# Central WENO Schemes
# for Hyperbolic Systems of Conservation Laws

Doron Levy† Gabriella Puppo‡ Giovanni Russo§

February 24, 1998

**Abstract**

We present a family of high-order, essentially non-oscillatory, central schemes for approximating solutions of hyperbolic systems of conservation laws. These schemes are based on a new centered version of the Weighed Essentially Non-Oscillatory (WENO) reconstruction of point-values from cell-averages, which is then followed by an accurate approximation of the fluxes via a natural continuous extension of Runge-Kutta solvers. We explicitly construct the third and fourth-order scheme and demonstrate their high-resolution properties in several numerical tests.

**Key words.** Hyperbolic conservation laws, central difference schemes, high-order accuracy, non-oscillatory schemes, WENO reconstruction, Runge-Kutta.

**AMS(MOS) subject classification.** Primary 65M10; secondary 65M05.

## Contents



†Département de Mathématiques et d'Informatique, Ecole Normale Supérieure, 45 rue d'Ulm, 75230 Paris Cedex 05, France; dlevy@dmi.ens.fr

‡Dipartimento di Matematica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; puppo@polito.it

§Dipartimento di Matematica, Università dell'Aquila, Via Vetoio, loc. Coppito - 67100 L'Aquila, Italy; russo@univaq.it

# 1 Introduction

In recent years, a tremendous amount of research was done in developing and implementing modern high-resolution methods for approximating solutions of hyperbolic systems of conservation laws. A review of such numerical methods can be found, e.g., in Godlewski and Raviart [7], LeVeque [17] and Tadmor [31].

Among the variety of methods for approximating solutions of such problems we focus on finite-difference methods, which can be divided into two main categories, namely *upwind* schemes and *central* schemes.

The prototype of *upwind* schemes is the first-order Godunov scheme in which a piecewise-constant interpolant (which is constructed based on previously computed cell-averages) is evolved exactly to the next time step according to the conservation law. This evolution involves a solution of Riemann problems on the boundaries of each cell, which is interpreted as an upwinding procedure, as one has to differ between left-going and right-going waves in order to compute the flux in these non-smooth regions.

For linear systems, a general procedure can be used, which is based on the characteristic decomposition of the field [17]. If the system is nonlinear, however, a general scheme for the (exact or approximate) solution of the Riemann problem is not known, and the upwind approach may be rather complicated and costly.

An alternative approach for quasilinear systems is given by the so-called *relaxation schemes* [14]. Given a quasilinear hyperbolic system of conservation laws, a semilinear system with stiff relaxation is constructed. Under suitable conditions the solution of the latter converges to the solution of the quasilinear system, as the relaxation parameter tends to zero. The numerical solution of the quasilinear system is obtained by a splitting procedure. During the convection step, a linear hyperbolic system is solved by upwind scheme. The relaxation step is treated by an implicit scheme (which can be explicitly solved). The two steps are suitably combined in order to obtain second order accuracy in time. They can be efficiently implemented in one and several dimensions. However, they have the drawback that the intermediate semilinear system is larger than the original quasilinear system, and it is difficult to design a splitting strategy that allows higher (larger than second) order it time.

A general procedure used to obtain high accuracy in space with upwind schemes is based on high order reconstruction of the field variables. This is obtained by approximating the field at a given time by a piecewise polynomial rather than by a piecewise constant. Generally, a piecewise reconstruction of degree $r$ should guarantee spatial accuracy of order $r+1$ for a smooth solution. Since the drawback of a high-order reconstruction is the oscillations it might create, several methods were suggested to combine the upwinding framework with a mechanism to prevent the creation and evolution of such spurious numerical oscillations. In particular, a class of *Essentially Non-Oscillatory* (ENO) schemes was presented in [9] and studied in numerous works (consult [8], [28], [27], and the references therein). ENO schemes are based on a reconstruction procedure (of either point-values or cell-averages) which involves a selection of a stencil in each cell of the grid. This stencil selection is based on minimizing the oscillations which can be created due to the approximation of the underlying function and/or its derivatives in non-smooth regions.

Recently, a new approach for the reconstruction procedure has been suggested in [22]. There, in the so-called *Weighted Essentially Non-Oscillatory* (WENO) reconstruction, instead of selecting one stencil according to a non-oscillatory criterion, one reconstruction is created by taking a convex combination of all the candidate stencils. The weights of this combination are determined through a non-linear computation which is based on the local smoothness of the stencil. Every stencil is weighted according to the oscillations which it might create. In discontinuous regions, e.g., the weights will be biased towards the stencils in the smoother regions. Since, effectively, in smooth regions the linear combination of the different stencils can be interpreted as a wide stencil, a higher-order scheme can be constructed without using polynomials of a higher degree in the reconstruction procedure. An efficient implementation of WENO schemes and a new criterion for measuring the local smoothness of the stencils was presented in [12].

*Central schemes* on the other hand, can be viewed as a high-order extension to the first-order Lax-Friedrichs scheme (LxF) [6]. In its staggered version, the LxF scheme is based on constructing a piecewise-constant reconstruction which is then evolved exactly in time and finally projected on its staggered cell-averages. Due to the staggering, the approximation of the evolved fluxes is done in smooth regions (up to an appropriate CFL condition). Hence, no characteristic decomposition is required and the upwinding is replaced by a straightforward centered computation of the quantities involved. In such a way, no Riemann-solvers are required; a quality which is evident for a general one-dimensional system and in higher space dimensions where no such Riemann-solvers exist.

The accuracy of the schemes can be increased by using a higher-order reconstruction and a sufficiently accurate quadrature rule for the approximation of the fluxes. The one-dimensional second-order Nessyahu-Tadmor scheme is presented in [24]. A different approach to second-order central differencing based on characteristics tracing was introduced by Sanders and Weiser in [26]. Extensions to third-order can be found in [23], [10]. For an extension of the second-order scheme to two space dimensions see [13], [1], [2]. In all of the above cases, a non-linear limiting augments the reconstruction in order to prevent oscillations. A third-order, two-dimensional central scheme for the two-dimensional incompressible Euler equations is presented in [20], [18]. A central-scheme on unstaggered meshes was derived in [3], [4].

We note in passing that staggering can be eliminated by transforming the staggered schemes into non-staggered schemes without loss of accuracy (consult [11]).

A first step to combining the upwind and central approaches was taken in [5]. There, high-order, non-oscillatory central schemes were constructed based on a high-order ENO reconstruction step. These schemes were shown to enjoy the desired properties of both approaches; the central framework provided the robustness and the simplicity while the

ENO reconstruction provided the required high-order, non-oscillatory reconstruction. In the same paper, Runge-Kutta schemes have been used for the time integration of the flux on cell boundaries. Function evaluations are minimized by using the Natural Continuous Extension of Runge Kutta schemes [34]. In particular, third and fourth-order schemes for general systems were presented. and the expected accuracy has been numerically confirmed.

In this work, which can be viewed as a natural extension of [5], we bring both the central and upwind approaches even closer. Here, we construct a central scheme in which the main ingredient is a new, centered version of a WENO reconstruction. This new approach enables one to create, e.g., from a piecewise-parabolic reconstruction (based on cell-averages), a central, fourth-order accurate method.

In a future work [19], we will extend these ideas to the setup of two-dimensional systems of conservation laws.

The paper is organized as follows: In §2 we briefly overview the general framework of central schemes for one-dimensional hyperbolic systems of conservation laws. Our new centered-WENO reconstruction is presented in §3. The resulting method is then summarized in §4, and finally, several canonical numerical examples can be found in §5. These numerical examples clearly demonstrate the accuracy, non-oscillatory and robustness properties of our scheme.

**Acknowledgment:** Research was supported by TMR grant #... Part of this work was done while the first and second authors were visiting L'Aquila, and while the third author was visiting ENS, Paris. We would like to thank G.-S. Jiang and S. Osher for their useful suggestions.

## 2 Central Schemes - a short overview

We are concerned with the approximate solutions of systems of hyperbolic conservation laws,

$$u_t + f(u)_x = 0, \qquad u \in \mathbb{R}^d, d \geq 1, \tag{2.1}$$

subject to the initial conditions, $u(x, t = 0) = u_0(x)$.

To approximate solutions of (2.1) we discretize both space and time assuming uniform mesh spacings of $h := \Delta x$ and $\Delta t$, respectively. We denote the spatial grid-points by $x_j = j\Delta x$ and the time steps by $t^n = n\Delta t$. Here and below $\lambda := \Delta t/\Delta x$ denotes the usual fixed mesh-ratio. Since the solutions of (2.1) can develop discontinuous solutions (shocks) even for smooth initial data, the quantities that will be used on the discrete level are cell-averages. The cell-average in the cell $I_j := [x_{j-1/2}, x_{j+1/2}]$ centered around $x_j$ at time $t^n$, is denoted by $\bar{u}_j^n$:

$$\bar{u}_j^n = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n)dx.$$

Assuming that the cell-averages at time $t^n$, $\bar{u}_j^n$, are known, our goal is to compute the cell-averages at the next time step $t^{n+1}$. First, from $\bar{u}_j^n$, we reconstruct a piecewise-polynomial interpolant

$$P_u(x,t^n) := \sum_j R_j(x)\chi_j, \tag{2.2}$$

taking into account conservation, accuracy and non-oscillatory requirements. Here, $\chi_j$ is the characteristic function of the interval $I_j$, and $R_j(x)$ is a polynomial of degree $r$ which is defined in $I_j$.

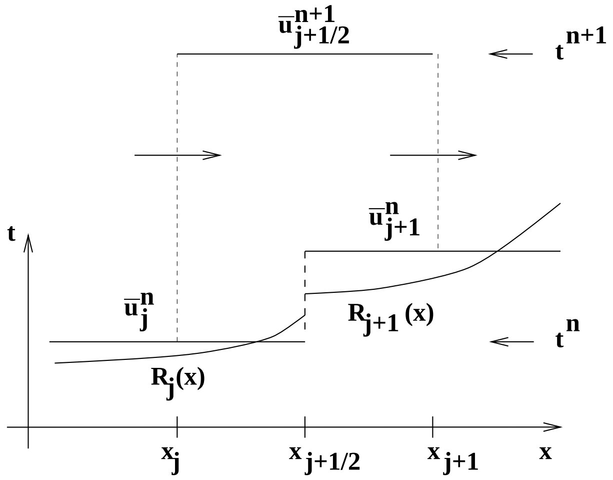


Figure 2.1: Reconstruction and Projection

The interpolant $P_u(x,t^n)$ in (2.2) is then evolved to the next time step integrating (2.1) on its staggered cells (consult figure 2.1). Note that due to the staggering the time integration of the fluxes is performed in a smooth region and therefore the integral can be accurately approximated by a quadrature formula. The staggered cell-averages, $\bar{u}_{j+1/2}^{n+1}$ are obtained by

$$\begin{aligned}\bar{u}_{j+1/2}^{n+1} &= \frac{1}{h}\int_{x_j}^{x_{j+1}} P_u(x,t^n)dx \\ &+\lambda\sum_{l=0}^{m}\gamma_l\left[f(\hat{u}(x_j,t^n+\beta_l\Delta t)) - f(\hat{u}(x_{j+1},t^n+\beta_l\Delta t))\right].\end{aligned} \tag{2.3}$$

The parameters $\gamma$ and $\beta_l$ are the weights and the nodes of the particular quadrature formula, and $\hat{u}$ are intermediate values predicted either by a Taylor approximation or by a Runge-Kutta (RK) method. For a fourth-order method one can use, e.g., Simpson's rule. The staggered cell-averages at time $t^n$, $\bar{u}_{j+1/2}^n$, are given by

$$\bar{u}_{j+1/2}^n = \frac{1}{h}\int_{x_j}^{x_{j+1}} P_u(x,t^n)dx = \frac{1}{h}\left[\int_{x_j}^{x_{j+1/2}} R_j(x)dx + \int_{x_{j+1/2}}^{x_{j+1}} R_{j+1}(x)dx\right]. \tag{2.4}$$

Given a specific reconstruction $\{R_j\}$, the integrals on the RHS of (2.4) can be explicitly computed.

