JOURNAL OF COMPUTATIONAL PHYSICS **144,** 194–212 (1998)
ARTICLE NO. CP985988

Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids

Oliver Friedrich

Institut für Angewandte Mathematik, Universität Hamburg, Bundesstrasse 55, D-20146 Hamburg

Received November 18, 1997

In this paper the weighted ENO (essentially non-oscillatory) scheme developed for the one-dimensional case by Liu, Osher, and Chan is applied to the case of unstructured triangular grids in two space dimensions. Ideas from Jiang and Shu, especially their new way of smoothness measuring, are considered. As a starting point for the unstructured case we use an ENO scheme like the one introduced by Abgrall. Beside the application of the weighted ENO ideas the whole reconstruction algorithm is analyzed and described in detail. Here we also concentrate on technical problems and their solution. Finally, some applications are given to demonstrate the accuracy and robustness of the resulting new method. The whole reconstruction algorithm described here can be applied to any kind of data on triangular unstructured grids, although it is used in the framework of compressible flow computation in this paper only.

1. INTRODUCTION

During the last years the class of essentially non-oscillatory (ENO) schemes introduced by Harten, Osher, and others has been transferred to the case of compressible flow computations on unstructured triangular grids in two space dimensions [HC91, A94, S97].

When implementing the method of Abgrall we noticed that there are some technical problems that should be worked out and written down in more detail than has been done in the past. Furthermore we noticed that the second order method worked quite well but the third order method did not show the accuracy we expected. Especially in smooth regions on relatively coarse grids the quality of the obtained results was poor.

One workaround known so far is to use ENO only near discontinuities and to use a central reconstruction technique in smooth regions. Another possibility is demonstrated in [LOC94] for the one-dimensional case. There the main idea of the ENO method is modified. Instead of computing a set of several reconstruction polynomials $p_1, \ldots, p_m$ and

0021-9991/98 $25.00

then digitally selecting that p_j with the minimal oscillation, a new polynomial p is computed as a weighted sum of all polynomials: $p = \sum_{i=1}^{m} \omega_i p_i$. The weights ω_i are chosen depending on the oscillation of the p_i such that a ω_i is low if the oscillation of p_i is high while ω_i is of order one if p_i has low oscillation. The resulting new scheme is called a *weighted ENO* scheme while for differentiation the original ENO scheme is called the *digital ENO* scheme. By setting exactly one weight ω_j to one and all other weights to zero the digital ENO scheme can be interpreted as a special case of the weighted ENO scheme.

We have applied that new technique to the case of flows on unstructured triangular grids in two dimensions.

We will first describe our method for compressible flow computation in short and will then concentrate on the reconstruction problem: In Section 2 some fundamentals and notations concerning reconstruction of mean values are given. In Section 3 the whole reconstruction algorithm is described in detail. Some numerical applications can be found in Section 4.

1.1. *Governing Equations*

For any bounded control volume $\Omega \subset \mathbb{R}^2$ the Euler equations for compressible flows in two dimensions can be written in integral form as

$$\frac{d}{dt} \int_{\Omega} Q(x,t)\, dx = -\int_{\partial\Omega} F(Q(x,t)) \cdot n\, ds, \tag{1}$$

where $n = (n_1, n_2)$ is the outer unit normal to the boundary of Ω denoted as $\partial\Omega$. The vector of conserved variables $Q = Q(x,t)$ and the convective fluxes $F(Q) = (F_1(Q), F_2(Q))$ are given by

$$Q = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{pmatrix},$$

$$F_1(Q) = \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ (\rho E + p) v_1 \end{pmatrix},$$

$$F_2(Q) = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ (\rho E + p) v_2 \end{pmatrix}.$$

The system is closed by the equation of state,

$$p = (\gamma - 1)\rho\big(E - 0.5\big(v_1^2 + v_2^2\big)\big),$$

where $\gamma = 1.4$ for all our computations.

1.2. *Finite Volume Formulation*

Computational cells are constructed from a triangulation of the computational domain by barycentric subdivision. This means that a cell Ω_k is constructed around each point P_k

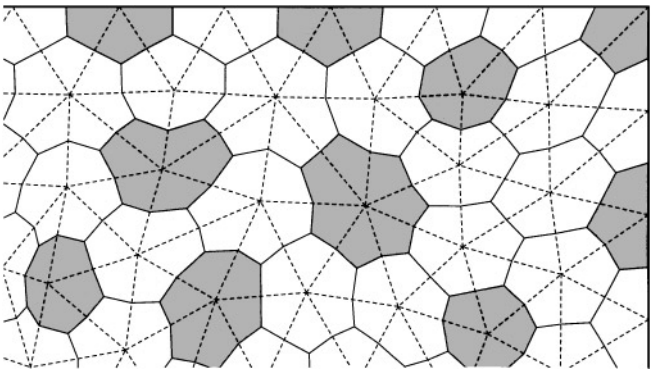

FIG. 1. Section of a triangulation (dashed) and corresponding cells (solid, some shaded).

of the triangulation by connecting the barycenters (centers of mass) of the triangles having P_k in common and the midpoints of the edges between these triangles (see Fig. 1). Note that there is a one-to-one relationship between the grid points of the triangulation and the cells of the so constructed computational grid.

The cells Ω_k are used as discrete control volumes. Then the semi-discrete finite volume formulation of Eq. (1) is given by

$$\frac{d}{dt}\bar{Q}_k(t) = -\frac{1}{|\Omega_k|}\int_{\partial\Omega_k} F(Q(x,t))\cdot n\, ds \tag{2}$$

with the short form

$$\frac{d}{dt}\bar{Q}_k(t) = \mathcal{L}_k(Q(x,t)),$$

where

$$\bar{Q}_k(t) := \frac{1}{|\Omega_k|}\int_{\Omega_k} Q(x,t)\, dx$$

is the (spatial) mean value of $Q(x,t)$ at time t over Ω_k and $|\Omega_k|$ is a short notation for the area of Ω_k.

1.3. *Spatial Discretization*

A cell Ω_k is polygonally bounded. This means that the boundary of Ω_k is given by a finite number of line segments:

$$\partial\Omega_k = \bigcup_j \Gamma_j.$$

Thus, the boundary integral from Eq. (2) can be decomposed into

$$\int_{\partial\Omega_k} F(Q(x,t))\cdot n\, ds = \sum_j \int_{\Gamma_j} F(Q(x,t))\cdot n\, ds. \tag{3}$$

Note that n is constant on each Γ_j. The line integral from (3) is discretized using a p point

Gaussian integration formula. We used $p = 2$ for all our computations. Let $G_1, \ldots, G_p$ and $w_1, \ldots, w_p$ be the Gaussian points and weights on line segment Γ_j.

Given the end points P_1 and P_2 of Γ_j for $p = 1$ the Gaussian point is just the mid-point of Γ_j: $G_1 = 1/2(P_1 + P_2)$ and the weight w_1 is one. For $p = 2$ define $\alpha := (\sqrt{3}+1)/(2\sqrt{3})$. Then $G_1 = \alpha P_1 + (1-\alpha)P_2$, $G_2 = (1-\alpha)P_1 + \alpha P_2$ and $w_1 = w_2 = 1/2$.

