UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

High Order Shock Capturing Methods

Bjorn Engquist
Bjorn Sjogreen

March 1995

CAM Report 95-14

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

High Order Shock Capturing Methods

Bjorn Engquist* and Bjorn Sjogreen†

Abstract

Shock capturing finite difference methods of TVD and ENO type are presented. Several schemes are described in detail both for scalar nonlinear conservation laws and for systems. Theoretical results are also discussed. Some new algorithmic variants and convergence analysis are given.

1 Introduction

The physical laws for conservation of mass, momentum and energy are fundamental in the mathematical modelling of fluid flow. We shall here disucss the numerical approximation of nonlinear hyperbolic conservation laws. For simplicity we shall mainly concentrate on scalar problems in one space dimension and time,

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

We shall also discuss the extension to general systems,

$$\begin{cases} (u_m)_t + \nabla_x f_m(u) = 0 \\ u_m(x,0) = u_m(x) \end{cases} \tag{1.2}$$

$$m = 1, 2, \ldots, M, \quad x \in \boldsymbol{R}^d, \quad d = 1, 2 \text{ or } 3.$$

A typical application would be the Euler equation for inviscid flow. The numerical methods for the inviscid equations are often used as building blocks also for the simulation of viscous flow at high Reynold's numbers.

Below follows a discussion of the basic principles guiding the scheme development. In sections 2, 3, and 4 the total variation diminishing (TVD) and essentially non oscillatory (ENO) schemes are presented both from theoretical and practical points of view. Some convergence theory is given in section 5 and in section 6 the most important algorithms are given in a form suited for computer implementation.

1.1 Background

Computations of solutions with shocks in the simulation of fluid flow goes back to the 1940's, [vNR]. The main numerical difficulty is the approximation at the discontinuities. Nonlinear hyperbolic conservation laws generically produce solutions with discontinuities even if the initial and boundary values are smooth.

The standard theories for finite difference and finite element methods do not apply and many of the common methods do not work well in practice either, in the presence of shocks. During the last few decades our knowledge in shock capturing techniques have increased drastically from theoretical results via algorithmic development to production codes. No theory can yet guarantee convergence in the approximation of solutions to multidimensional systems of nonlinear conservation laws. However, in practice efficient 3-D calculations of complete airplane configurations are possible. The design of these modern high resolution algorithms are based on principles some of which can be established on simpler model equations as e.g. (1.1). We shall consider explicit difference schemes and an

*Department of Mathematics, UCLA. Research supported by ARPA/ONR-N00014-92-J-1890 and NSF DMS-91-03104

†Department of Scientific Computing, Uppsala University

approximation of (1.1) will have the form,

$$\begin{cases} u_j^{n+1} = u_j^n - \lambda \Delta_- h_{j+\frac{1}{2}}^n, \\ u_j^0 = u_0(x_j), \end{cases} \tag{1.3}$$

$$\begin{aligned} & h_{j+\frac{1}{2}}^n = h(u_{j-r+1}^n, \ldots, u_{j+r}^n), \\ & x_j = j\Delta x,\ t_n = n\Delta t, \\ & j = \ldots, -1, 0, 1, \ldots, n = 0.1, \ldots, \end{aligned} \tag{1.4}$$

where u_j^n is the approximation of $u(x_j, t_n)$. The notation $\lambda = \Delta t/\Delta x$ is used. Δ_+ and Δ_- are the forward and backward undivided difference operators, and D_+ and D_- are the corresponding divided difference operators.

1.2 Design Principles

In [30], Strang showed that, for smooth solutions to nonlinear hyperbolic equations, consistent and linearly stable finite difference approximation convergence. If the local truncation error is of order p the global error in L_2 is $\mathcal{O}(\Delta t^p)$. Even if stability of the linearized discrete approximation can be replaced by a nonlinear control of the growth of perturbations, [13] it is still a useful concept in scheme design together with a high order of the local truncation error.

Around the discontinuities, shocks and contacts, the local errors are pointwise $\mathcal{O}(1)$. In order for these errors not to cause blow-up of the numerical approximation or to spread to the smooth parts of the solution, extra conditions are needed.

The most straight forward sufficient conditions for convergence are that consistency, conservation forms and monotonicity implies L_1 convergence with error $O(\Delta t^{\frac{1}{2}})$, [17], [6],

The scheme (1.3) is on conservation form. Consistency and monotonicity are respectively defined by,

$$\begin{aligned} & h(u, u, \ldots, u) = f(u), \\ & u_j^n - \lambda \Delta_- h_{j+\frac{1}{2}}^n \text{ monotone as} \\ & \text{function of } u_{j-r}^n, \ldots, u_{j+r}^n. \end{aligned}$$

The initial value and flux function are assumed to satisfy,

$$u_0 \in L^1 \cap L^\infty \cap BV,\ f, h \in c^1.$$

The monotonicity requirement restricts the approximation to be of only first order. The other conditions consistency and conservation forms are present in all approximations. Conservation form is needed to produce the correct shock speed, [18]. In order to allow for higher orders in the approximation, monotonicity must be replaced by other features which can control oscillations close to the discontinuities. Furthermore, monotonicity does not make sense for systems.

