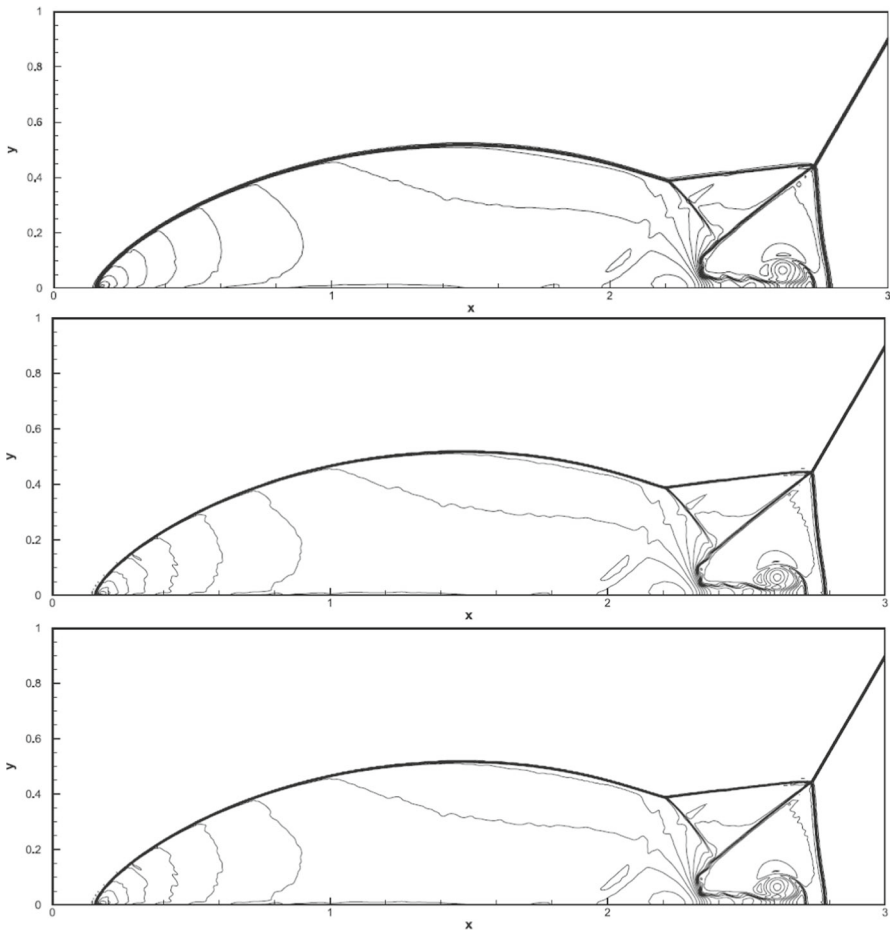


Fig. 27 Double Mach reflection problem (Example 8). 30 density contours are equally spaced from 1.5 to 22.7 on a system size 960×240 for Stag-CRCM-5 (top), NONS-CRCM-5 (middle) and CRCM-5-TVD (bottom)

(6.12) is replaced by the sequence of two one-dimensional problems [16, 19]

$$U_t + [F(U)]_x = 0 \tag{6.13a}$$

$$U_t + [G(U)]_y = 0 \tag{6.13b}$$

If U^n is the data at time level n for the problem (6.12) are given, then the solution U^{n+1} at the time level $n + 1$ is obtained in the following two steps:

- (a) solve the Eq. (6.13a) with data U^n to obtain an intermediate solution \bar{U}^{n+1} (x-sweep);

Table 3 CPU time (in seconds) for the double Mach reflection problem (Example 8)

Method	480 × 120	960 × 240	1920 × 480
Stag-CRCM-5	1821	14,270	111,827
NONS-CRCM-5	1637	12,863	100,315
CRCM-5-TVD	1137	9187	71,682

(b) solve the Eq. (6.13b) with data \bar{U}^{n+1} to obtain the complete solution U^{n+1} (y-sweep);

For three-dimensional problems, we add an extra z-sweep.

6.6.1 Example 8. Double Mach reflection problem

This problem is the two-dimensional Eqs. (6.12) on the domain is $[0, 4] \times [0, 1]$. The reflecting wall is starting from $x = \frac{1}{6}$ at the bottom of the domain. A right-moving Mach 10 shock is, initially, positioned at $(x, y) = (\frac{1}{6}, 0)$ and forms 60° an angle with the x-axis. The exact post-shock condition is assumed from $x = 0$ to $x = \frac{1}{6}$, for the bottom boundary, and a reflective boundary condition is used for the rest of the x-axis. The data is put to describe the exact motion of the Mach 10 shock, at the top boundary of the computational domain [21]. We take the output time as $t = 0.2$. Figure 27 shows the density obtained by NONS-CRCM-5 STAG-CRCM-5 and CRCM-5TVD schemes on the 960×240 cells. Comparing these results with those in the existing literature e.g., [7, 21] it is noticed that our schemes produce the flow pattern generally accepted at present as corrected. All discontinuities are correctly positioned and well resolved. In Table 3, we show the CPU time for the schemes presented here. We observe that the CRCM-5TVD is more efficient.