Using the integration formula we get the following approximation:

$$\int_{\Gamma_j} F(Q(x,t)) \cdot n \, ds \approx |\Gamma_j| \sum_{l=1}^{p} w_l F(Q(G_l,t)) \cdot n. \tag{4}$$

The finite volume method will compute approximations for the spatial mean values $\bar{Q}_k(t)$. From these mean values we reconstruct polynomial representations for the *primitive variables* ρ, v_1, v_2, p on each Ω_k. The reconstruction procedure is the main topic of this paper and is discussed later in detail. Let us now assume that on each cell Ω_k we have a polynomial representation of the primitive variables. From these polynomials we can now deduce a functional representation $\tilde{Q}_k(x,t)$ on each Ω_k.

The reconstruction procedure does not force $\tilde{Q}_k(x,t)$ to be continuous over the cell boundaries. Hence, the boundary integrals cannot be computed or approximated directly as described above. Instead, a numerical flux function $\tilde{F}$ is used to approximate the values on the cell boundaries. We use the numerical flux of Osher and Solomon that is described in [OS82]. Let $\Omega_{\mathcal{N}(j)}$ be the cell touching Ω_k at line segment Γ_j. Then the flux evaluation in the right hand side of Eq. (4) is replaced as

$$F(Q(G_l,t)) \cdot n \approx \tilde{F}(\tilde{Q}_k(G_l,t), \tilde{Q}_{\mathcal{N}(j)}(G_l,t); n).$$

Taking together the above discretizations and approximations results in spatially discretized equations for (2):

$$\frac{d}{dt}\bar{Q}_k(t) = \tilde{\mathcal{L}}_k(\bar{Q}_k(t)). \tag{5}$$

1.4. *Temporal Discretization*

Equation (5) defines a finite set of ordinary differential equations. These are discretized using the kth order version of the TVD Runge–Kutta scheme described in [SO88]. Let the time step size be given by Δt. For convenience we define $\bar{Q}_k^{(0)} := \bar{Q}_k(t)$ and $\bar{Q}_k(t+\Delta t) := \bar{Q}_k^{(k)}$.

For $k = 2$ the scheme is given by

$$\begin{aligned}
\bar{Q}_k^{(1)} &:= \bar{Q}_k^{(0)} + \Delta t \tilde{\mathcal{L}}_k(\bar{Q}_k^{(0)}),\\
\bar{Q}_k^{(2)} &:= \frac{1}{2}\bar{Q}_k^{(0)} + \frac{1}{2}\bar{Q}_k^{(1)} + \frac{1}{2}\Delta t \tilde{\mathcal{L}}_k(\bar{Q}_k^{(1)}).
\end{aligned}$$

The scheme for $k = 3$ is

$$\begin{aligned}
\bar{Q}_k^{(1)} &:= \bar{Q}_k^{(0)} + \Delta t \tilde{\mathcal{L}}_k(\bar{Q}_k^{(0)}),\\
\bar{Q}_k^{(2)} &:= \frac{3}{4}\bar{Q}_k^{(0)} + \frac{1}{4}\bar{Q}_k^{(1)} + \frac{1}{4}\Delta t \tilde{\mathcal{L}}_k(\bar{Q}_k^{(1)}),\\
\bar{Q}_k^{(3)} &:= \frac{1}{3}\bar{Q}_k^{(0)} + \frac{2}{3}\bar{Q}_k^{(2)} + \frac{2}{3}\Delta t \tilde{\mathcal{L}}_k(\bar{Q}_k^{(2)}).
\end{aligned}$$

2. RECONSTRUCTION OF MEAN VALUES

For the remainder of the paper we will treat the following problem: *Given* (*spatial*) *mean values* $\bar{u}_k$ *of a function* $u(x)$ *for each cell* Ω_k *and a positive integer number* n, *for each* Ω_l *find a polynomial* $p(x)$ *of degree* n *that is a good approximation to* $u(x)$ *on cell* Ω_l *and which has the mean value* $\bar{u}_l$ *on* Ω_l. For our flow solver this problem has to be solved for $\bar{u}_k$ replaced by the mean values of the primitive variables ρ, v_1, v_2, p on Ω_k.

For convenience some notations will be fixed. The (spatial) mean value of a function $f(x)$ on a cell Ω_l is written as

$$\langle f(x)\rangle_{\Omega_l} := \frac{1}{|\Omega_l|}\int_{\Omega_l} f(x)\,dx. \tag{6}$$

To obtain a short notation for a polynomial in two space dimensions we use multi-indices. For $\alpha = (\alpha_1, \alpha_2)$, $\alpha_i \in \{0, 1, 2, \ldots\}$, and $x = (x_1, x_2) \in \mathbb{R}^2$ let $|\alpha| := \alpha_1 + \alpha_2$ and $x^\alpha = x^{(\alpha_1,\alpha_2)} := x_1^{\alpha_1} x_2^{\alpha_2}$.

The number of all α with $|\alpha| \le n$ is denoted by $N(n)$. It is $N(0) = 1$, $N(1) = 3$, and $N(2) = 6$. We assume that the set $\{\alpha : |\alpha| \le n\}$ is ordered such that we can use multi-indices in a natural way as indices for vectors and matrices. For $n = 1$ we use $\{(0, 0), (1, 0), (0, 1)\}$; $\{(1, 1), (2, 0), (0, 2)\}$ is appended for $n = 2$.

Let b_l denote the barycenter of cell Ω_l. For given n the following standard expansion of a polynomial $p(x)$ for Ω_l is taken,

$$p(x) = \sum_{|\alpha|\le n} a_\alpha (x - b_l)^\alpha, \tag{7}$$

where the $a_\alpha \in \mathbb{R}$, $|\alpha| \le n$ are the unknown coefficients.

To fix the a_α at least $N(n)$ conditions are required. These are taken to be interpolation conditions for the given mean values. Let us for the sake of simplicity reconstruct a polynomial for cell Ω_1. We take additional $N(n) - 1$ cells that shall again for the sake of simplicity be numbered Ω_2 to $\Omega_{N(n)}$. The resulting set of cells

$$S := \{\Omega_1, \Omega_2, \ldots, \Omega_{N(n)}\}$$

is called a *stencil*.

Then $N(n)$ interpolation conditions are given by

$$\begin{aligned}
\langle p(x)\rangle_{\Omega_1} &= \bar{u}_1,\\
\langle p(x)\rangle_{\Omega_2} &= \bar{u}_2,\\
&\cdots\\
\langle p(x)\rangle_{\Omega_{N(n)}} &= \bar{u}_{N(n)}.
\end{aligned}$$

It is obvious that the above conditions form a linear system of $N(n)$ equations for the $N(n)$ unknowns a_α, $|\alpha| \le n$ if the $\langle p(x)\rangle_{\Omega_k}$ are expanded:

$$\langle p(x)\rangle_{\Omega_k} = \left\langle \sum_{|\alpha|\le n} a_\alpha (x - b_1)^\alpha \right\rangle_{\Omega_k} = \sum_{|\alpha|\le n} a_\alpha \left\langle (x - b_1)^\alpha \right\rangle_{\Omega_k}.$$

This system can be written in a compact matrix form as

$$\mathcal{A}a = \bar{u},$$

with

$$\mathcal{A}_{k,\alpha} = \left\langle (x - b_1)^\alpha \right\rangle_{\Omega_k} = \frac{1}{|\Omega_k|} \int_{\Omega_k} (x - b_1)^\alpha \, dx, \qquad 1 \le k \le N(n),\ |\alpha| \le n,$$

where a is the vector of the $N(n)$ unknowns a_α, $|\alpha| \le n$, and $\bar{u} = (\bar{u}_1, \ldots, \bar{u}_{N(n)})$ is the vector of the given mean values.