Some of the design principles for achieving the desired features are artificial viscosity, upwinding and limiters.

The traditional method of controlling the oscillations is by adding artificial viscosity. This corresponds to approximating (1.2) with a right hand side which typically can have the form $\frac{d}{dx} C \frac{d}{dx} u$. The viscosity coefficient C is usually a function of Δt and $\frac{d}{dx} u$. It should decrease as Δt decreases. The early method of von Neumann and Richtmyer, [36], used artificial viscosity and so does for example the modern finite volume schemes by Jameson and the finite element schemes by Hughes and Johnson [17], [40]. We shall not consider those classes of methods but concentrate on the high resolution techniques based on upwinding, limiters and for systems field by field decomposition.

Entropy conditions are also important in order to rule out unphysical discontinuities.

Limiters and upwinding are techniques for controlling oscillations near the shocks. If the total variation of the approximation is not increasing, no oscillations can be generated. It is thus an important design goal to create methods for which the total variation is not increasing or increasing very little when applied to scalar problems. The total variation of the solution of (1.1) is not increasing with time. The total variation of (1.1) and (1.3) are defined as,

$$\begin{aligned} & TV(u(\cdot, t)) = \exp_{\{x_j\}} \sum_j |ux_{j+1}, t) - u(x_j, t)|, \\ & TV(u^n) = \sum_j |u_{j+1}^n - u_j^n|. \end{aligned}$$

2 Second order TVD Schemes

The TVD concept was first introduced in [19], but it was in [11] that it was given a name (total variation non-increasing, TVNI, which was later changed to TVD), and became widely known. Furthermore, in [11] TVD schemes of formal order of accuracy two, was for the first time derived. Long before these results, the flux corrected transport method (FCT) had been derived [1]. The FCT method has much in common with the TVD scheme derived in [11], however FCT was derived for systems of conservation laws, and the TVD issue was never adressed directly. Actually the FCT method is TVD only if the CFL number is sufficiently small. We will below present a new and more general description of the FCT method.

Both Harten's TVD method [11] and the FCT method, are based on the Lax-Wendroff scheme. The methods consist of inserting limiters into the Lax-Wendroff scheme, which will act to supress any Gibbs type of oscillations.

The paper [11] was the starting point for an intense activity in the field. Several TVD schemes, simplifying the scheme in [11] were derived [31] [7] [38].

The advantage with the Lax-Wendroff scheme is its low computational cost, it is optimal in the sense that it has the highest possible accuracy of all explicit three point schemes. However, it has also two disadvantages. Firstly, it it not straighforward to generalize to two or three space dimensions. This is always done by dimensional splitting. To insert limiters into a truly two or three dimensional Lax-Wendroff scheme is extremely complicated. Some effort to use Lax-Wendroff ideas for systems in several space dimensions has been made in [13], but it seems to be very costly and complicated. Secondly, in steady state computations, the steady state computed by a Lax-Wendroff method will depend on the size of the time step. One can argue that this is not serious, since the time step can be seen as an artificial viscosity coefficient. However, this excludes using implicit method with large time steps, in which case the steady state error would be large. Furthermore, convergence iteration with the multi grid method becomes more complicated, when one as to take into account how the time step changes on the different grid levels.

There was another line of development, originated in [33] [34], where the schemes were based on the a semi-discretization with pure centered differences for the space derivatives. TVD schemes were derived by non-oscillatory interpolation of either the grid function itself or by interpolating the numerical fluxes. Schemes based on interpolation of the grid function, which sometimes are called the MUSCL scheme were derived and analyzed in e.g., [4] [22]. A thorough survey of various limiting techiniques for this scheme can be found in [29]. Schemes based on flux interpolation were described and analyzed in [23]. We will here denote the MUSCL scheme by the *inner* TVD scheme, and the scheme based on flux interpolation the *outer* TVD scheme, to stress the analogy with the centered difference fluxes

$$h_{j+1/2} = f((u_j + u_{j+})/2)$$
$$h_{j+1/2} = (f(u_j) + f(u_{j+1}))/2$$

on which these two methods are based.

It has turned out today that the *outer* TVD scheme in a simplified form is probably the most efficient to use in CFD computations [35] [14]. This simplified scheme, which we describe in Subsection 2.2.2 below, was also derived in [38] from the Lax-Wendroff approach, by simplifying away the Lax-Wendroff viscosity term.