In the case of systems a RK method is simpler because it does not require the computation of the Jacobian or the Hessian of the system. A great saving in function evaluation when using a RK method can be obtained constructing its *Natural Continuous Extension* (NCE) described below. The main idea is to advance to the last time node of the quadrature using a one-step RK method and computing the other intermediate values by a suitable polynomial reconstruction.

## 2.1 Time evolution: Runge-Kutta with Natural Continuous Extension

Consider the Cauchy problem

$$\begin{cases} y' = F(t, y(t)) \\ y(t_0) = y_0. \end{cases}$$

We adapt to our context the notation used in [34] according to which the solution at the $(n+1)$-th time step obtained with a $\nu$ stage Runge-Kutta scheme can be written as

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{\nu} b_i g^{(i)},$$

where the $g^{(i)}$'s are the approximate Runge-Kutta fluxes

$$g^{(i)} = F\left(t^n + \Delta t c_i, y^n + \Delta t \sum_{j=1}^{\nu-1} a_{ij} g^{(j)}\right),$$

and the $c_i$ are given by $c_i = \sum_j a_{ij}$. The method is completely determined by the vector $b$ and the matrix $a$, which is lower triangular for explicit schemes.

In our case we are solving a sequence of Cauchy problems in order to obtain the predicted mid-values required for the quadrature of the fluxes. At the $j$-th grid point we have

$$\begin{cases} y_j'(\tau) &=& F(\tau, y_j(\tau)) = -f_x\left(y(x_j, t^n + \tau)\right) \\ y_j(\tau = 0) &=& P_u(x_j, t^n). \end{cases} \tag{2.5}$$

Thus the computation of the $i$-th Runge-Kutta flux $g^{(i)}$ requires the evaluation of the $x$-derivative of $f$ at the intermediate time $t = t^n + c_i \Delta t$. The predicted point-values of $u_j$ at time $t = t^n + c_i \Delta t$ are used to compute the point-values, $f(u_j)$. These predicted values of $f$ are then used for reconstructing an interpolant from which the point-values of the derivative $(f_x)_j$ are computed. To maintain high accuracy and control over oscillations in the evaluation of $f_x$ required in (2.5), the reconstruction of the interpolant of both $f$ and $u$ is essentially the same and is described in §3.

Following [5], we use here *Natural Continuous Extensions* (NCE) of Runge-Kutta schemes which provide a uniform accuracy of the solution in the time interval $[t^n, t^{n+1}]$ (consult [34]).

Each $\nu$-stage Runge-Kutta method of order $p$ has an NCE $u$ of degree $d \leq p$ in the sense that there exist $\nu$ polynomials $b_i(\theta), i = 1, \cdots, \nu$ of degree at most $d$, such that

1. $u(t^n + \theta\Delta t) := y^n + \Delta t \sum_{i=1}^{\nu} b_i(\theta) g^{(i)} \qquad 0 \le \theta \le 1;$
2. $u(t^n) = y^n \qquad$ and $\qquad u(t^n + \Delta t) = y^{n+1};$
3. $\max_{t^n \le t \le t^n + \Delta t} \left| w^{(l)}(t) - u^{(l)}(t) \right| = O((\Delta t)^{d+1-l}), \qquad 0 \le l \le d,$

where $y^n$ is the numerical solution computed with the RK scheme at time level $t^n$, and $w(t)$ is the exact solution of the equation with $y(t^n) = y^n$. Note that the polynomials $b_i(\theta)$ depend only on the Runge-Kutta method chosen and not on the particular ODE being solved. In §4 we shall show how to use Runge-Kutta methods with NCE in order to guarantee high order accuracy in time.

We end this section listing the NCE's which are of interest for our scheme.

$$
\begin{array}{ll}
1) & \mathsf{RK1} \qquad \text{(for our \textit{second-order} scheme)} \\
& d = \nu = p = 1 \\
& b_1(\theta) = \theta; \\
& \\
2) & \mathsf{RK2} \qquad \text{(for our \textit{third-order} scheme)} \\
& d = \nu = p = 2 \\
& b_1(\theta) = (b_1 - 1)\theta^2 + \theta \\
& b_2(\theta) = b_2\theta^2; \\
& \\
3) & \mathsf{RK3} \qquad \text{(Not used since no NCE of degree } d = 3 \text{ exists in this case)} \\
& d = 2, \nu = p = 3 \\
& b_i(\theta) = 3(2c_i - 1)b_i\theta^2 + 2(2 - 3c_i)b_i\theta \qquad i = 1, 2, 3 \\
& \\
4) & \mathsf{RK4} \qquad \text{(for our \textit{fourth-order} scheme)} \\
& d = 3, \nu = p = 4 \\
& b_1(\theta) = 2(1 - 4b_1)\theta^3 + 3(3b_1 - 1)\theta^2 + \theta \\
& b_i(\theta) = 4(3c_i - 2)b_i\theta^3 + 3(3 - 4c_i)b_i\theta^2 \qquad i = 2, 3, 4 \\
& \\
& \text{No NCE of degree } d = 4 \text{ exists in this case.}
\end{array}
\tag{2.6}
$$

In our case, we used the following set of coefficients. For our third order method:

$$
b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \qquad\qquad a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{2.7}
$$

For our fourth order method:

$$
b = \begin{pmatrix} 1/6 \\ 1/3 \\ 1/3 \\ 1/6 \end{pmatrix} \qquad\qquad a = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.8}
$$

We proceed in the next section by presenting our new Centered-WENO (CWENO) reconstruction which supplies the required elements, $R_j$, for the overall piecewise-polynomial interpolant (2.2).

# 3 The Centered-Weno (CWENO) Reconstruction

In this section, we present our new Centered-WENO (CWENO) piecewise-parabolic reconstruction which will be then utilized in §4 to construct a fourth-order method.

Let $u$ be the exact solution at time $t^n$, and $\bar{u}_j$ be its cell averages on the cell $(x_j - h/2, X_j + h/2)$. Starting from the data $\{\bar{u}_j\}$ we apply the reconstruction scheme, obtaining the function $P_u(x,t^n)$. By construction, $P_u(x,t^n) = \sum_j R_j(x)\chi_j$, where in this case $R_j(x)$ is a polynomial of degree 2. We require that our reconstruction satisfies the following properties:

1. Accuracy
   We shall require that Eq. (2.4) provides a high order accurate approximation of the staggered cell averages at time $t^n$, i.e.

$$\bar{u}^n_{j+1/2} = \frac{1}{h}\int_{x_j}^{x_{j+1}} u(x,t^n)\,dx + O(h^s),$$

   where $s$ denotes the spatial order of the method.

   Thus the polynomial reconstruction from cell-averages must satisfy:

$$\begin{aligned}\frac{1}{2h}\int_{x_{j-1/2}}^{x_j} P_u(x,t^n)dx &= \frac{1}{2h}\int_{x_{j-1/2}}^{x_j} u(x,t^n)dx + O(h^s),\\ \frac{1}{2h}\int_{x_j}^{x_{j+1/2}} P_u(x,t^n)dx &= \frac{1}{2h}\int_{x_j}^{x_{j+1/2}} u(x,t^n)dx + O(h^s).\end{aligned} \tag{3.1}$$

   Moreover we require:

$$P_u(x_j,t^n) = u(x_j,t^n) + O(h^s). \tag{3.2}$$

   A separate reconstruction of $f$ will be used for the derivatives of the fluxes, $f'(u(x,t))$, for which we require

$$P_f'(x_j,t^n) = f'(u(x_j,t^n)) + O(h^{s-1}). \tag{3.3}$$

2. Conservation

$$\frac{1}{h}\int_{I_j} R_j(x)dx = \bar{u}_j. \tag{3.4}$$

3. Non-oscillatory Reconstruction
   Avoid spurious oscillations in the sense of ENO/WENO reconstruction ([9], [22]).

In each cell, $I_j$, we reconstruct three polynomials of degree 2, $P_{j-1}(x), P_j(x), P_{j+1}(x)$. Each of these polynomials is constructed by posing the following interpolation requirements:

$$\begin{cases} \frac{1}{h}\int_{I_{k-1}} P_k(x)dx = \bar{u}_{k-1}, \\ \frac{1}{h}\int_{I_k} P_k(x)dx = \bar{u}_k, \\ \frac{1}{h}\int_{I_{k+1}} P_k(x)dx = \bar{u}_{k+1}. \end{cases} \qquad k = j-1, j, j+1 \tag{3.5}$$

The reconstruction is created by considering a convex combination of the above polynomials, $P_k(x)$,

$$R_j(x) = w^j_{j-1}P_{j-1}(x) + w^j_j P_j(x) + w^j_{j+1}P_{j+1}(x), \tag{3.6}$$

where the weights $w^j_k, k = j-1, j, j+1$, satisfy $w^j_k \geq 0$, and $\sum_{k=j-1}^{j+1} w^j_k = 1$. The stencil used in the reconstruction of the second degree polynomial $R_j(x)$ contains five points. Note that this convex combination retains the interpolation requirement (3.4) for $R_j(x)$ in $I_j$, but otherwise does not fulfill any other interpolation requirements in the neighboring cells.

Since $deg(P_k(x)) = 2, k = j-1, j, j+1$, one can rewrite

$$P_k(x) = \tilde{u}_k + \tilde{u}'_k(x - x_k) + \frac{1}{2}\tilde{u}''_k(x - x_k)^2, \quad k = j-1, j, j+1. \tag{3.7}$$

The reconstructed point-values, $\tilde{u}_k$, and the reconstructed discrete first and second derivatives, $\tilde{u}'_k, \tilde{u}''_k$, are uniquely determined by the interpolation requirements (3.5), as

$$\begin{aligned} \tilde{u}_k &= \bar{u}_k - \frac{\bar{u}_{k-1} - 2\bar{u}_k + \bar{u}_{k+1}}{24}, \\ \tilde{u}'_k &= \frac{\bar{u}_{k+1} - \bar{u}_{k-1}}{2h}, \\ \tilde{u}''_k &= \frac{\bar{u}_{k+1} - 2\bar{u}_k + \bar{u}_{k-1}}{h^2}, \qquad k = j-1, j, j+1. \end{aligned} \tag{3.8}$$

Hence, the interpolant $R_j(x)$ can be written as

$$R_j(x) = u_j + u'_j(x - x_j) + \frac{1}{2}u''_j(x - x_j)^2, \tag{3.9}$$

where its reconstructed point-values $u_j$, and its reconstructed derivatives, $u'_j, u''_j$, are given by

$$\begin{aligned} u_j &= w^j_{j-1}(\tilde{u}_{j-1} + h\tilde{u}'_{j-1} + \frac{1}{2}h^2\tilde{u}''_{j-1}) + w^j_j\tilde{u}_j + w^j_{j+1}(\tilde{u}_{j+1} - h\tilde{u}'_{j+1} + \frac{1}{2}h^2\tilde{u}''_{j+1}), \\ u'_j &= w^j_{j-1}(\tilde{u}'_{j-1} + h\tilde{u}''_{j-1}) + w^j_j\tilde{u}'_j + w^j_{j+1}(\tilde{u}'_{j+1} - h\tilde{u}''_{j+1}), \\ u''_j &= w^j_{j-1}\tilde{u}''_{j-1} + w^j_j\tilde{u}''_j + w^j_{j+1}\tilde{u}''_{j+1}. \end{aligned} \tag{3.10}$$

All that is left in order to end the reconstruction is to determine the weights, $w_k^j, k = j-1, j, j+1$. Two ingredients are taken into account in their construction: the accuracy requirements and the non-oscillatory requirements.

In order to guarantee convexity, $\sum_{k=j-1}^{j+1} w_k^j = 1$, the weights, $w_k^j$, are written as

$$w_k^j = \frac{\alpha_k^j}{\alpha_{j-1}^j + \alpha_j^j + \alpha_{j+1}^j}, \qquad k = j-1, j, j+1, \tag{3.11}$$

where

$$\alpha_k^j = \frac{C_k}{(\epsilon + IS_k^j)^p}, \qquad k = j-1, j, j+1. \tag{3.12}$$

The constants, $C_k, \epsilon, p$, and the smoothness indicator, $IS_k^j$, will be determined below.