The stencil S is called *admissible* if the system is uniquely solvable, which is equivalent to the condition that the matrix $\mathcal{A}$ is invertible.

3. THE RECONSTRUCTION ALGORITHM IN DETAIL

Let us now come to the complete description of the reconstruction algorithm that is the key component of our weighted ENO scheme.

A reconstruction polynomial for each cell Ω_k has to be computed, but the scheme is local in the sense that the reconstruction polynomial for cell Ω_l can be computed independently from the reconstruction polynomial for cell Ω_k. For this reason we describe the reconstruction algorithm for only one cell Ω_l.

In principle the reconstruction algorithm is quite clear:

1. Find some admissible stencils $S_1, \ldots, S_m$.
2. For each of these stencils S_i compute the reconstruction polynomial p_i from the $\bar{u}_k$.
3. For each of the p_i compute the oscillation.
4. Depending on the oscillations of the p_i compute non-negative weights ω_i such that the sum of the ω_i is one.
5. Compute the reconstruction polynomial p as the weighted sum of the p_i: $p := \sum_{i=1}^m \omega_i p_i$.

Note that only the last step is trivial. In the following subsections we will have a closer look at the other steps of the algorithm.

3.1. Selection of Admissible Stencils

3.1.1. *The Admissibility Condition*

Unfortunately we do not know if there is a (simple) geometrical property giving us the admissibility of a stencil for a polynomial degree greater than one. So we first select stencils S_i which are probably inadmissible. Later, during the computation of the polynomials p_i, the inadmissibility is detected and those stencils are skipped.

3.1.2. *Selection of Stencils*

After some testing it can be said that stencil selection is a very critical part of the reconstruction algorithm; some more or less contradictory aspects have to be taken into account:

- The number of stencils should be low to keep the computational costs low.
- To obtain high accuracy and stability in smooth regions it is necessary that the stencils have a small diameter and that they are well centered with respect to Ω_l.
- On the other hand ENO methods are based on the idea that in case of non-smooth data $\bar{u}_k$ one-sided stencils are selected to avoid interpolation across discontinuities.

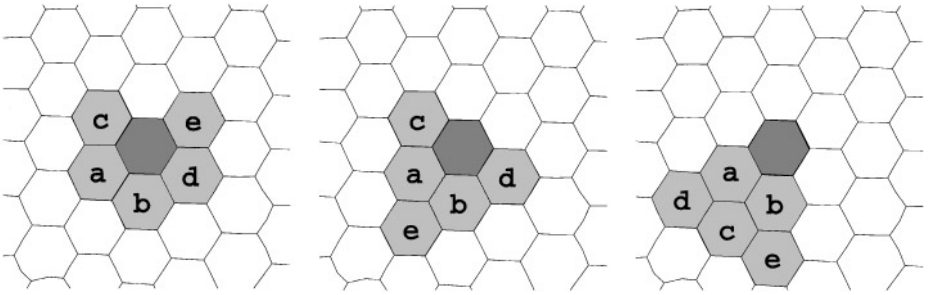

FIG. 2. Different types of stencils for quadratic reconstruction.

Furthermore, stencil selection is the only part of the algorithm that depends on the kind of the chosen spatial discretization. Our spatial discretization has a topological property which we use within the stencil selection: If two cells have a common point then they already have a line segment in common. We call such touching cells *neighbors*.

Polynomial degree 1. For a polynomial degree $n = 1$ the required stencil size is three. We select all those sets of three cells $\{\Omega_l, \Omega_a, \Omega_b\}$ as stencils for cell Ω_l which have the following properties:

- Ω_a is a neighbor of Ω_l, and
- Ω_b is a neighbor of Ω_l and of Ω_a.

Polynomial degree 2. For a polynomial degree $n = 2$ the required stencil size is six. We select all those sets of six cells $\{\Omega_l, \Omega_a, \Omega_b, \Omega_c, \Omega_d, \Omega_e\}$ as stencils for cell Ω_l which have the following properties: First, $\{\Omega_l, \Omega_a, \Omega_b\}$ has to be a selected stencil for polynomial degree $n = 1$. Second, Ω_c, Ω_d, and Ω_e have to fulfill one of the following three conditions (see Fig. 2 for an example of each type; Ω_l is dark shaded):

(1) Central stencil:
- Ω_c is a neighbor of Ω_l and of Ω_a, and
- Ω_d is a neighbor of Ω_l and of Ω_b, and
- Ω_e is a neighbor of Ω_l and of either Ω_c or Ω_d.

(2) Almost central stencil:
- Ω_c is a neighbor of Ω_l and of Ω_a, and
- Ω_d is a neighbor of Ω_l and of Ω_b, and
- Ω_e is a neighbor of Ω_a and of Ω_b.

(3) One-sided stencil:
- Ω_c is a neighbor of Ω_a and of Ω_b, and
- Ω_d is a neighbor of Ω_a and of Ω_c, and
- Ω_e is a neighbor of Ω_b and of Ω_c.

For most cells of our grids, the number of neighbors is six as in the figure. That means that the number of stencils with each type is six, resulting in a total number of 18 stencils for quadratic reconstruction.

3.2. Computation of the Reconstruction Polynomial for One Stencil

For this subsection we focus on one fixed stencil S. Furthermore we assume that the cells are enumerated such that $S = \{\Omega_1, \ldots, \Omega_{N(n)}\}$ and that we are going to reconstruct a

polynomial for cell Ω_1 with the standard representation (see Eq. (7))

$$p = \sum_{|\alpha| \le n} a_\alpha (x - b_1)^\alpha.$$

It is a mathematically uniquely solvable problem to compute the a_α for any data $\bar{u}_k$ if S is an admissible stencil, but in practice one encounters two serious difficulties:

- For the matrix entries $\langle (x - b_1)^\alpha \rangle_{\Omega_k}$ integrals $\int_{\Omega_k} (x - b_1)^\alpha$ have to be computed which is not trivial for $|\alpha| > 1$.
- The resulting matrix has a condition number which grows like h^{-n}, where h is the mesh width (see [A94]).

3.2.1. *Computation of the Matrix Entries*

Obviously for each k

$$\begin{aligned}
\left\langle (x - b_1)^{(0,0)} \right\rangle_{\Omega_k} &= 1, \\
\left\langle (x - b_1)^{(1,0)} \right\rangle_{\Omega_k} &= (b_k - b_1)_1, \\
\left\langle (x - b_1)^{(0,1)} \right\rangle_{\Omega_k} &= (b_k - b_1)_2.
\end{aligned}$$

For $|\alpha| > 1$ there are two options for the computation of $\int_{\Omega_k} (x - b_1)^\alpha$: First, the cell Ω_k can be subdivided into triangles and a quadrature rule that is exact for at least polynomial degree $|\alpha|$ can be used. Second, a primitive function $\mathcal{X}_\alpha$ for x^α in the sense that $\nabla \cdot \mathcal{X}_\alpha = x^\alpha$ can be used to reduce the problem to a boundary integral which can then be solved with a line integration quadrature rule on the boundary edges of Ω_k like it is described in [HMS96]. We choose the second technique because it fits very well into the environment of our finite volume flow solver.