6.6.2 Example 9. Two-dimensional Vortex evolution problem

Here we examine the accuracy of our methods in the two dimensions. We solve the Eq. (6.12) with the initial conditions, corresponding to a smooth vortex, moving at 45° to the Cartesian mesh lines in the domain $[-5, 5] \times [-5, 5]$. We use periodic boundary conditions. The temperature and entropy are given as $T = \frac{P}{\rho}$ and $S = \frac{P}{\rho^\gamma}$. The vortex is defined as the following isentropic perturbation to the uniform flow of unit values of primitive variables $(\rho, P, u, v) = (1, 1, 1, 1)$:

$$u = 1 - \frac{\epsilon}{2\pi} e^{\frac{1}{2}(1-r^2)} y, \quad v = 1 + \frac{\epsilon}{2\pi} e^{\frac{1}{2}(1-r^2)} x,$$

and the temperature

$$T = 1 - \frac{(\gamma - 1) \epsilon^2}{8\gamma\pi^2} e^{(1-r^2)}, \quad \frac{P}{\rho^\gamma} = 1$$

Table 4 Convergence study for Example 9

Mesh	ADER-5		Stag-CRCM-5		NONS-CRCM-5		CRCM-5-TVD		ADER-5-TVD	
	L ¹ error	L ¹ order	L ¹ error	L ¹ order	L ¹ error	L ¹ order	L ¹ error	L ¹ order	L ¹ error	L ¹ order
25 × 25	7.15E-4		2.46E-6		4.19E-6		3.17E-8		3.03E-8	
50 × 50	3.81E-5	4.23	7.59E-8	5.02	1.27E-7	5.05	3.31E-10	6.58	3.12E-10	6.60
100 × 100	2.19E-6	4.12	2.43E-9	4.96	4.01E-9	4.98	3.51E-12	6.56	2.73E-12	6.83
200 × 200	1.47E-7	3.90	7.36E-11	5.05	1.02E-10	5.07	3.69E-14	6.57	2.75E-14	6.63

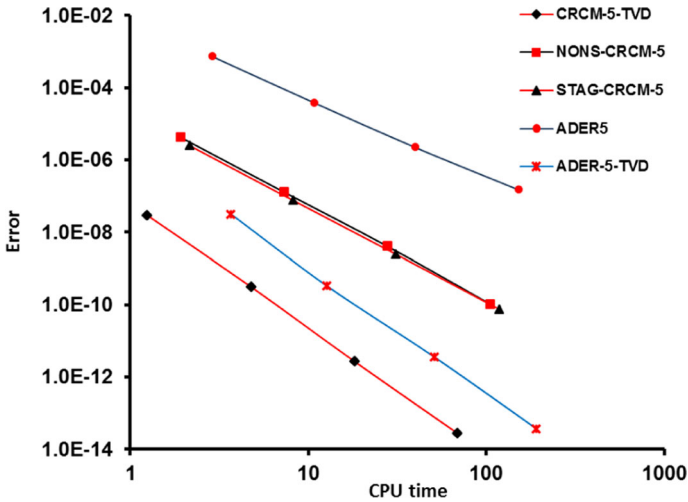


Fig. 28 CPU time and L^1 error curve for Example 9

where $r^2 = x^2 + y^2$ and the vortex strength is $\epsilon = 5$. The numerical solution is computed at the time $t = 10$ when the vortex returns to the initial position. We used periodic boundary conditions in both directions. We observe that, although the flow is smooth over the whole domain, there are critical points at the center of the vortex and symmetric planes. Table 4 shows the errors and convergence for the NONS-CRCM-5, Stag-CRCM-5, ADER-5, and RCM-5-TVD schemes. We observe that NONS-CRCM-5 and Stag-CRCM-5 schemes achieve the theoretical accuracy while the ADER-5 scheme achieves approximately a fourth-order convergence rate. We note here that the CRCM-5-TVD is superior. Figure 28 shows the comparison of the efficiency of L^1 errors versus CPU time. We observe that the scheme with the least CPU time to achieve the same error would have the best efficiency, therefore CRCM-5-TVD is the most efficient.