Since we are in the framework of central schemes and not upwind schemes, our accuracy requirements are different from those found in [22]. Here is exactly where the *central* philosophy enters. The constants $C_k$ introduced here, are computed such that they are symmetric around the center of each cell. Consequently, we name our new centered reconstruction as *Central-WENO (CWENO).*

Since we can not satisfy all the accuracy requirements simultaneously, we split the computation into two parts and by that we are led to two different sets of constants $C_k$. The first set corresponds to the accuracy requirement in the reconstruction of the cell-averages (3.1), while the second set of constants corresponds to the accuracy requirements in the reconstruction of the derivatives (3.3). Due to cancellation, any symmetric choice of coefficients will result in a fourth-order approximation of the point-values in the center of the cells ($s = 4$ in (3.2)). In order to satisfy (3.2) for $s = 5$ one has to use non-positive constants, namely a *non-convex* combination of the stencils – see the remarks below. Clearly, the use of two separate sets of constants imposes absolutely no problems on the implementation of the algorithm. We note that also in the original WENO paper [22], two sets of (different) constants were suggested. There, the motivation was to adapt the constants to the upwinding and hence they were determined by the characteristic variables. In our case, however, no such characteristic decomposition is required as the weights are *fixed.* The simplicity of the central framework is projected onto our new Central-WENO reconstruction.

A straightforward computation results with the desired constants which are displayed in Table 3.1.

| | $C_{j-1}$ | $C_j$ | $C_{j+1}$ | accuracy |
|---|---|---|---|---|
| cell-averages | 3/16 | 5/8 | 3/16 | $h^5$ |
| derivatives | 1/6 | 2/3 | 1/6 | $h^4$ |
| point-values | any symmetric combination | | | $h^4$ |

Table 3.1: The constants of the Central-WENO reconstruction

Several different ways to determine the smoothness indicator were suggested in the literature (see, e.g., [22] and [12]). Here we use the measure taken from [12], which amounts to a measure on the $L^2$-norms of the derivatives:

$$IS_k^j = \sum_{l=1}^{2} \int_{x_{j-1/2}}^{x_{j+1/2}} h^{2l-1} (P_k^{(l)})^2 dx, \qquad k = j-1, j, j+1, \tag{3.13}$$

where $P_k^{(l)}$ denotes the $l$-th derivative of $P_k(x)$. An explicit integration of (3.13) yields

$$\begin{aligned} IS_{j-1}^j &= \frac{13}{12}(\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j)^2 + \frac{1}{4}(\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j)^2, \\ IS_j^j &= \frac{13}{12}(\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1})^2 + \frac{1}{4}(\bar{u}_{j-1} - \bar{u}_{j+1})^2, \\ IS_{j+1}^j &= \frac{13}{12}(\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2})^2 + \frac{1}{4}(3\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2})^2. \end{aligned} \tag{3.14}$$

In smooth regions, a Taylor expansion of (3.14) gives

$$IS_k^j = (\tilde{u}'h)^2 + \frac{13}{12}(\tilde{u}''h^2)^2 + O(h^6), \qquad k = j-1, j, j+1. \tag{3.15}$$

Hence, $IS_k^j = O(h^2)$, and in critical points it is $O(h^4)$. In non-smooth regions, $IS_k^j = O(1)$, and by that the normalized weight of the corresponding stencil will be negligible. Therefore, our reconstruction follows the WENO methodology by automatically avoiding the information coming from non-smooth regions which are the cause for spurious oscillations.

The remaining parameters to be determined in (3.12) are $\epsilon$ and $p$. The constant $\epsilon$ was inserted in the denominator in order to prevent it from vanishing. In [12] an $\epsilon = 10^{-6}$ was empirically selected. Here, we find that the scheme is almost not sensitive to the value of $\epsilon$, and $\epsilon = 10^{-6}$ proved adequate for the numerical results in §5. The value of $p$ was determined in [22] as one above the degree of the reconstruction polynomial, which in our case amounts to 3. In [12] a $p = 2$ was empirically selected and here we use the same value $p = 2$.

In short, our reconstruction from cell averages routine accepts in input the values $\{\bar{u}_j\}$ of cell averages at time $t^n$, and produces in output the point values $u_j, u'_j, u''_j$ which completely determine the reconstruction polynomial $R_j(x)$.

A few modifications are needed to compute the reconstruction from point values for the flux $f_j = f(u_j)$ which is needed in the Runge-Kutta step.

Here the candidate polynomials $P_k$ satisfy the interpolation requirements (compare with (3.5)):

$$\begin{cases} P_k(x_{k-1}) = f_{k-1}, \\ P_k(x_k) = f_k, \\ P_k(x_{k+1}) = f_{k+1}. \end{cases} \qquad k = j-1, j, j+1 \tag{3.16}$$

Thus the reconstructed point values $\tilde{f}_k$ and the reconstructed first and second derivatives $\tilde{f}'_k$ and $\tilde{f}''_k$ are given by:

$$\tilde{f}_k = f_k, \quad \tilde{f}'_k = \frac{f_{k+1} - f_{k-1}}{2h}, \quad \tilde{f}''_k = \frac{f_{k+1} - 2f_k + f_{k-1}}{h^2}, \quad k = j-1, j, j+1.$$

For the evaluation of the intermediate values, all that will be needed is a pointwise reconstruction of the space derivative $f'_j$ which is given by:

$$f'_j = w^j_{j-1}(\tilde{f}'_{j-1} + h\tilde{f}''_{j-1}) + w^j_j \tilde{f}'_j + w^j_{j+1}(\tilde{f}'_{j+1} - h\tilde{f}''_{j+1}). \tag{3.17}$$

The computation of the weights in (3.17) is the same as above (3.11)–(3.15). Here, we use the second set of constants $C_k$'s of Table 3.1, and the computation of the smoothness indicators $IS^j_k$ in (3.14) involves the point values $f_j$, instead of the cell averages.

Several remarks are in order.

**Remarks:**

1. In [5] the ENO reconstruction was first combined with central schemes. There, it was shown that an ENO reconstruction of degree 3 is required to obtain an overall third-order method (when the flux integration is done by a Runge-Kutta solver). Here we construct a piecewise-parabolic interpolant that is then utilized to obtain a fourth-order accurate method.

2. The piecewise-parabolic upwind-WENO reconstruction is utilized in [22] to obtain a fifth-order method, which is more accurate than our fourth-order method. On the other hand, it suffers from the drawbacks of the upwind approach. We can also obtain a fifth-order method, requiring $s = 5$ in (3.1)–(3.3). Unfortunately, in (3.2), $s = 5$ leads to a *non-convex* combination with constants given by $C_{j-1} = C_{j+1} = -9/80$. Such a combination does not fit in our framework. It should be possible to construct a fifth-order method which reduces to lower order convex reconstruction by taking a convex combination of the fifth-order method and the fourth-order method described above.

3. Note that the value of $\epsilon$ depends on the order of magnitude of the solution and on the value of $h$. Here, we assume that the solution is of order 1. One can avoid this dependence by a suitable rescaling of the indicators (as explained in the case of systems below).

To summarize, our new Central-WENO reconstruction enjoyed the best of two worlds. On one hand, it follows the WENO methodology by making use of the existing information in order to automate the selection of the stencil and to increase the overall order of accuracy of the resulting reconstruction while retaining its non-oscillatory properties. On the other hand, since we are dealing with the central framework, our reconstruction makes no use of characteristic decomposition and by that we are left with a simple and robust machinery.

# 4 The Method

In this section we combine the central framework which was overviewed in §2 with our new CWENO reconstruction of §3. The derivation of the resulting scheme is straightforward and is summarized in the following algorithm, which applies to the scalar case.

**Algorithm:**

Given the cell-averages $\bar{u}_j^n$, at time $t^n$, compute the staggered cell-averages at the next time step $\bar{u}_{j+1/2}^{n+1}$, as follows.

Step 1:
Compute the coefficients of the parabolic interpolant $R_j(x)$ $\forall j$, (3.9),

$$R_j(x) = u_j + u_j'(x - x_j) + \frac{1}{2}u_j''(x - x_j)^2.$$

The point-values $u_j$, and the discrete derivatives, $u_j'$ and $u_j''$, are given in (3.10),(3.8). The weights $w_k$ required in (3.10) are computed by (3.11)-(3.12) and are based on the oscillatory indicators $IS_k^j$ in (3.14). The constants $C_k$ are the first set of constants appearing in Table 3.1 (cell-averages).

Step 2:
Compute the point-values of the flux $f(u_j^n)$, where the point-values $u_j$ are given from the reconstruction Step 1. Utilize these point-values to reconstruct the discrete first derivative of the flux $f'(u_j^n)$ following the reconstruction procedure (3.16)–(3.17).

$$f_j' = w_{j-1}^j(\tilde{f}_{j-1}' + h\tilde{f}_{j-1}'') + w_j^j\tilde{f}_j' + w_{j+1}^j(\tilde{f}_{j+1}' - h\tilde{f}_{j+1}'').$$

The weights in (3.17) are based on (3.12) where this time the constants $C_k$ are the second set of constants appearing in Table 3.1, and the smoothness indicators are given by (3.14) where $\bar{u}_k$ is replaced by $f_k$, $k = j-2, \ldots, j+2$.

Step 3:
Compute the predicted values, at the nodes $\beta_l$ of the quadrature formula $\hat{u}(x_j, t^n + c_l\Delta t)$. We use a Runge-Kutta method with NCE (see §2.1):

$$\hat{u}(x_j, t^n + \beta_l\Delta t) = u_j + \Delta t\sum_{i=1}^{\nu} b_i(\beta_l)K_{i;j},$$

$$K_{i;j} = F(Y_j^i; j), \qquad Y_j^i = u_j + \Delta t\sum_{s=1}^{i-1} a_{is}K_s,$$

where the function $F(\hat{u}; j)$ is an approximation of the space derivative $-\partial f(\hat{u})/\partial x$, computed as to satisfy (3.3), $F(Y_j^i; j) = -f'(Y_j^i)$ according to Step 2. The values of the coefficients $a_{ij}$ are listed in (2.7) for our third order method and in (2.8) for our fourth order method. In particular, for the third and fourth-order schemes one can use Simpson's quadrature rule which requires to predict $\hat{u}_j^{n+1/2}, \hat{u}_j^{n+1}$, $(\beta_1 = 1/2, \beta_2 = 1)$.

Step 4:
Compute the staggered cell-averages $\bar{u}^{n+1}_{j+1/2}$ according to (2.3),

$$\begin{aligned}\bar{u}^{n+1}_{j+1/2} &= \bar{u}^{n}_{j+1/2} + \lambda \sum_{l=0}^{m} \gamma_l \left[ f(\hat{u}(x_j, t^n + \beta_l \Delta t)) - f(\hat{u}(x_{j+1}, t^n + \beta_l \Delta t)) \right] \\ &:= \mathcal{I}_1 + \mathcal{I}_2. \end{aligned} \tag{4.18}$$

Here:

1. The staggered cell-averages on the RHS of (4.18), $\mathcal{I}_1$, are given in (2.4),

$$\begin{aligned}\mathcal{I}_1 := \bar{u}^{n}_{j+1/2} &= \frac{1}{h}\left[ \int_{x_j}^{x_{j+1/2}} R_j(x)dx + \int_{x_{j+1/2}}^{x_{j+1}} R_{j+1}(x)dx \right] \\ &= \frac{u_j + u_{j+1}}{2} - \frac{u'_{j+1} - u'_j}{8}h + \frac{u''_j + u''_{j+1}}{48}h^2. \end{aligned}$$

   The point-values and the derivatives, $u_j, u'_j, u''_j$, were computed in Step 1.

2. For a third and fourth-order schemes, $\mathcal{I}_2$ can be written as Simpson's rule

$$\begin{aligned}\mathcal{I}_2 \;:=\; & \frac{\lambda}{6}\{[f(\hat{u}^n_j) + 4f(\hat{u}^{n+1/2}_j) + f(\hat{u}^{n+1}_j)] \\ & -[f(\hat{u}^n_{j+1}) + 4f(\hat{u}^{n+1/2}_{j+1}) + f(\hat{u}^{n+1}_{j+1})]\}, \end{aligned}$$

   where the predicted point-values $\hat{u}^{n+1/2}_j = \hat{u}(x_j, t^n + 1/2\Delta t$ and $\hat{u}^{n+1}_j = \hat{u}(x_j, t^n + \Delta t$, are the result of Step 3.