For $|\alpha| = 2$ the following three primitive functions are used:

$$\begin{aligned}
\mathcal{X}_{(2,0)}(x_1, x_2) &:= \left(\frac{1}{3} x_1^3, 0 \right), \\
\mathcal{X}_{(0,2)}(x_1, x_2) &:= \left(x_1 x_2^2, 0 \right), \\
\mathcal{X}_{(1,1)}(x_1, x_2) &:= \left(\frac{1}{2} x_1^2 x_2, 0 \right).
\end{aligned}$$

Note, that in any case the computation of the $\int_{\Omega_k} (x - b_1)^\alpha$ is quite expensive. Thus, one could try to compute them only once and store them for later use. However, the number of cells Ω_k appearing in stencils for quadratic reconstruction of one cell Ω_1 is approximately 20. Thus, storing all these integrals for all cells would cost too much memory.

This can be circumvented by rewriting the integrals. One finds that for $|\alpha| = 2$

$$\langle (x - b_1)^\alpha \rangle_{\Omega_k} = (b_k - b_1)^\alpha + \langle (x - b_k)^\alpha \rangle_{\Omega_k}.$$

Using this formula in the quadratic case it is sufficient to store the values $\langle (x - b_k)^{(2,0)} \rangle_{\Omega_k}$, $\langle (x - b_k)^{(0,2)} \rangle_{\Omega_k}$, and $\langle (x - b_k)^{(1,1)} \rangle_{\Omega_k}$ for each cell Ω_k. The barycenters b_k have to be stored in any case.

3.2.2. *Circumventing Bad Condition Numbers of the Interpolation Matrix*

Let us first look at the standard representation of the reconstruction polynomial p of cell Ω_1. The corresponding interpolation matrix $\mathcal{A}$ is given by

$$\mathcal{A}_{k,\alpha} := \langle (x - b_1)^\alpha \rangle_{\Omega_k}, \qquad 1 \le k \le N(n),\ |\alpha| \le n.$$

One should note that there are two possible reasons for a bad condition number of $\mathcal{A}$. First, the condition number grows like h^{-n} if the computational grid gets finer. Second, stencil S may be (almost) inadmissible. The second reason cannot be circumvented easily (see above). One has to detect this and skip this stencil.

The bad behavior of the condition number on grid refinement can be circumvented by replacing the standard representation of the polynomial p.

Interpolation using barycentric coordinates. Abgrall in [A94] changes the standard representation (7), see earlier, into an expansion that uses barycentric coordinates.

Therefore he takes one of the admissible stencils S_i. There exists at least one subset of three cells out of S_i which is an admissible stencil for polynomials of degree one. For a simple notation we assume that this subset is given by $\{\Omega_1, \Omega_2, \Omega_3\}$. He considers the three polynomials Λ_i of degree one, defined by

$$\langle \Lambda_i \rangle_{\Omega_j} = \delta_i^j, \qquad 1 \le i \le 3,\ 1 \le j \le 3,$$

where δ_i^j denotes the Kronecker symbol. One finds that $\Lambda_1 + \Lambda_2 + \Lambda_3 = 1$. These polynomials are the barycentric coordinates of the triangle formed by the barycenters of Ω_1, Ω_2, and Ω_3. Using Λ_2 and Λ_3, Abgrall obtains a new representation of p:

$$p = \sum_{|\alpha| \le n} \hat{a}_\alpha \Lambda_2^{\alpha_1} \Lambda_3^{\alpha_2}.$$

The corresponding interpolation matrix $\hat{\mathcal{A}}$,

$$\hat{\mathcal{A}}_{k,\alpha} := \left\langle \Lambda_2^{\alpha_1} \Lambda_3^{\alpha_2} \right\rangle_{\Omega_k}, \qquad 1 \le k \le N(n),\ |\alpha| \le n,$$

has a condition number which is independent from h (see [A94]).

Interpolation using scaling. We here introduce a simpler technique to obtain a condition number independent from h. We define a local scaling factor

$$s := \frac{1}{\sqrt{|\Omega_1|}}$$

which can be read as an approximation to $1/h$ and change the standard representation (7) into

$$p = \sum_{|\alpha| \le n} \tilde{a}_\alpha s^{|\alpha|} (x - b_1)^\alpha. \tag{8}$$

The interpolation matrix $\tilde{\mathcal{A}}$ for this representation is given by

$$\tilde{\mathcal{A}}_{k,\alpha} := \left\langle s^{|\alpha|}(x-b_1)^\alpha \right\rangle_{\Omega_k}, \qquad 1 \le k \le N(n),\ |\alpha| \le n.$$

Note that $\tilde{\mathcal{A}}_{k,\alpha} = s^{|\alpha|}\mathcal{A}_{k,\alpha}$ and we get the standard representation (7) very easily from (8): $a_\alpha = s^{|\alpha|}\tilde{a}_\alpha$.

Remark 1. The matrix $\tilde{\mathcal{A}}$ is invariant with respect to grid scaling. That means, given an affine transformation $\mathcal{T}$ of the form

$$\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2, \qquad x \mapsto fx + y,\ 0 < f \in \mathbb{R},\ y \in \mathbb{R}^2$$

and given the transformed cells $\mathcal{T}(\Omega_k)$ and the transformed barycenters $\mathcal{T}(b_k)$ then the matrix $\tilde{\mathcal{A}}^{(\mathcal{T})}$ belonging to this transformed geometry is identical to $\tilde{\mathcal{A}}$. Especially, the condition number of $\tilde{\mathcal{A}}^{(\mathcal{T})}$ does not depend on the scaling factor f.

The proof of this remark is a simple application of the transformation rule for integrals: For the scaling factor $s^{(\mathcal{T})}$ computed from the transformed cell $\mathcal{T}(\Omega_1)$ we find

$$s^{(\mathcal{T})} = \frac{1}{\sqrt{|\mathcal{T}(\Omega_1)|}} = \frac{1}{\sqrt{f^2|\Omega_1|}} = \frac{s}{f}.$$

For $1 \le k \le N(n)$ and $|\alpha| \le n$ we have

$$\begin{aligned}
\tilde{\mathcal{A}}^{(\mathcal{T})}_{k,\alpha} &= \frac{1}{|\mathcal{T}(\Omega_k)|}\int_{\mathcal{T}(\Omega_k)} \left(s^{(\mathcal{T})}\right)^{|\alpha|}(\mathcal{T}(x)-\mathcal{T}(b_1))^\alpha\, d\mathcal{T}(x) \\
&= \frac{1}{f^2|\Omega_k|}\int_{\Omega_k}\left(\frac{s}{f}\right)^{|\alpha|}(fx-fb_1)^\alpha f^2\, dx \\
&= \frac{1}{|\Omega_k|}\int_{\Omega_k} s^{|\alpha|}(x-b_1)^\alpha\, dx = \tilde{\mathcal{A}}_{k,\alpha}.
\end{aligned}$$

Remark 2. It should be mentioned that the presented scaling technique only solves the condition problems induced by the regular refinement or scaling of regular grids. For distorted meshes like those for Navier–Stokes computations this simple scaling would not help much. In such cases the use of barycentric coordinates should be preferred.