6.6.3 Example 10 Forward-facing step problem [21]

The problem is as follows: the wind tunnel is of length wide unit and three length units long. The step is 0.2 length units high and is located at 0.6 length units from the left-hand end of the tunnel. The problem is started by a right-going Mach 3 flow. Along the wall of the tunnel, the reflective boundary conditions are applied. At the entrance and the exit, in and outflow boundary conditions are applied. The corner of the step is a singular point and we treat it the same way as in [21], which is based on the assumption of a nearly steady flow in the near the corner. We compute the solution at $t = 4$. We show in Fig. 29 the numerical results on mesh with 480×160 uniform cells by the NONS-CRCM-5 STAG-CRCM-5 and CRCM-5TVD schemes. The results are shown in Fig. 29 at $t = 4$. The figure shows 30 equally spaced density contours from 0.32 to 6.15. We can see that all the schemes capture the vortex sheet roll-up with better resolution in comparison to the other schemes.

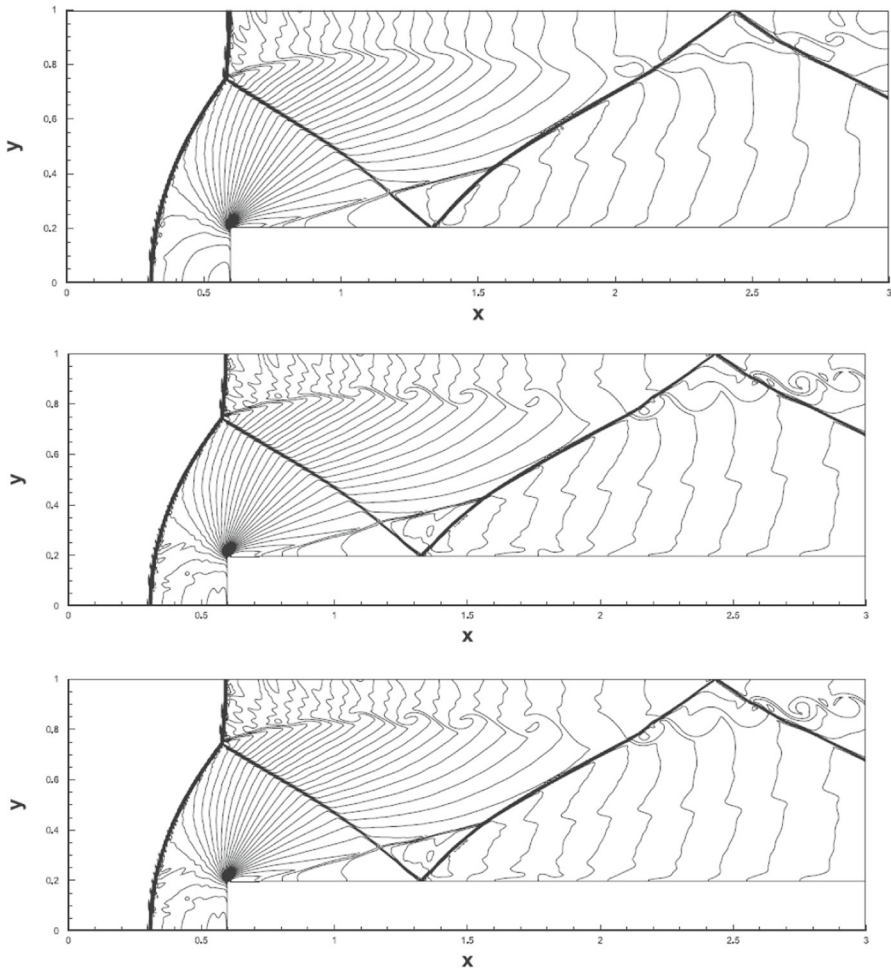


Fig. 29 Forward step problem (Example 10). 30 density contours are equally spaced from 0.32 to 6.15 on a system size 480×160 for Stag-CRCM-5 (top), NONS-CRCM-5 (middle), and CRCM-5-TVD (bottom)