We end this section sketching the algorithm for the case of systems of $d$ conservation laws. In the following $\mathbf{u}$ denotes the vector with components $\mathbf{u} = (u^1, \cdots, u^d)$.

**Algorithm:**
Given the cell-averages $\bar{\mathbf{u}}^n_j$, at time $t^n$, compute the staggered cell-averages at the next time step $\bar{\mathbf{u}}^{n+1}_{j+1/2}$, as follows.

Step 1:
Compute the coefficients of the parabolic interpolant $R_j(x)$ $\forall j$, applying Step 1 of the algorithm for the scalar case at each component of the vector $\bar{\mathbf{u}}^n_j$. Note that the coefficients of the interpolant are now $d$ components vectors. For the computation of the oscillatory indicators $IS^j_k$ several strategies are possible. One can apply formulas (3.14) to each component of $\bar{\mathbf{u}}^n_j$, obtaining different smoothness indicators for each component or use information coming from all components as in the Global strategy defined in (5.1). For more details, see the discussion in §5.2.

Step 2:
Compute the point-values of the flux $\mathbf{f}(\mathbf{u}_j^n)$, where the point-values $(u_j^1, \cdots, u_j^d)$ are given from the reconstruction Step 1. Utilize these point-values to reconstruct the discrete first $x$-derivative of each component of the flux function $\mathbf{f}(\mathbf{u}_j^n)$ applying the reconstruction procedure (3.16)–(3.17) at each component of $\mathbf{f}$. For the smoothness indicators, we can apply the scheme described in Step 2 of the scalar case to each component of $\mathbf{f}$, or apply the observation of the remark below, and use the smoothness indicators already compted in Step 1 from cell averages.

Step 3:
Compute the predicted values of $\mathbf{f}(\mathbf{u})$, at the nodes $\beta_l$ of the quadrature formula, using a Runge-Kutta method with NCE (see §2.1). We can follow the same lines of Step 3 for the scalar case, except that now $\hat{u}$, $u$, $K_i$, and $Y_j^i$ are $d$ components vectors. In particular, note that each component of $\mathbf{Y}_j^i$ must be updated before $\mathbf{f}(\mathbf{Y}_j^i)$ can be computed. Next each componet of $\mathbf{F}(\mathbf{Y}_j^i; j)$ can be computed applying our discrete differentiation scheme component by component to the vector $\mathbf{f}(\mathbf{Y}_j^i)$.

Step 4:
Apply Step 4 of the algorithm for the scalar case to each component of the conservation law.

**Remark**

The central framework is based on the assumption that the solution remains smooth at the boundaries of the staggered cell $(x_j, x_{j+1})$, if the appropriate CFL condition is satisfied. The discontinuities arising from the generalized Riemann problems defined at $x_{j+1/2}$do not have the time to reach the cell boundaries. Thus it is quite reasonable to assume that the degree of smoothness of the solution at $x_j$ and $x_{j+1}$ will not change abruptly within one time step. Therefore the smoothness indicators can be computed only once per time step. More precisely, we can compute the smoothness indicators from cell averages, as indicated in Step 1 of our algorithms, and utilize these same quantities when computing the weights of our reconstruction at each intermediate time of the RK step.

This technique will be applied to the case of systems of equations in §5.2. We emphasize that this simplification in the computation of the smoothness indicators is closely linked to the central framework.

# 5 Numerical Results

In this section we test our third and fourth-order schemes summarized in §4. We start from a single scalar equation where we numerically compute the order of accuracy of our schemes. We also demonstrate on a model problem how the smoothness indicators trigger the selection of the correct stencil when discontinuities are present. We end the discussion of the scalar case with a linear problem proposed by Shu [12].

The construction of the smoothness indicators is more delicate for systems of equations than in the scalar case. We show with numerical examples that a naive component by component extension of the scalar scheme does not yield the best results. We formulate and test two different algorithms to address this problem.

We then apply our scheme to some classical test problems of gas dynamics. Our results show that the use of the same smoothness indicator for all the components produces better results and is computationally cheaper compared with the componentwise indicator.

## 5.1 Scalar equation

We study the performance of our schemes by applying them to the following test problems:

**Test #1**

$u_t + u_x = 0$,
$u(x, t = 0) = \sin(\pi x)$,
periodic boundary conditions on [-1,1],
integration time: $T = 10$.

This test is used to check the convergence rate at large times.

**Test #2**

$u_t + u_x = 0$,
$u(x, t = 0) = \sin^4(\pi x)$,
periodic boundary conditions on [-1,1],
integration time: $T = 1$.

This test is used to detect possible deteriorations of accuracy due to strong oscillations in the parameters that determine the stencil (such as in ENO schemes). See the discussion in [5] and references therein.

**Test #3**

$u_t + (\frac{1}{2}u^2)_x = 0$,
$u(x, t = 0) = 1 + \frac{1}{2}\sin(\pi x)$,
periodic boundary conditions on [-1,1],
integration times: $T = 0.33$ and $T = 1.5$.

Here $T = 0.33$ is used for convergence tests, and $T = 1.5$ for the shock capturing test (the shock develops at $T_s = 2/\pi$).

**Test #4**

$u_t + u_x = 0$,
$u(x, t = 0) = u_0(x)$,
periodic boundary conditions on [-1,1],
integration time: $T = 8$.

This test is used to show the resolution properties of the scheme. The initial data $u_0(x)$ is defined in [12, Example 1] as:

$$u_0(x) = \begin{cases} \frac{1}{6}\left(G(x, z-\delta) + G(x, z+\delta) + 4G(x, z)\right), & -0.8 \le x \le -0.6, \\ 1, & -0.4 \le x \le -0.2, \\ 1 - |10(x-0.1)|, & 0 \le x \le 0.2, \\ \frac{1}{6}\left(F(x, a-\delta) + F(x, a+\delta) + 4F(x, a)\right), & 0.4 \le x \le 0.6, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} G(x, z) &= e^{-\beta(x-z)^2}, \\ F(x, a) &= \left(\max(1 - \alpha^2(x-a)^2, 0)\right)^{1/2}. \end{aligned}$$

The constants are taken as $a = 0.5, z = -0.7, \delta = 0.005$, $\alpha = 10$, and $\beta = (\log 2)/36\delta^2$.

Let $u(x, t^n)$ and $u_j^n$ be, as usual, the exact solution and the reconstructed solution at $(x_j, t^n)$, respectively. Then the norms of the error are given by:

$$\begin{aligned} L^1 - \text{error}: &\quad ||\text{Error}||_1 = \textstyle\sum_{j=1}^N |u(x_j, t^n) - u_j^n| h, \\ L^\infty - \text{error}: &\quad ||\text{Error}||_\infty = \max_{1 \le j \le N} |u(x_j, t^n) - u_j^n|. \end{aligned}$$

The computational parameters used in the following tests are $\epsilon = 10^{-6}$, $p = 2$. The mesh ratio was chosen as $\lambda = 0.9\lambda_{\max}$ for the linear tests, while $\lambda = 0.66\lambda_{\max}$ for the nonlinear (Burgers) equation. The parameter $\lambda_{\max}$ was computed in order to satisfy stability conditions and it depends on the accuracy of the scheme. We used $\lambda_{\max} = 3/7$ for $m = 3$, while $\lambda_{\max} = 2/7$ for $m = 4$.

Table 5.1 shows the results obtained with Test #1. It can be seen that our third and fourth order schemes are fully third and fourth order accurate, even after a long integration time, in both the $L^1$ and $L^\infty$ norms.

Comparing the magnitudes of the errors with the results obtained with our previous Central ENO scheme in [5], we note that our new third order method yields smaller errors than the third order Central ENO scheme (here the space reconstruction is more accurate). On the other hand, the old $m = 4$ Central ENO scheme is more accurate than the present 4-th order C-WENO scheme. It must be noted, however, that the new scheme is based on second degree polynomials in the reconstruction, instead of the much more costly fourth degree polynomials of the Central ENO $m = 4$ reconstruction.

The results obtained for Test #2 are shown in Table 5.2. Here we observe "super-convergence", a phenomenon already observed in [12]. We can explain this behavior comparing the values of the error in Table 5.2 with those obtained with the linear code (i.e. C-WENO scheme with fixed weights). These results are not shown for brevity, but it appears that the linear C-WENO code has the expected order of accuracy (namely 3 for $m = 3$ and 4 for $m = 4$). The difference with the full non-linear C-WENO code is noticeable only on coarse grids, where the linear C-WENO code is more accurate. On fine grids, both codes yield comparable errors: thus the full C-WENO scheme "catches up" as the grid is refined, and the order of accuracy, for this test problem, appears overestimated.

Linear advection, $u_0 = \sin(\pi x)$

| | $m = 3$ | | $m = 4$ | |
|---|---|---|---|---|
| N | $L^1$ error | $L^1$ order | $L^1$ error | $L^1$ order |
| 20 | 0.8464E-02 | | 0.1224E-01 | |
| 40 | 0.6110E-03 | 3.7921 | 0.4938E-03 | 4.6311 |
| 80 | 0.5485E-04 | 3.4776 | 0.2282E-04 | 4.4357 |
| 160 | 0.6090E-05 | 3.1709 | 0.1261E-05 | 4.1777 |
| 320 | 0.7346E-06 | 3.0514 | 0.7573E-07 | 4.0575 |
| 640 | 0.9091E-07 | 3.0145 | 0.4670E-08 | 4.0192 |
| N | $L^\infty$ error | $L^\infty$ order | $L^\infty$ error | $L^\infty$ order |
| 20 | 0.8653E-02 | | 0.1202E-01 | |
| 40 | 0.5047E-03 | 4.0996 | 0.4674E-03 | 4.6847 |
| 80 | 0.4354E-04 | 3.5350 | 0.1831E-04 | 4.6739 |
| 160 | 0.4793E-05 | 3.1834 | 0.9943E-06 | 4.2027 |
| 320 | 0.5770E-06 | 3.0542 | 0.5962E-07 | 4.0599 |
| 640 | 0.7136E-07 | 3.0154 | 0.3682E-08 | 4.0173 |

Table 5.1: $T = 10, \lambda = 0.9 * \lambda_{\max}$. For $m = 3$, $\lambda_{\max} = 3/7$. For $m = 4$, $\lambda_{\max} = 2/7$. $\epsilon = 10^{-6}, p = 2$.

Linear advection, $u_0 = \sin^4(\pi x)$

| | $m = 3$ | | $m = 4$ | |
|---|---|---|---|---|
| N | $L^1$ error | $L^1$ order | $L^1$ error | $L^1$ order |
| 20 | 0.5514E-01 | | 0.9541E-01 | |
| 40 | 0.6353E-02 | 3.1174 | 0.7728E-02 | 3.6258 |
| 80 | 0.5247E-03 | 3.5981 | 0.8175E-03 | 3.2408 |
| 160 | 0.2940E-04 | 4.1577 | 0.3002E-04 | 4.7673 |
| 320 | 0.2625E-05 | 3.4853 | 0.1130E-05 | 4.7309 |
| 640 | 0.3048E-06 | 3.1063 | 0.6141E-07 | 4.2023 |
| N | $L^\infty$ error | $L^\infty$ order | $L^\infty$ error | $L^\infty$ order |
| 20 | 0.6643E-01 | | 0.1012 | |
| 40 | 0.8657E-02 | 2.9398 | 0.9660E-02 | 3.3896 |
| 80 | 0.9784E-03 | 3.1454 | 0.1510E-02 | 2.6778 |
| 160 | 0.3827E-04 | 4.6760 | 0.7870E-04 | 4.2618 |
| 320 | 0.2669E-05 | 3.8421 | 0.2185E-05 | 5.1704 |
| 640 | 0.2983E-06 | 3.1613 | 0.6022E-07 | 5.1814 |

Table 5.2: $T = 1, \lambda = 0.9 * \lambda_{\max}$. For $m = 3$, $\lambda_{\max} = 3/7$. For $m = 4$, $\lambda_{\max} = 2/7$. $\epsilon = 10^{-6}, p = 2$.