Final remarks on the condition problem. Consider that we wish to use the standard representation (7) of p. To prevent the condition problems during the computation of p's coefficients a_α we first use our scaling technique and compute the coefficients $\tilde{a}_\alpha$ of the scaled representation. Second, we obtain the standard representation by computing $a_\alpha = s^{|\alpha|}\tilde{a}_\alpha$.

What happens with the a_α and with p if they are computed this way and the right hand data $\bar{u}_k$ are changed by a perturbation of magnitude ε?

Due to the scaling technique that omits the condition problems, the perturbation of the $\tilde{a}_\alpha$ is also of magnitude ε, independent from the grid spacing h, but the perturbation of the a_α is of magnitude $\varepsilon/h^{|\alpha|}$. This shows that also this method for computing the coefficients a_α has a condition number of $1/h^n$; but if

$$p(x) = \sum_{|\alpha|\le n} a_\alpha (x-b_1)^\alpha$$

is evaluated for $\|x - b_1\|$ of magnitude h then the perturbation of $p(x)$ is of magnitude ε because the powers of h within $(x - b_1)^\alpha$ eliminate the powers of $1/h$ within the perturbation of a_α. In this sense the condition problem really is omitted by computing the standard representation using the scaled representation as described above.

3.3. Choice of an Oscillation Indicator

Given a polynomial $p(x) = \sum_{|\alpha|\le n} a_\alpha (x - b_l)^\alpha$ for cell Ω_l we are searching for an indicator that measures the smoothness of p or, equivalently, detects how much p oscillates. In [A94] it is shown that as in the one-dimensional case the sum of the leading coefficient's absolute value

$$OI_1(p) := \sum_{|\alpha|=n} |a_\alpha|$$

has the property to tend to infinity if the mesh width h tends to zero and the data have a discontinuity in the kth derivative, $k < n$, while it remains always bounded if the data are smooth. Thus, the value OI_1 can be used as an oscillation indicator.

Aiming at the minimization of total variation one could think of using the L_1-norm of the first derivatives of p which we are normalizing to

$$OI_2(p) := \sum_{|\alpha|=1} \int_{\Omega_l} h^{-2} |D^\alpha p(x)|\, dx.$$

Because the L_1-norm is quite complicated to evaluate on our two-dimensional control volumes the L_2-norm can be used instead. This leads to

$$OI_3(p) := \left(\sum_{|\alpha|=1} \int_{\Omega_l} h^{-2} (D^\alpha p(x))^2\, dx \right)^{\frac{1}{2}}.$$

In [JS96] a new measurement for smoothness is presented. In our two-dimensional environment we are normalizing that indicator in a different way and come to

$$OI_4(p) := \left(\sum_{1\le|\alpha|\le n} \int_{\Omega_l} h^{2|\alpha|-4} (D^\alpha p(x))^2\, dx \right)^{\frac{1}{2}}.$$

Numerical tests with classical shock-tube problems were carried out to compare the indicators OI_1, OI_3, and OI_4 for $n = 2$. The results can be viewed in Subsection 4.2.

It turns out that indicator OI_1 is not suitable. Although it detects discontinuities of the data $\bar{u}_k$ as well as discontinuities of the first derivatives, the obtained solutions oscillate significantly. This may be explained by the extreme sensibility of the high order coefficients as described in Subsection 3.2.2. On the other hand indicator OI_3 works very well. Only small overshoots at discontinuities of the solution's first derivative can be seen. These decrease if the mesh width decreases.

With indicator OI_4 the solutions are very similar to those obtained with indicator OI_3. At a closer look features slightly sharper are resolved and overshoots are slightly smaller with OI_4 so this indicator is the first choice. Note that removing the $h^{2|\alpha|}$ weights within OI_4 results in an indicator that behaves similar to OI_1, hence, these weights are really essential.

3.4. Computation of Weights from the Oscillation Indicators

It should first be mentioned that the digital ENO scheme can be obtained by simply setting

$$\omega_i := \begin{cases} 1 & \text{if } OI(p_i) = \min\limits_{k=1,\dots,m} OI(p_k), \\ 0 & \text{else.} \end{cases}$$

Here one has to assume that there is only one minimum.

Choosing more than one weight different from zero leads to the weighted ENO scheme. Note that in any case the sum of the ω_i has to be 1 and weights must not be negative.

In our unstructured two-dimensional environment we cannot hope to improve the order of accuracy by choosing the right weights like in the one-dimensional case (see [LOC94, JS96]).

Let us assume that we have selected m admissible stencils $S_1, \dots, S_m$ and that for each S_i the reconstruction polynomial p_i of the data $\bar{u}_k$ was computed. The weights ω_i are then computed as

$$\omega_i := \frac{(\varepsilon + OI(p_i))^{-r}}{\sum_{k=1}^{m} (\varepsilon + OI(p_k))^{-r}},$$

where r is a positive integer. In [LOC94] the "ENO property" is defined. It would require here that for a stencil S_i in a smooth region ω_i is of magnitude 1 while for S_i in a discontinues region ω_i is of magnitude h^m. To fulfill this property we would have to set r to m for indicators OI_3 as well as for OI_4. On the other hand this power m within the definition of the "ENO property" seems quite arbitrary, so we forget about this strict definition and only use the principle idea that for stencils in a smooth region the weight should be of magnitude one while it should be *low* for stencils in a discontinues region. This is fulfilled for any positive r.

A strong property of the weighted ENO scheme is that in smooth regions the reconstruction is very smooth and stable. This property is partially lost if r is chosen too large. In that case the scheme tends to behave like the classical digital switching ENO scheme.

We did computations for test cases with even strong discontinuities (see Subsection 4.2 and 4.3) which showed that at least for $n \leq 2$, a power of $r = 4$ is large enough. The same was found in [JS96] in the one-dimensional case.

4. NUMERICAL APPLICATIONS

We have chosen three types of test cases: first, a smooth two-dimensional flow with an analytically known solution to demonstrate the accuracy of the method; second, classical one-dimensional shock-tube problems that were computed on a two-dimensional domain to show the behavior of the method for non-smooth flows; and, third, a true two-dimensional flow with strong discontinuities.

4.1. *Ringleb's Flow*

In [AR-211] flow cases of fully exact analytic solutions are presented. We took Ringleb's flow which is defined by its streamlines. We used a computational domain that is bounded

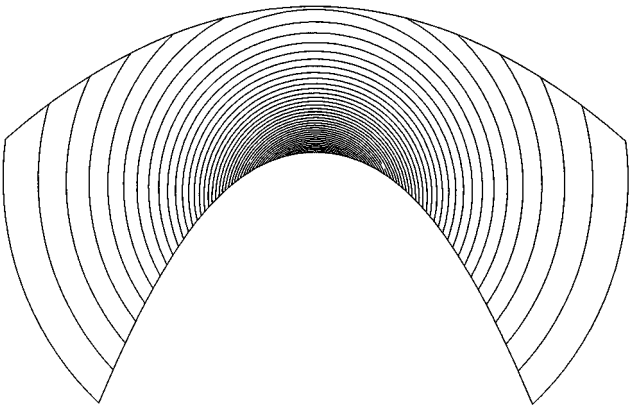

FIG. 3. Ringleb's flow: Density contours of exact solution.

by streamlines ($k = 0.4$; $k = 0.8$) and by the iso-velocity line at $q = 0.3$. The domain and the flow were rotated by 90° to obtain a left to right flow direction.