7 Conclusions

In this paper, we presented an improved version of the RCM. We merge the central ADER technique with the RCM for solving hyperbolic conservation laws. The resulting scheme is called Central RCM (CRCM). We use the WENO reconstruction for the initial data instead of constant piece-wise reconstruction in RCM. Also, we use the high order central finite difference schemes to solve the RPs numerically and then use random sampling to update the solution, as the original RCM. Here we use the staggered and non-staggered RCM. To enhance the accuracy of the new methods, we use a third-order TVD flux [22], instead of a first-order flux. The resulting methods have the advantages of the following methods: original RCM, ADER, and central

finite difference methods. They capture the discontinuities with infinite resolution, are more accurate, faster, simple to implement, RP solver free, use less computer memory, and improve the accuracy in smooth regions. An extension of the schemes to general systems of nonlinear hyperbolic conservation laws in one and two dimensions is presented. Many numerical examples are presented for one and two-dimensional problems to confirm that the new schemes are superior to the other high-order schemes. In the future, we plan to use the present schemes for balance laws, multidimensional computations, and unstructured meshes.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Balsara, D.S., Shu, C.-W.: Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.* **160**, 405–452 (2000)
2. Chorin, A.J.: Random choice solution of hyperbolic systems. *J. Comput. Phys.* **22**, 517–536 (1976)
3. Glaister, P.: An approximate linearized Riemann solver for Euler equations for real gases. *J. Comput. Phys.* **74**, 382–408 (1988)
4. Glimm, J.: Solution in the large for nonlinear hyperbolic systems of equations. *Commun. Pure Appl. Math.* **18**, 697–715 (1965)
5. Gottlieb, S.: On high order strong stability preserving Runge-Kutta and multistep time discretizations. *J. Sci. Comput.* **25**(1), 105–128 (2005)
6. Harten, A.: High-resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49**, 357–393 (1983)
7. Hu, F.: The 6th-order weighted ENO schemes for hyperbolic conservation laws. *J. Comput. Fluids* **174**, 34–45 (2018)
8. Huang, C., Li, L.: A new adaptively central-upwind sixth-order WENO scheme. *J. Comput. Phys.* **357**, 1–15 (2018)
9. Jiang, G.S., Shu, C.W.: Efficient implementation of Weighted ENO schemes. *J. Comput. Phys.* **126**, 202–228 (1996)
10. Nessyahu, H., Tadmor, E.: Non-oscillatory differencing for hyperbolic conservation laws. *J. Comput. Phys.* **87**, 408–463 (1990)
11. Takakura, Y.: Direct expansion forms of ADER schemes for conservation laws and their verification. *J. Comput. Phys.* **219**, 855–876 (2006)
12. Titarev, V.A., Toro, E.F.: ADER: arbitrary high order Godunov approach. *J. Sci. Comput.* **17**, 609–616 (2002)
13. Titarev, V.A., Toro, E.F.: High order ADER schemes for scalar advection-diffusion-reaction equations. *J. Comput. Fluid Dyn.* **12**, 1–5 (2003)
14. Titarev, V.A., Toro, E.F.: ADER schemes for three-dimensional nonlinear hyperbolic systems. *J. Comput. Phys.* **204**, 715–736 (2005)

15. Titarev, V.A., Toro, E.F.: Analysis of ADER and ADER-WAF schemes. *IMA J. Numer. Anal.* **27**, 616–630 (2007)
16. Toro, E.F.: A fast Riemann solver with constant covolume applied to random choice method. *Int. J. Numer. Methods Fluids* **9**, 1145–1164 (1989)
17. Toro, E.F., Titarev, V.A.: TVD Fluxes for the High-Order ADER Schemes. *J. Sci. Comput.* **24**(3), 285–309 (2005)
18. Toro, E.F., Titarev, V.A.: Derivative Riemann solvers for systems of conservation laws ADER methods. *J. Comput. Phys.* **212**, 150–165 (2006)
19. Toro, E.F.: *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd edn. Springer, Berlin (2009)
20. Wang, Y., Du, Y., Zhao, K., Yuan, L.: A new 6th-order WENO scheme with modified stencils. *J. Comput. Fluids* **208**, 104625 (2020)
21. Woodward, P., Colella, P.: The numerical solution of two-dimensional fluid flow with strong waves. *J. Comput. Phys.* **54**, 115–173 (1984)
22. Zahran, Y.H.: Third-order TVD scheme for hyperbolic conservation laws. *Bull. Belg. Math. Soc. Simon Stevin* **14**, 259–275 (2007)
23. Zahran, Y.H.: Central ADER schemes for hyperbolic conservation laws. *J. Math. Anal. Appl.* **346**, 120–140 (2008)
24. Zahran, Y.H.: An efficient WENO scheme for solving hyperbolic conservation laws. *Appl. Math. Comput.* **212**, 37–50 (2009)
25. Zahran, Y.H., Abdalla, A.H.: A new ninth-order central Hermite weighted essentially nonoscillatory scheme for hyperbolic conservation laws. *Int. J. Numer. Methods Fluids* **193**, 1645–1667 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.