Burgers equation. Solution and weights

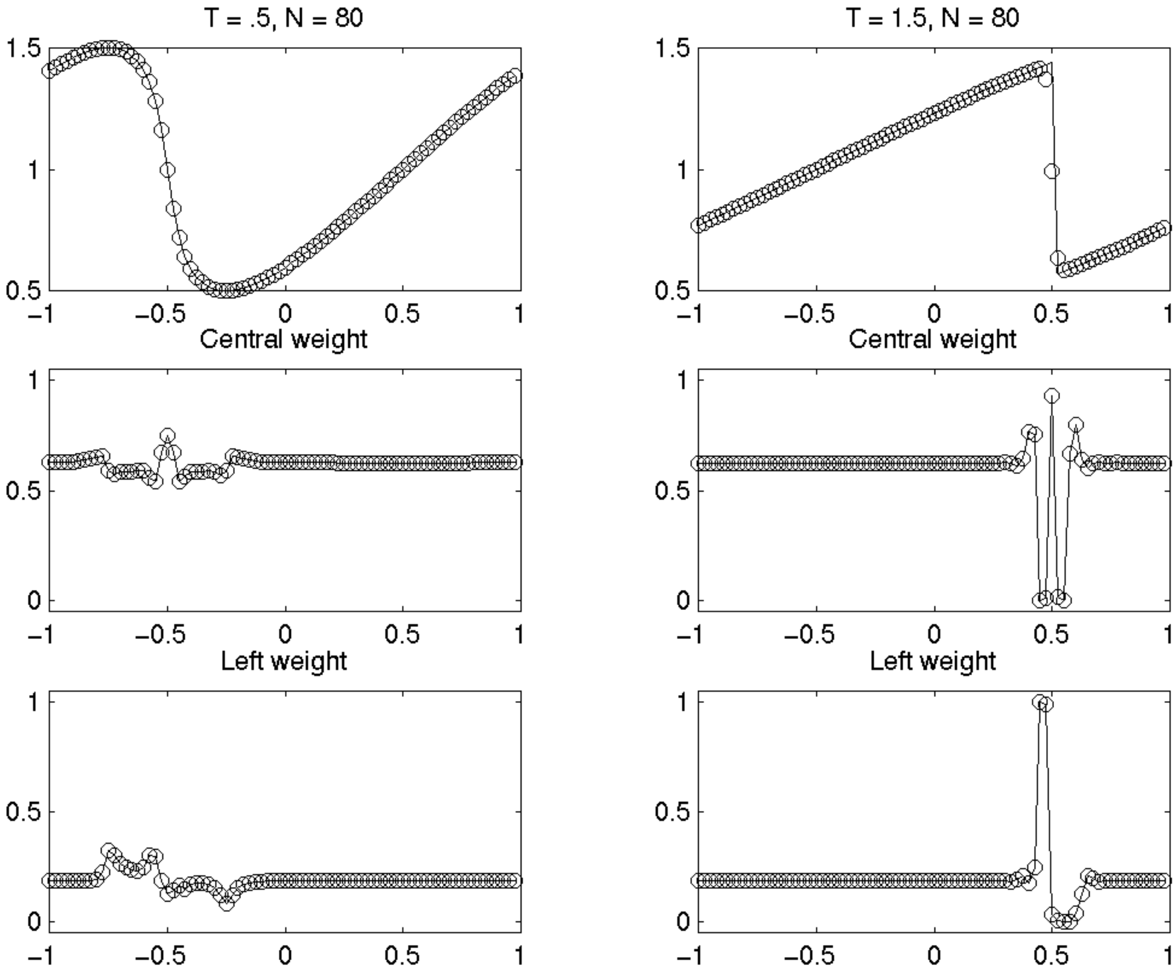


Figure 5.1: Weights computed from cell averages, $m = 4$

Burgers equation, $u_0 = 1 + 1/2\sin(\pi x)$

| | $m=3$ | | $m=4$ | |
|---|---|---|---|---|
| N | $L^1$ error | $L^1$ order | $L^1$ error | $L^1$ order |
| 20 | 0.2010E-02 | | 0.2926E-02 | |
| 40 | 0.1770E-03 | 3.5052 | 0.2459E-03 | 3.5728 |
| 80 | 0.1019E-04 | 4.1187 | 0.1419E-04 | 4.1150 |
| 160 | 0.5285E-06 | 4.2692 | 0.6821E-06 | 4.3787 |
| 320 | 0.3785E-07 | 3.8034 | 0.3227E-07 | 4.4017 |
| 640 | 0.4376E-08 | 3.1126 | 0.1766E-08 | 4.1916 |
| N | $L^\infty$ error | $L^\infty$ order | $L^\infty$ error | $L^\infty$ order |
| 20 | 0.6699E-02 | | 0.9462E-02 | |
| 40 | 0.8913E-03 | 2.9100 | 0.1139E-02 | 3.0549 |
| 80 | 0.5859E-04 | 3.9272 | 0.8631E-04 | 3.7216 |
| 160 | 0.2624E-05 | 4.4809 | 0.4461E-05 | 4.2742 |
| 320 | 0.1542E-06 | 4.0884 | 0.2296E-06 | 4.2800 |
| 640 | 0.1338E-07 | 3.5273 | 0.1269E-07 | 4.1779 |

Table 5.3: $T = 0.33, \lambda = 0.66 * \lambda_{\max}$. For $m = 3$, $\lambda_{\max} = 3/7$. For $m = 4$, $\lambda_{\max} = 2/7$. $\epsilon = 10^{-6}, p = 2$.

We end the accuracy tests with a non linear problem (Test #3), computing the order of accuracy at T=0.33, well before the shock formation time $T = 2/\pi$. The results appear in Table 5.1. Once again, we observe the correct order of accuracy.

The shock-capturing properties of the C-WENO scheme are illustrated in Figure 5.1 for the Burgers equation (Test #3). The pictures on the left refer to the solution before shock formation (T=0.5) and the pictures on the right refer to the solution after shock formation (T=1.5). The bottom part of the picture show the weights $w_k^j$ computed in the reconstruction from cell averages. In particular the central weight corresponds to $w_j^j$, while the left weight corresponds to $w_{j-1}^j$. Before the shock formation, the weights remain close to their equilibrium values (given in Table 1). An abrupt change can be seen after the shock formation. Here the stencils that would yield oscillations are assigned almost a zero weight. Thus the solution is oscillation-free even after the shock forms. The shock transition occurs within two cells.

Figure 5.2 shows the results obtained on Test #4, for $m = 3$ and two grid sizes, $N = 200$ and $N = 400$. The solution has no spurious oscillations: the scheme is able to control oscillations arising from discontinuities in the solution and in its derivatives. The resolution of the contact discontinuities present in the square wave seems better than the analogous results obtained in the ENO and WENO case with Roe flux by Jiang and Shu (see Figure 1 in [12]). Their results are better if they couple their scheme with the artificial compression method by Yang, [33]. To our knowledge, at present, there is no algorithm analogous to Yang's for the central framework. We also note that the resolution of the left peak in the wave train is not as sharp as in the WENO case.

There is a curious lack of symmetry appearing in the right hump of the figure in

Shu's linear test, with $m = 3$

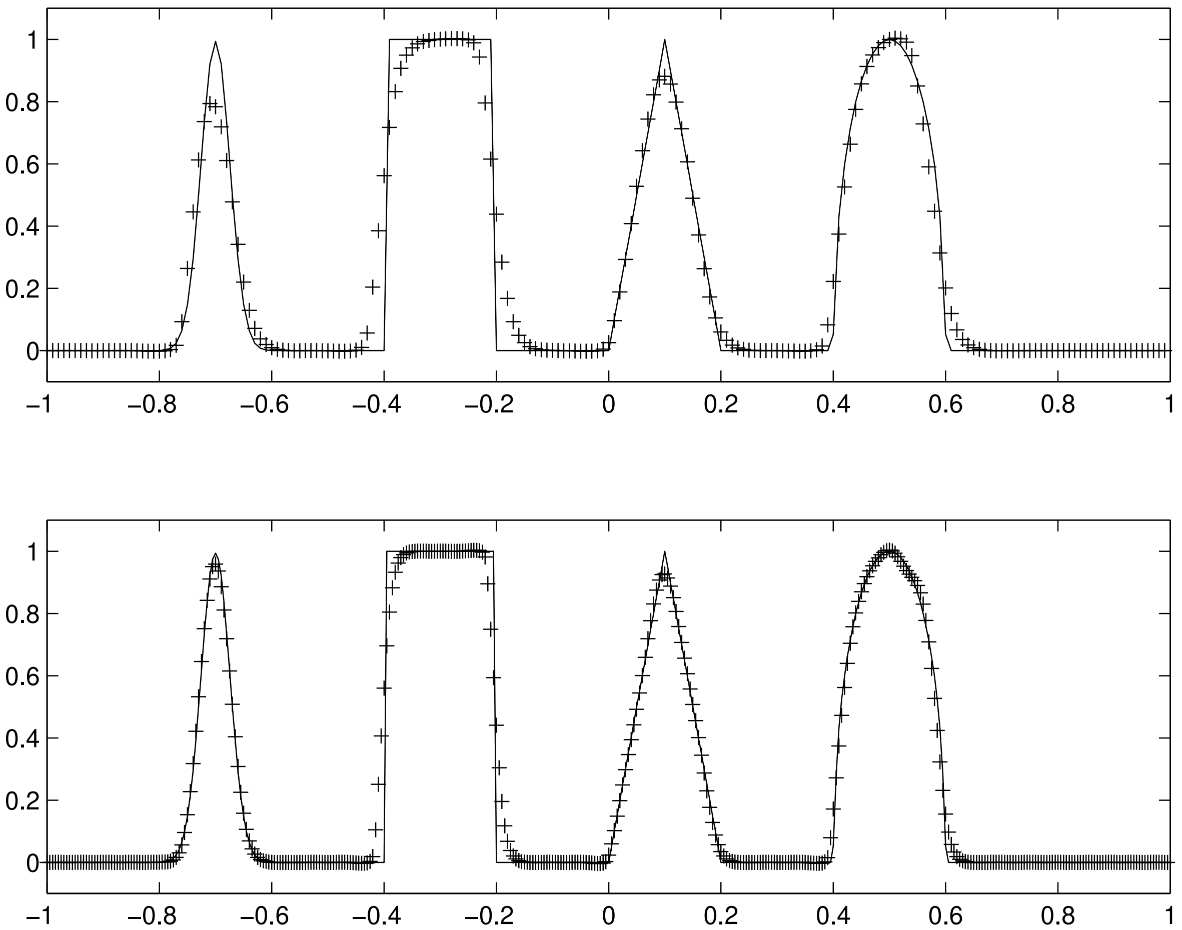


Figure 5.2: N=200 (top), N=400 (bottom)