We used our Riemann solver with the outer state being the exact solution for boundary treatment at all but the outflow boundary. For the outflow boundary, characteristic boundary conditions were used.

In Fig. 3 one can see density contour lines of the interpolated exact solution. The aim of this test case was to measure the accuracy of the method for different mesh widths h. In Fig. 4 the discrete L_2 norm of the error in the density component is displayed over the mesh width h.

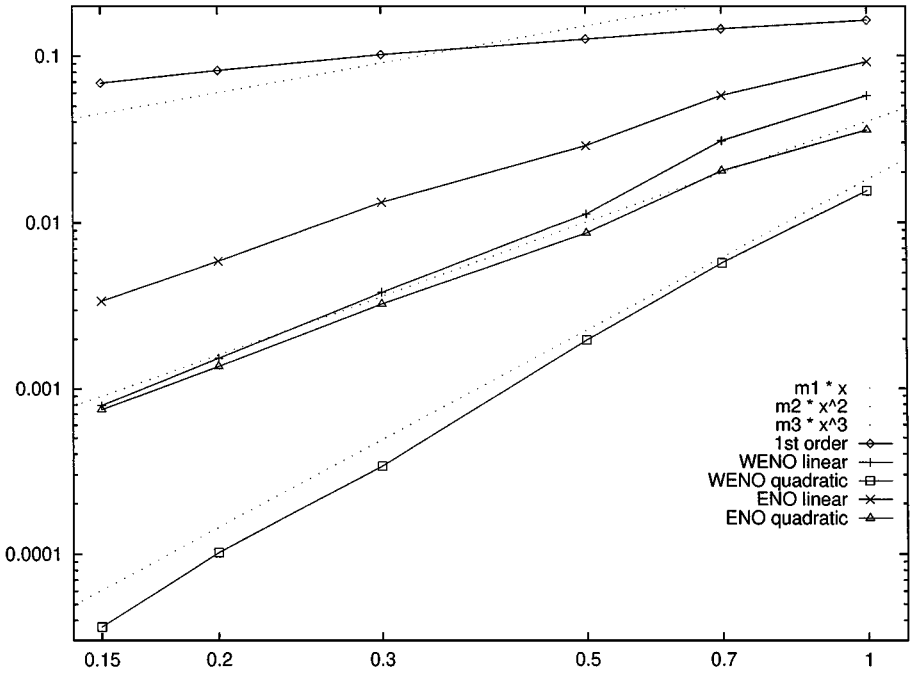

FIG. 4. Ringleb's flow: Discrete L_2 norm of density error over mesh width h (log-log plot).

The computations were carried out with the digital ENO scheme and with the weighted ENO scheme. For both schemes the oscillation indicator OI_4 was used. It should be mentioned that the results for indicator OI_3 are very similar but the errors for OI_3 are slightly larger.

The functions x, x^2, and x^3 are plotted so that one can compare with an idealistic 1st, 2nd, and 3rd order behavior. The 1st order finite volume method does not even reach a linear progression with h.

The results for the weighted ENO method using polynomials of degree $n = 1$ (labeled WENO linear) show an error behavior slightly better than quadratic. The error progression for the corresponding digital ENO method (ENO linear) is similar but the error level is significantly higher.

The results for the weighted ENO method using polynomials of degree $n = 2$ (WENO quadratic) show an error progression better than cubic. For $n = 2$ one notices an order loss for the digital ENO method (ENO quadratic) to about second order.

The results demonstrate the clear advantage of the weighted ENO scheme over the digital ENO scheme in smooth flow regions. Note that these improvements do not require a significant increase of computational work. Little additional work is only required for the computation of the weights ω_i and the weighted sums $p = \sum_{i=1}^{m} \omega_i p_i$ which is negligible compared to the overall costs of the scheme.

It should be mentioned that Ringleb's flow is steady. With all of the used oscillation indicators the weighed ENO scheme was able to drive $d\rho/dt$ to machine epsilon ($\approx 10^{-15}$) while the digital ENO scheme failed in this respect. It could not even reach 10^{-4}.

4.2. *Shock-Tube Problems*

Two of the standard one-dimensional shock-tube problems were computed on a two-dimensional grid. No special treatments to keep the flow one-dimensional were applied (see Fig. 5).

The displayed one-dimensional results were obtained by extracting the data along the central horizontal cut line. For all computations the degree of the reconstruction polynomials was set to $n = 2$.

The first test case was proposed by Lax. It is defined by

$$(\rho, q, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x < 0.5, \\ (0.5, 0, 0.571) & \text{if } x > 0.5. \end{cases}$$

In Figs. 6 and 7 one can see density and velocity obtained for Lax's problem with the weighted ENO scheme using different oscillation indicators. With indicator OI_1 a significant overshoot of the density at the contact discontinuity occurred although the resolution of the discontinuities is not better than with indicators OI_3 and OI_4.

The velocity distribution shows that also in regions where the solution is continuous but the first derivative is discontinuous the results obtained with indicators OI_3 and OI_4 are less oscillatory than those obtained with OI_1. In this region indicator OI_4 shows some

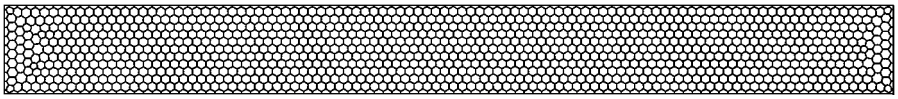

FIG. 5. Shock-tube problems: Used grid corresponds to typical resolution of 100 points.

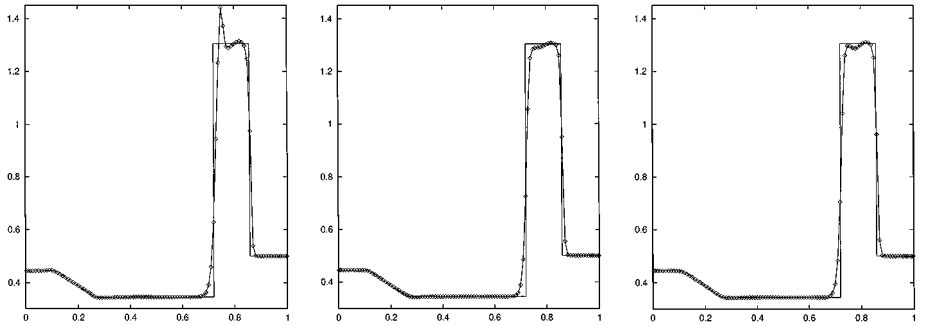

FIG. 6. Lax's problem: Density on the centerline; weighted ENO: OI_1; OI_3; OI_4.

improvement over OI_3: The height of the overshoot is smaller and at the same time the resolution of all features is nowhere worse.

Even for this test case dominated by discontinuities the digital ENO scheme does not provide better results. Comparing with Fig. 8 one can see that the weighted ENO scheme using indicator OI_4 resolves the flow features sharper than the digital ENO scheme and at the same time the overshoots are even a little smaller.

The second shock-tube problem is Sod's problem which is defined as

$$(\rho, q, p) = \begin{cases} (1, 0, 1) & \text{if } x < 0.5, \\ (0.125, 0, 0.1) & \text{if } x > 0.5. \end{cases}$$

For this test case we present a comparison of the density distributions for the weighted ENO scheme and the digital ENO scheme in Fig. 9. For both schemes oscillation indicator OI_4 was used.

Again we find that the weighted ENO scheme resolves the flow features slightly sharper than the digital ENO scheme. For both schemes no kinds of overshoots or oscillations are visible.

For this case we found the same ranking of the results' qualities obtained with the different indicators as for Lax's problem but in this case the differences were less visible so that we do not present the results for indicators OI_1 and OI_3 here.

4.3. *Reflection of a Shock on a Wedge*

We chose one test case from [WC84], the *Double Mach Reflection of a Strong Shock*. We used the setup that drives a shock down a tube which contains a wedge (30°). The shock

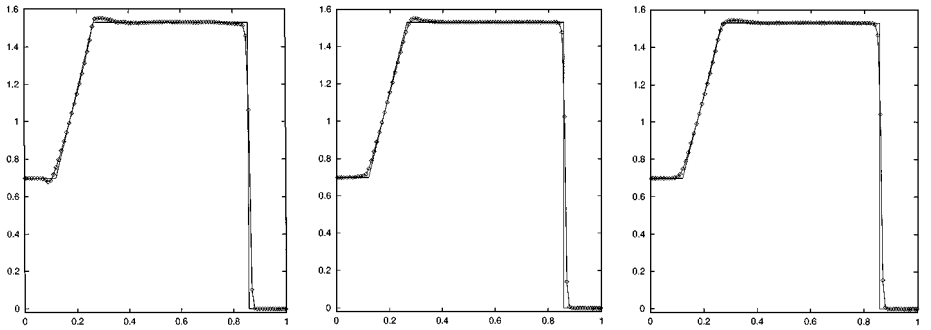

FIG. 7. Lax's problem: Velocity on the centerline; weighted ENO: OI_1; OI_3; OI_4.

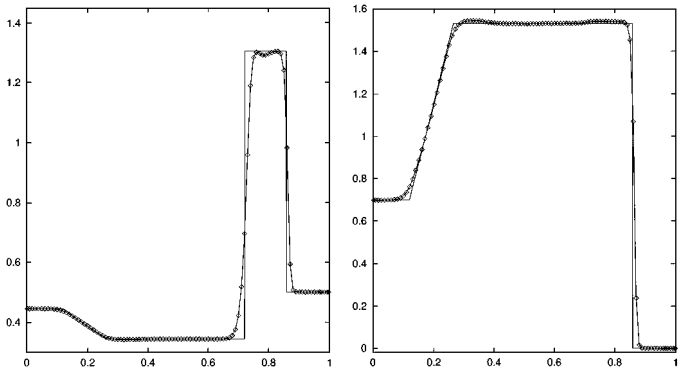

FIG. 8. Lax's problem: Density (left) and velocity (right) on the centerline; digital ENO; OI_4.

moves with a Mach number of 10; the undisturbed air ahead the shock has a density of 1.4 and a pressure of 1. These data require a state of density $\rho = 8$, velocity $u = 8.25$, and pressure $p = 116.5$ on the left hand side of the shock.

We did the computations on a grid with mesh width $h = 1/60$ (16,316 grid points) and one with $h = 1/120$ (63,497 grid points) (see Fig. 10). These data correspond to the medium and the finest grid in [WC84]. The grids were constructed using the method described in [F93].

The computations where carried out with the weighted ENO and the digital ENO scheme both using oscillation indicator OI_4. For this test case with reconstruction degree $n = 2$ we needed a safety check for the reconstruction polynomial like the one described in [A94]. If a reconstructed value at an integration point of a cell was illegal (negative pressure or density) then the reconstruction degree for that cell was reduced to $n = 1$. At some time steps this happened for one or two cells inside the primary shock. Note that we did not reduce the reconstruction degree at the boundary like it was done in [A94].

The results for the weighted ENO scheme shown in Figs. 11 and 12 are quite good when compared to those in [JS96] and to the MUSCL results in [WC84]. The instabilities that can be seen between the reflected shock and the wedge can also be seen in the MUSCL results in [WC84].

Note that we used the reconstruction polynomials obtained from the flow solver to draw the isolines locally on each cell. No interpolation procedure was utilized. This is the reason for the discontinuous isolines that can be seen in some regions especially near the shocks.

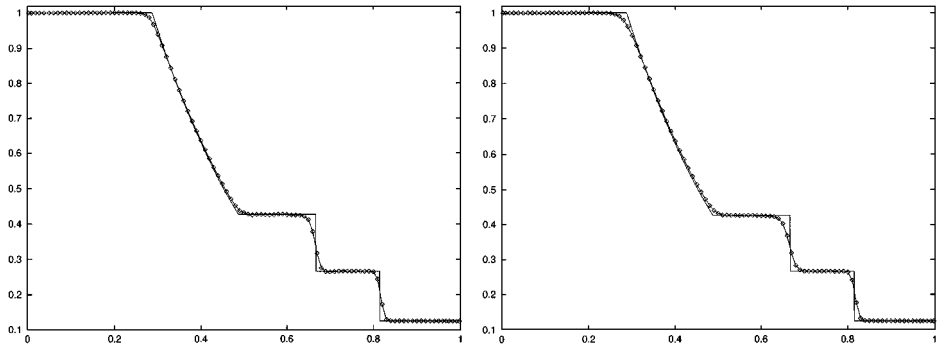

FIG. 9. Sod's problem: Density distribution on the centerline; weighted ENO; digital ENO; OI_4.

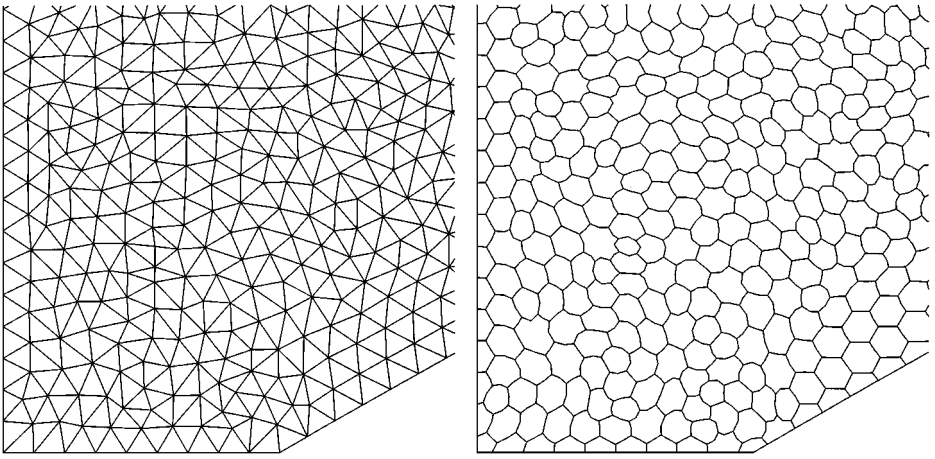

FIG. 10. Double Mach reflection: Section of the medium grid: triangulation; computational grid.