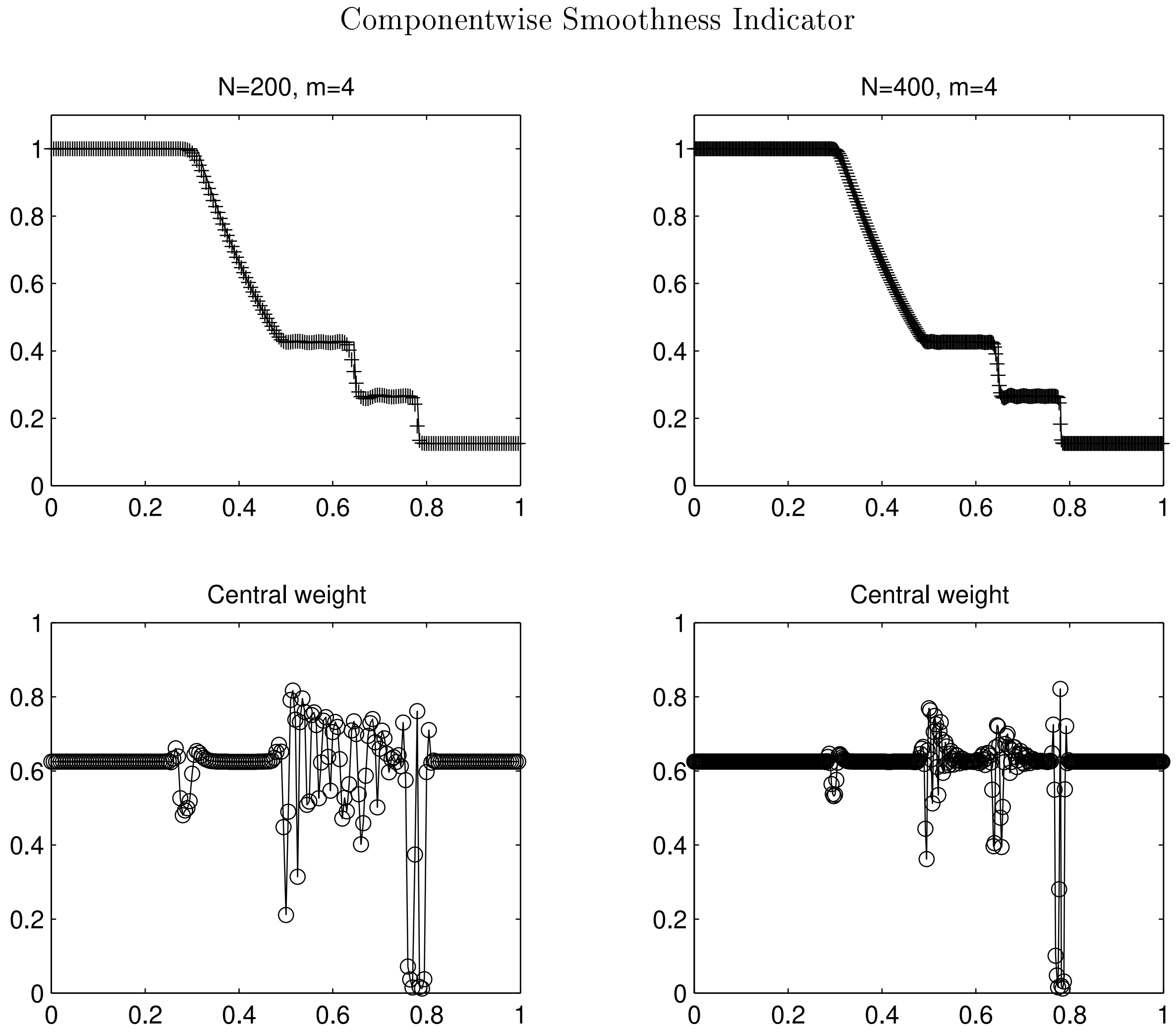


Figure 5.3: Sod's problem: $\lambda = 0.1$, $T = 0.16$. The weight shown is computed in the reconstruction from cell-averages for the density component

Global Smoothness Indicator

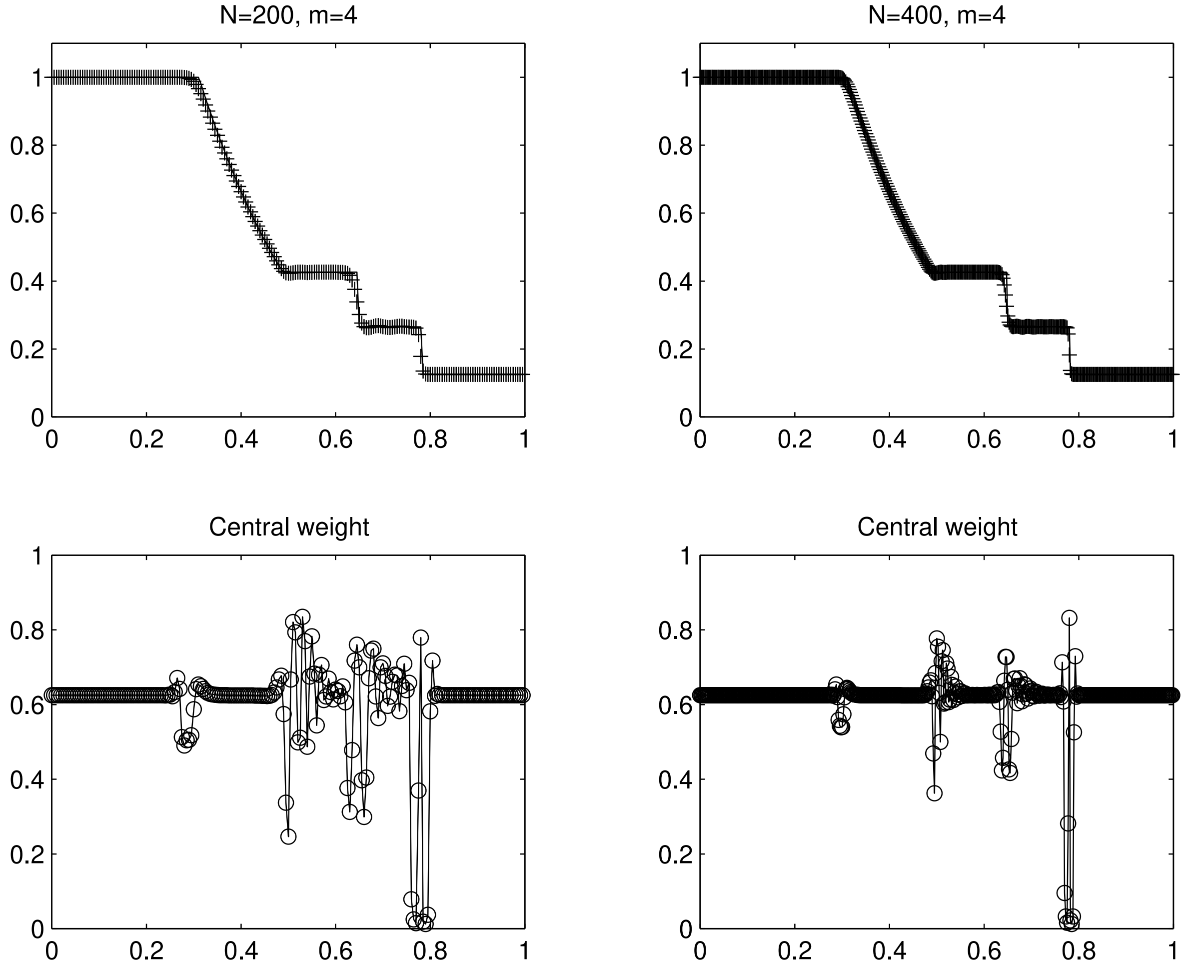


Figure 5.4: Sod's problem: $\lambda = 0.1$, $T = 0.16$. The weight shown is computed once per time step during the reconstruction from cell-averages

the $N = 400$ case, which changes according to the direction of the wind. It is less pronounced in the $N = 800$ case. The same feature appears for $m = 4$. This is the only test problem in which we observed this phenomenon. We did not observe similar phenomenon in our non-linear tests and we leave its investigation for a future study.

We obtain very similar results if we compute the smoothness indicators only once per time step instead of computing them at every stage of the RK scheme, as described in the Remark at the end of /S 4. We do not include these results because they appear identical to the plots we have already shown.

## 5.2 Systems of equations

We apply our schemes to the system of Euler equations of gas dynamics for a polytropic gas with constant $\gamma = 1.4$. The variables $\rho, m, E$, and $p$ below, denote the density, momentum, total energy per unit volume and the pressure, respectively. We consider the following test problems:

**Test #5**. Shock tube problem with Sod's initial data [29].

$$\begin{cases} (\rho_l, m_l, E_l) = (1, 0, 2.5), & x < 0.5, \\ (\rho_r, m_r, E_r) = (0.125, 0, 0.25), & x > 0.5. \end{cases}$$

**Test #6**. Shock tube problem with Lax' initial data [15].

$$\begin{cases} (\rho_l, m_l, E_l) = (0.445, 0.311, 8.928), & x < 0.5, \\ (\rho_r, m_r, E_r) = (0.5, 0, 1.4275), & x > 0.5. \end{cases}$$

In both cases the computational domain is $[0, 1]$; we integrate the equations up to $T = 0.16$, i.e. before the perturbations reach the boundary of the computational region. The computational parameters are $\lambda = 0.1$ and $N = 200$.

**Test #7**. Double blast wave by Woodward and Colella, [32].

$$\begin{cases} (\rho_l, m_l, p_l) = (1, 0, 1000), & x < 0.1, \\ (\rho_c, m_c, p_c) = (1, 0, 0.01), & 0.1 < x < 0.9, \\ (\rho_r, m_r, p_r) = (1, 0, 100), & x > 0.9. \end{cases}$$

In this case, the boundary is reflective at both ends. The equations are integrated up to $T = 0.038$.

We start applying our scheme component by component to each equation of the Euler system, as described in §3. With this strategy the smoothness indicators are computed as in (3.14) using the formulas of the scalar case for each component. The smoothness indicators are computed at each intermediate time level $t^n + c_l\Delta t$ of the Runge Kutta integration, applying again our scheme componentwise. We call this straightforward extension to systems of the scalar formulas for the computation of the weights "Componentwise Smoothness Indicators". In the $m = 4$ case, we need 5 evaluations of the smoothness indicators for each component (1 from cell averages and 1 from point values

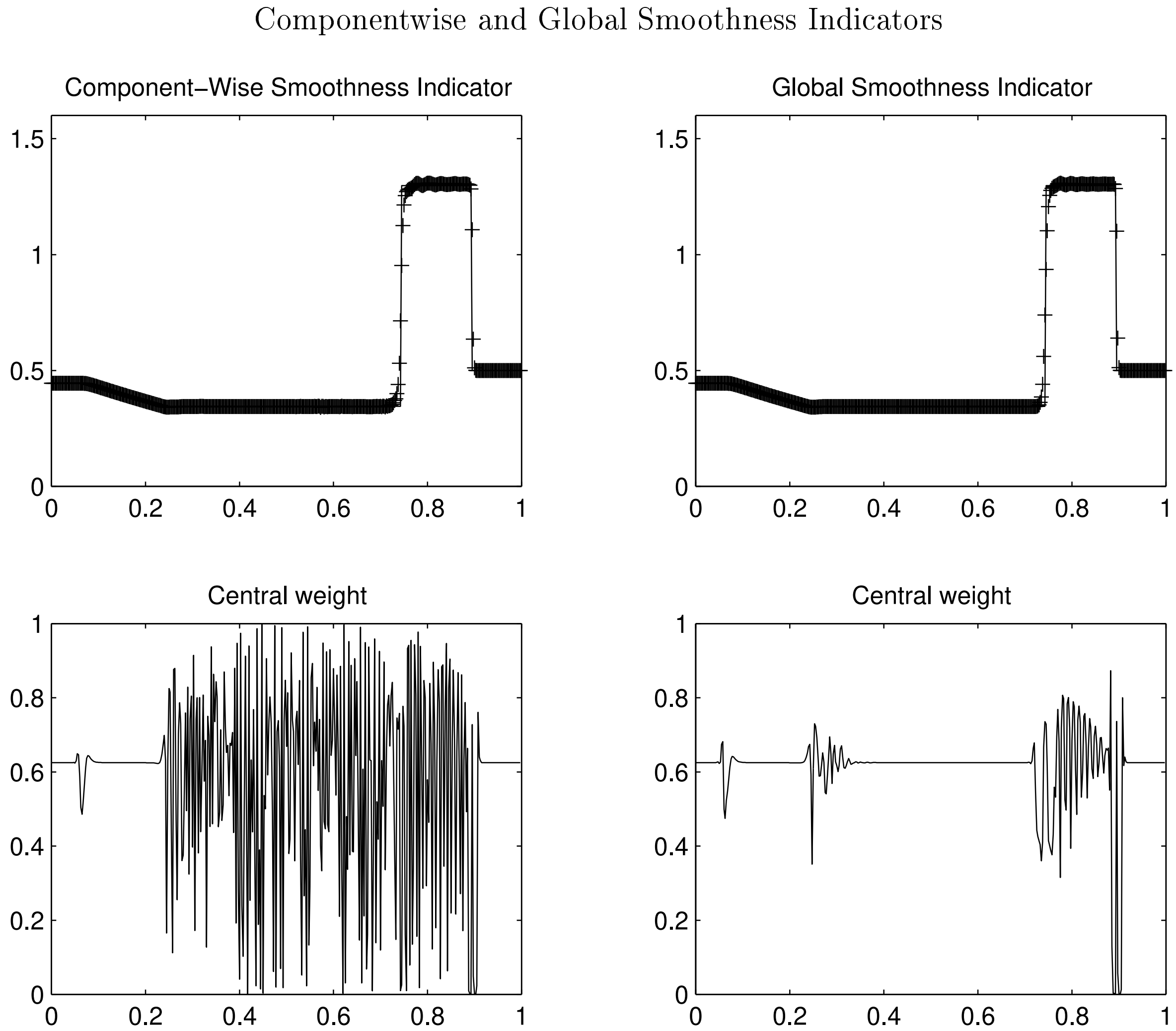


Figure 5.5: Lax' problem: $\lambda = 0.1$, $T = 0.16$, $m = 4$, $N = 400$. The weight shown is computed once per time step during the reconstruction from cell-averages for the density.