One can very well see the improvement achieved by the third order method ($n = 2$) over the second order method ($n = 1$). The third order method results in a similar resolution of the flow features on the medium grid (Fig. 11, right) as the second order method on the fine grid (Fig. 12, left). If one takes into account that the fine grid has approximately 4 times as many cells than the medium grid, that the time step size is halved, and that the computing time of the third order method is approximately 4 times higher than that of the second order method, then it is clear that the computing time for the third order method on the medium grid is only half of the computing time for the second order method on the fine grid and that the memory requirement saved is even more than a factor of two. In other words, for this test case it is much more efficient to use the third method on a coarser grid instead of the second order method on a finer grid.

In Fig. 13 one can see that also for the digital ENO scheme there is an improvement when the polynomial degree is raised from $n = 1$ to $n = 2$. But comparing Figs. 12 with 13 one can see again that the weighted ENO scheme is much more accurate than the digital ENO scheme. Especially the contact discontinuity generated at the shock triple point and the weak shock generated at the second Mach reflection are much sharper resolved by the weighted ENO scheme. The result for the digital ENO scheme with $n = 2$ has about the

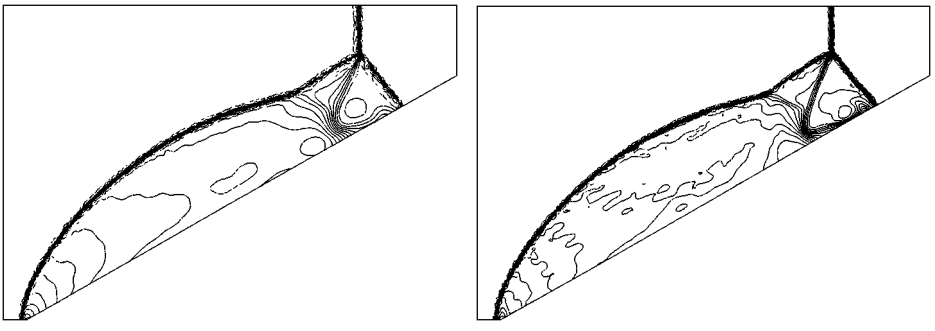

FIG. 11. Double Mach reflection: 30 density contour lines from 1.731 to 20.92 on the medium grid. Weighted ENO, OI_4: $n = 1, 2$.

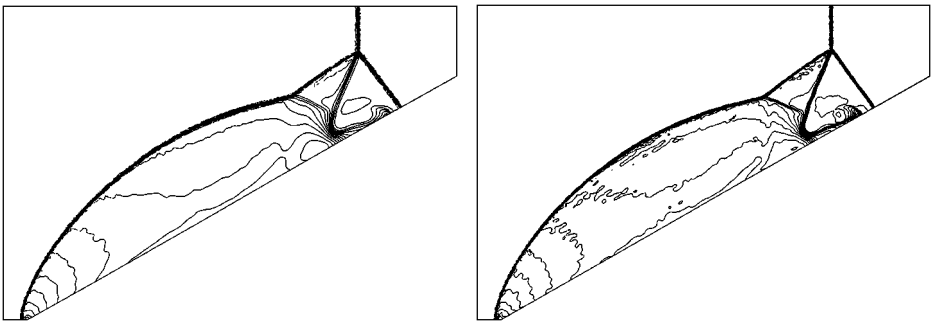

FIG. 12. Double Mach reflection: 30 density contour lines from 1.731 to 20.92 on the fine grid. Weighted ENO, OI_4: $n = 1, 2$.

same quality as the weighted ENO scheme on the same grid but with only $n = 1$. The same correspondence was found for Ringleb's flow (see Fig. 4).

5. CONCLUSIONS

The weighted ENO scheme applied to the Euler equations using unstructured grids was demonstrated to be a massive improvement of the digital ENO scheme applied to the same type of grids. One does not have to investigate much work to implement a weighted ENO scheme from a classical digital ENO scheme.

The weighted ENO scheme is not much more expensive than the digital ENO scheme but here one should mention that the inductive stencil selection technique presented in [A94] which can reduce the costs of a digital ENO scheme cannot be applied.

The numerical results show that the weighted ENO scheme is more accurate than the digital ENO scheme and at the same time it seems to be even more stable. This is a very nice result because usually one loses stability when increasing the accuracy of a method. With the weighted ENO scheme smooth flows can be computed much better than with digital ENO schemes which can be seen clearly for Ringleb's flow. Additionally, convergence to machine accuracy for steady flows is obtained with the weighted ENO scheme while that does not seem to be possible for the digital ENO scheme.

Seeing these advantages of the weighted ENO scheme over the digital ENO scheme one should expect that it is applicable for an even wider range than the digital ENO scheme already is.

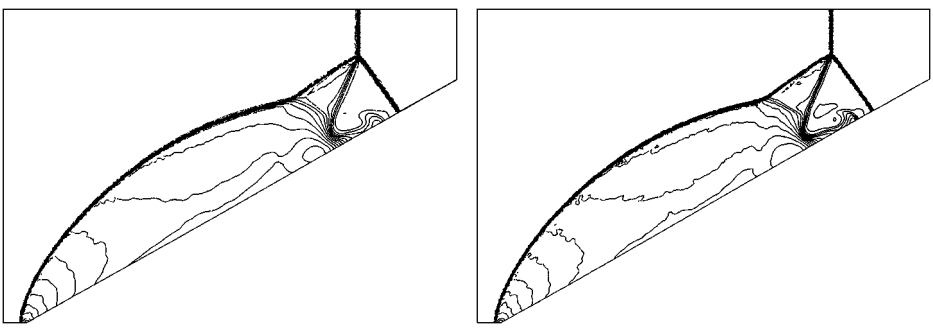

FIG. 13. Double Mach reflection: 30 density contour lines from 1.731 to 20.92 on the fine grid. Digital ENO, OI_4: $n = 1, 2$.

REFERENCES

[A94] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation, *J. Comput. Phys.* **114**, 45 (1994).

[AR-211] *Test Cases for Inviscid Flow Field Methods*, AGARD Advisory Report No. 211.

[F93] O. Friedrich, A new method for generating inner points of triangulations in two dimensions, *Comput. Methods Appl. Mech. Engrg.* **104**, 77 (1993).

[HC91] A. Harten and S. R. Chakravarthy, *Multi-Dimensional ENO Schemes for General Geometries*, ICASE Report No. 91-76, 1991.

[HMS96] D. Hietel, A. Meister, and Th. Sonar, On the comparision of four different implementations of an implicit third-order ENO scheme of box type for the computation of unsteady compressible flow, *Numer. Algorithms* **13**, 77 (1996).

[JS96] G.-S. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* **126**, 202 (1996).

[LOC94] X.-D. Liu, S. Osher, and T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* **115**, 200 (1994).

[OS82] S. Osher and F. Solomon, Upwind difference schemes for hyperbolic conservation laws, *Math. Comp.* **38**, 339 (1982).

[SO88] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* **77**, 439 (1988).

[S97] Th. Sonar, On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: Polynomial recovery, accuracy and stencil selection, *Comput. Methods Appl. Mech. Engrg.* **140**, 157 (1997).

[WC84] P. Woodward and Ph. Colella, The numerical simulation of two-dimensional fluid flows with strong shocks, *J. Comput. Phys.* **54**, 115 (1984).