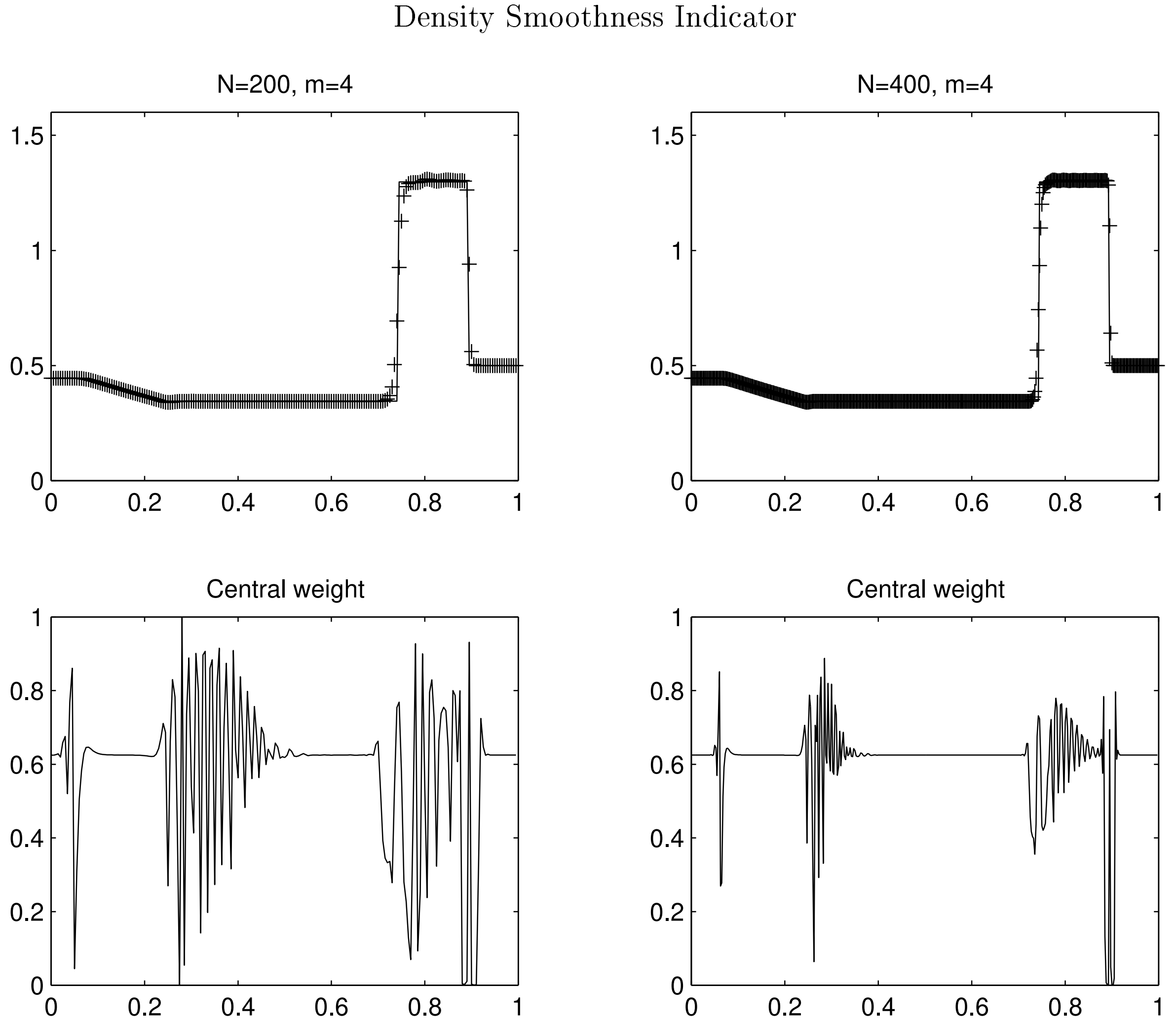


Figure 5.6: Lax' problem: $\lambda = 0.1$, $T = 0.16$, $m = 4$. The weight shown is computed once per time step during the reconstruction from cell-averages for the density.

at each of the 4 Runge-Kutta steps). Thus we have 15 evaluations of the quantities $IS_k^j$ at each time step, in the case of Euler equations.

Applying this scheme with $m = 4$ to Sod's problem with a fixed value of $\lambda = 0.1$, we find the results shown in Figure 5.3, for $N = 200$ and $N = 400$. We note that there are some small amplitude oscillations close to the contact discontinuity. These oscillations, however, are of ENO type: their amplitude decreases as the grid is refined.

The bottom part of the picture shows the behavior of the central weights obtained at the beginning of the last time step in the reconstruction from cell averages for the density. We note that the solution is oscillation free close to the shock, where the smoothness indicators sharply recognize the discontinuity, setting almost to zero the weight of the "bad" stencils. The situation is more confused close to the contact wave. The weights of the "bad" stencils do not go to zero, because the contact wave is spread on several cells (unlike the shock), and therefore it is not recognized as a discontinuity by the scheme.

The behavior of the weights becomes very irregular in the neighborhood of the contact wave, probably because the pressure and the velocity fields are continuous across the contact wave, and therefore each component selects a different stencil. This suggests that the scheme might improve if all components feel the presence of a discontinuity through their smoothness indicators. This can obtained, e.g., by using one smoothness indicator for all the components:

$$IS_k^j = \frac{1}{d}\sum_{r=1}^{d} \frac{1}{||\bar{u}_r||_2} \left( \sum_{l=1}^{2} \int_{x_{j-1/2}}^{x_{j+1/2}} h^{2l-1} \left(P_{k,r}^{(l)}\right)^2 \, dx \right), \quad k = j-1, j, j+1. \tag{5.1}$$

Here $d$ is the number of equations, and $P_{k,r}$ denotes the $k$-th polynomial for the $r$-th component. The quantity $||\bar{u}_r||_2$ is a scaling factor, and it is defined as the $L^2$ norm of the cell averages of the $r$-th component of $u$, namely:

$$||\bar{u}_r||_2 = \left( \sum_{\mathrm{all}\, j} |\bar{u}_{j,r}|^2 h \right)^{1/2}.$$

The integral in (5.1) can be exactly integrated (see (3.14)).

We find that this strategy is effective even if the quantities $IS_k^j$ are computed at the beginning of the time step in the reconstruction from cell averages, and are not changed at every level of the RK scheme. See the remark at the end of /S 4. This strategy will be called "Global smoothness indicators": its results on Sod's problem are shown in Figure 5.4. We note a slight improvement on the control of the spurious oscillations, and the oscillations in the weights are more localized. A more marked improvement can be observed on Lax' problem (see Figure 5.5).

To summarize, the "Global smoothness indicators" strategy is much less expensive and gives better results than the "Componentwise smoothness indicators" strategy.

We can further improve our results if we take into account the particular structure of the system. Since the density jumps at both shocks and contact waves, we can use the density alone to compute the smoothness indicators for all components.

Again, we find that it is enough to compute the smoothness indicators only once per time step. This strategy is slightly more diffusive than the previous one, because the biased stencils will be chosen more often. The results are slightly better than the "Global smoothness indicators" strategy and are omitted for brevity. The drawback of this approach is that we use information about the particular structure of the system.

The application of the three strategies that we are proposing to Lax' test problem (Test #6) are shown in Figure 5.5 and Figure 5.6. The first figure shows the density and the central weight computed for the density component for the componentwise and the Global strategies, for $m = 4$ and $N = 400$. The second figure shows the results obtained with the density strategy with $m = 4$ and two grid sizes, $N = 200$ and $N = 400$. From Figure 5.6, we see that the amplitude of the spurious wiggles near the contact discontinuity decays as the grid is refined.

In these pictures we see similar behavior as in Sod's problem. In all three strategies, shock waves are well resolved and oscillation free. Clearly, the Global strategy produces

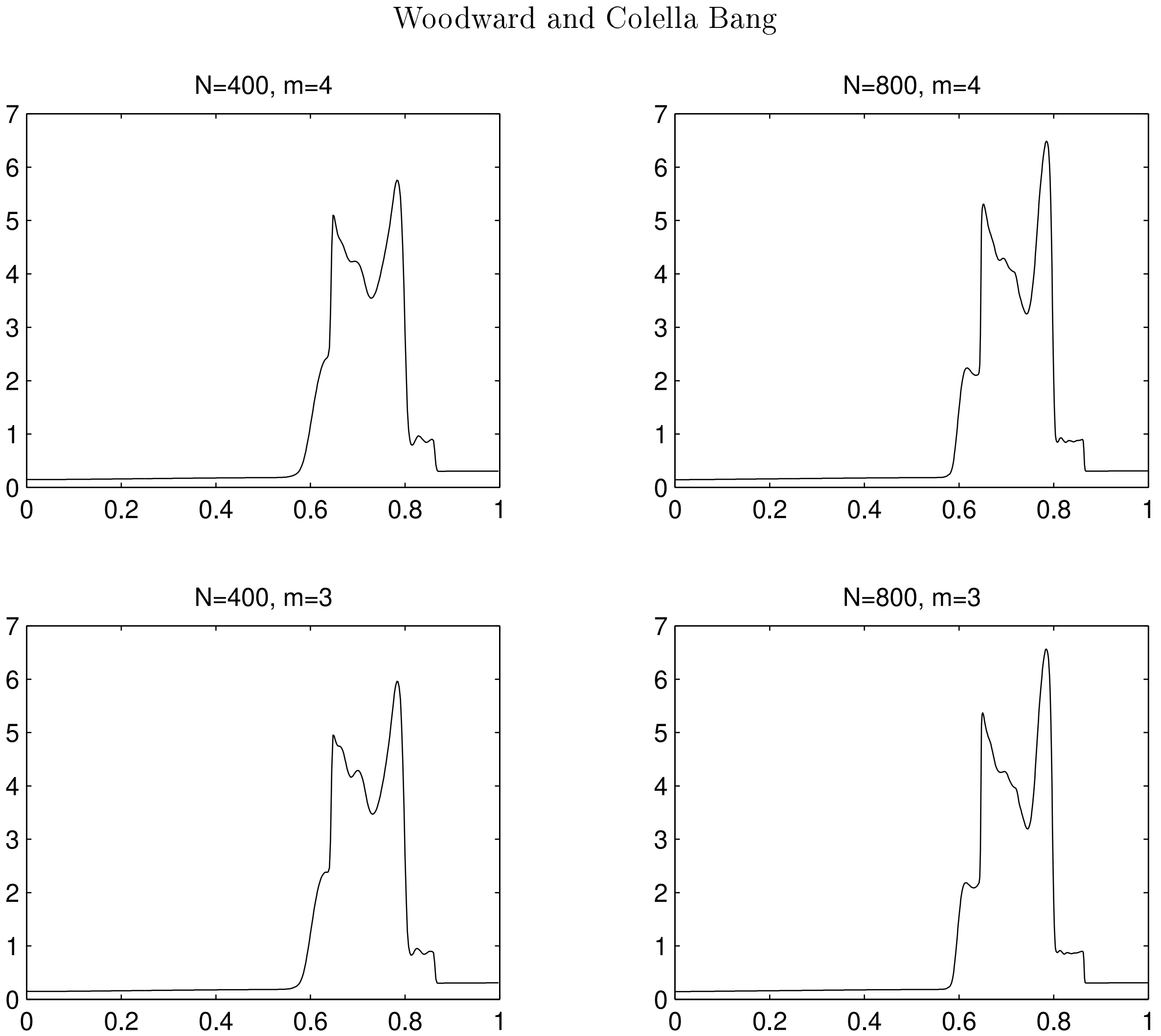


Figure 5.7: Global Smoothness Indicator, $T = 0.038$. The weight shown is computed once per time step during the reconstruction from cell-averages for the density

Woodward and Colella Bang

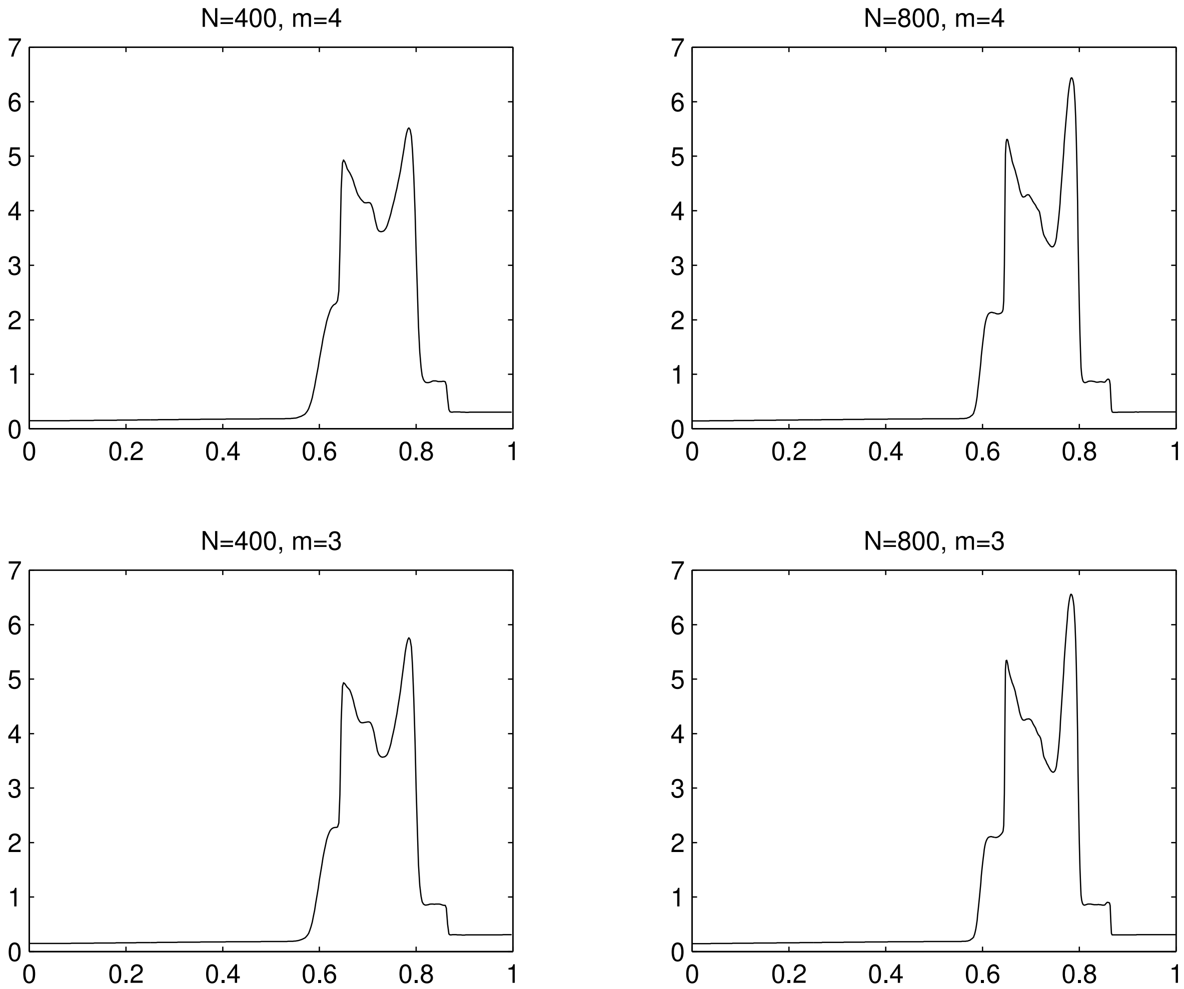


Figure 5.8: Density Smoothness Indicator, $T = 0.038$. The weight shown is computed once per time step during the reconstruction from cell-averages for the density

a more reasonable selection of the weights, compared with the Componentwise approach. This behavior results in smaller wiggles in the numerical solution. It is remarkable that even the wild weights produced by the componentwise strategy result in a reasonable solution. We believe that this fact illustrates the robustness of our scheme.

We end this section applying our scheme to Test #8: the blast wave problem of Woodward and Colella, [32]. Since in this problem the characteristic speeds change wildly with time, we used an adaptive evaluation of the time step, namely:

$$\Delta t = C\frac{h}{\max_j(c_j + |u_j|)}, \quad C = 0.9\lambda_{\max},$$

where $c_j$ and $u_j$ are the local sound speed and velocity respectively. In our tests $\lambda_{\max}$ depends on the accuracy of the scheme, as already discussed for the scalar case.

We show the density component of the solution at $T = 0.038$ in Figure 5.7 and Figure 5.8, for the Global and the Density strategies respectively. The Global strategy is clearly slightly less dissipative, especially on the coarse grid, but there are small ENO-type wiggles. In both cases, the two peaks in the density are very well resolved (compare with WENO schemes in [12]).

# 6 Conclusions

We have presented new third and fourth order central schemes which are based on a new central WENO (CWENO) reconstruction. The time marching scheme is constructed through Runge-Kutta integration with Natural Continuous Extensions.

In particular, we develop particularly simple and efficient techniques for the evaluation of the smoothness indicators in the case of systems.

Our results show that our schemes are fast and robust tools for the integration of general systems of conservation laws.

These new schemes require no approximate Riemann solvers, no projection along characteristic directions, no fancy splitting of the flux function in upwind and downwind directions, and finally no exact or approximate evaluation of eigenvalues and eigenvectors of the Jacobian of the flux. The CWENO scheme with the Global strategy for the computation of the weights really goes towards the construction of black-box schemes for hyperbolic systems: only an estimate of the characteristic speeds is needed to choose a time step satisfying the CFL condition.

A 2D extension of these schemes will be presented in a following paper [19].

# References

[1] Arminjon P., Stanescu D., Viallon M.-C., *A Two-Dimensional Finite Volume Extension of the Lax-Friedrichs and Nessyahu-Tadmor Schemes for Compressible Flow*, Proc. 6th. Int. Symp. on CFD, Lake Tahoe, 1995, M. Hafez and K. Oshima, editors, Vol. IV, pp.7-14.

[2] Arminjon P., Viallon M.-C., *Généralisation du Schéma de Nessyahu-Tadmor pour Une Équation Hyperbolique à Deux Dimensions D'espace*, C.R. Acad. Sci. Paris, **t. 320** , série I. (1995), pp.85-88.

[3] Arminjon P., Viallon M.-C. and Madrane A., *A Finite Volume Extension of the Lax-Friedrichs and Nessyahu-Tadmor Schemes for Conservation Laws on Unstructured Grids*, IJCFD, **9**, (1997), pp.1-22.

[4] Arminjon P., Viallon M.-C., Madrane A., Kaddouri L., *Discontinuous Finite Elements and Finite Volume Versions of the Lax-Friedrichs and Nessyahu-Tadmor Schemes for Compressible Flows on Unstructured Grids*, Computational Fluid Dynamics Review, Wiley, 1997, M. Hafez and K. Oshima, editors.

[5] Bianco F., Puppo G., Russo G., *High Order Central Schemes for Hyperbolic Systems of Conservation Laws*, SIAM J. Sci. Comp., to appear.

[6] Friedrichs K. O., Lax P. D., *Systems of Conservation Equations with a Convex Extension*, Proc. Nat. Acad. Sci., **68**, (1971), pp.1686-1688.

[7] Godlewski E., Raviart P.-A., *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer, New York, 1996.

[8] Harten A., *High Resolution Schemes for Hyperbolic Conservation Laws*, JCP, **49**, (1983), pp.357-393.

[9] Harten A., Engquist B., Osher S., Chakravarthy S., *Uniformly High Order Accurate Essentially Non-oscillatory Schemes III*, JCP, **71**, (1987), pp.231-303.

[10] Huynh H. T., *A Piecewise-parabolic Dual-mesh Method for the Euler Equations*, AIAA-95-1739-CP, The 12th AIAA CFD Conf., 1995.

[11] Jiang G.-S., Levy D., Lin C.-T., Osher S., Tadmor E. *High-Resolution Non-Oscillatory Central Schemes with Non-Staggered Grids for Hyperbolic Conservation Laws*, SINUM, to appear.

[12] Jiang G.-S., Shu C.-W., *Efficient Implementation of Weighted ENO Schemes*, JCP, **126**, (1996), pp.202-228.

[13] Jiang G.-S., Tadmor E., *Nonoscillatory Central Schemes for Multidimensional Hyperbolic Conservation Laws*, SIAM J. Sci. Comp., to appear.

[14] Jin S., Xin Z.-P., *The Relaxation Schemes for Systems of Conservation Laws in Arbitrary Space Dimensions*, CPAM, **48**, (1995), pp.235-277.

[15] Lax P. D., *Weak Solutions of Non-Linear Hyperbolic Equations and Their Numerical Computation*, CPAM, **7**, (1954), pp.159-193.

[16] van Leer B., *Towards the Ultimate Conservative Difference Scheme, V. A Second-Order Sequel to Godunov's Method*, JCP, **32**, (1979), pp.101-136.

[17] LeVeque R. J., *Numerical Methods for Conservation Laws*, Lectures in Mathematics, Birkhauser Verlag, Basel, 1992.

[18] Levy D., *A Third-order 2D Central Schemes for Conservation Laws*, INRIA School on Hyperbolic Systems, 1998.

[19] Levy D., Puppo G., Russo G. *Central Weno Schemes for Multi-Dimensional Hyperbolic Systems of Conservation Laws*, in preparation.

[20] Levy D., Tadmor E., *Non-oscillatory Central Schemes for the Incompressible 2-D Euler Equations*, Math. Res. Let., **4** (1997), pp.1-20.

[21] Liu X.-D., Osher S., *Nonoscillatory High Order Accurate Self-Similar Maximum Principle Satisfying Shock Capturing Schemes I*, SINUM, **33**, no. 2 (1996), pp.760-779.

[22] Liu X.-D., Osher S., Chan T., *Weighted Essentially Non-oscillatory Schemes*, JCP, **115**, (1994), pp.200-212.

[23] Liu X.-D., Tadmor E., *Third Order Nonoscillatory Central Scheme for Hyperbolic Conservation Laws*, Numer. Math., to appear.

[24] Nessyahu H., Tadmor E., *Non-oscillatory Central Differencing for Hyperbolic Conservation Laws*, JCP, **87**, no. 2 (1990), pp.408-463.

[25] Roe P. L., *Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes*, JCP, **43**, (1981), pp.357-372.

[26] Sanders R., Weiser A., *A High Resolution Staggered Mesh Approach for Nonlinear Hyperbolic Systems of Conservation Laws*, JCP, **1010**, (1992), pp.314-329.

[27] Shu C.-W., *Numerical experiments on the accuracy of ENO and modified ENO schemes*, J. Sci. Comp., **5**, vol. 2, (1990), pp.127-149.

[28] Shu C.-W., Osher S., *Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II*, JCP, **83**, (1989), pp.32-78.

[29] Sod G., *A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws*, JCP, **22**, (1978), pp.1-31.

[30] Sweby P. K., *High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws*, SINUM, **21**, no. 5 (1984), pp.995-1011.

[31] Tadmor E., *Approximate Solutions of Nonlinear Conservation Laws*, CIME Lecture notes, 1997, UCLA CAM Report 97-51.

[32] Woodward P., Colella P., *The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks*, JCP, **54**, (1984), pp.115-173.

[33] Yang H., JCP, **89**, (1990), pp. 125-.

[34] Zennaro M., *Natural Continuous Extensions of Runge-Kutta Methods*, Math. Comp., **46**, (1986), pp.119-133.