The classification of TVD methods above, is summarized in Fig. 2.1.

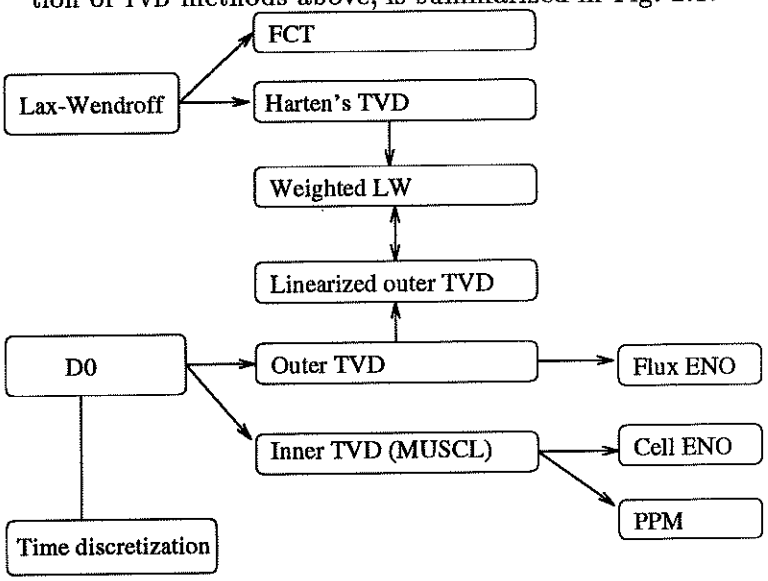

FIG. 2.1. *Classification of TVD schemes.*

2.1 Methods based on the Lax-Wendroff method

2.1.1 Flux corrected transport

This is the oldest shock capturing high resolution method. The method is in the form

$$\begin{aligned} u^* &= L(u^n) \\ u^{n+1} &= u^* + M(u^*) \end{aligned} \tag{2.5}$$

where L is a first order TVD scheme and M is a modification such that $L(u^n) + M(L(u^n))$ is TVD and the Lax-Wendroff scheme whenever possible. We thus implement the second order modification as a corrector step to the TVD predictor. We use the predictor step

$$u_j^* = u_j^n - \lambda \Delta_- h_{j+1/2}^n$$

where $h_{j+1/2}^n$ is the numerical flux of a first order TVD method. The corrector step is

$$u_j^{n+1} = u_j^* - (b_{j+1/2} - b_{j-1/2}) \tag{2.6}$$

where

$$b_{j+1/2} = \begin{cases} 0 & if\ \Delta_+ u_j^* \Delta_- u_j^* < 0\ or \\ & \Delta_+ u_{j+1}^* \Delta_- u_{j+1}^* < 0 \\ s \min(\frac{1}{2}|\Delta_- u_j^*|, d_{j+1/2}|\Delta_+ u_j^*|, \frac{1}{2}|\Delta_+ u_{j+1}^*|) & \\ otherwise & \end{cases} \tag{2.7}$$

Here s =sign$(\Delta_+ u_j^*)$ and $d_{j+1/2} = \frac{1}{2}(Q_{j+1/2} - Q_{j+1/2}^{LW})$, where $Q_{j+1/2}$ is the numerical viscosity of the first order predictor, and $Q_{j+1/2}^{LW}$ is the numerical viscosity of the Lax-Wendroff method.

We can see that no change is made at extrema, and thus that the accuracy is only first order there. It is not hard to prove the following.

Theorem 2.1. *The FCT method (2.5) where L is a first order TVD scheme and M is given by (2.6), (2.7) is TVD and second order accurate away from extrema.*

Remark: The method of artificial compression (ACM) [10] was an early attempt to design high resolution shock capturing schemes. ACM is on the form (2.5), (2.6), but with

$$b_{j+1/2} = \begin{cases} 0 & if\ \Delta_+ u_j^* \Delta_- u_j^* < 0\ or \\ & \Delta_+ u_{j+1}^* \Delta_- u_{j+1}^* < 0 \\ sign(\Delta_+ u_j^*) \min(|\Delta_- u_j^*|, |\Delta_+ u_j^*|, |\Delta_+ u_{j+1}^*|) & \\ otherwise & \end{cases}$$

this correction sharpens discontinuities and can be made TVD with some changes, but is not in general second order accurate (not even away from extrema).

Originally, FCT was defined using the scheme with numerical flux

$$h_{j+1/2} = (f_j + f_{j+1})/2 - \frac{1}{2\lambda}\frac{1}{8}\Delta_+ u_j$$

as predictor. This scheme is TVD and first order accurate under the CFL condition $\lambda a_{j+1/2} \le \sqrt{3}/2$. This gives $d_{j+1/2} = \frac{1}{8}$, a constant, and the computation of the antidiffusive flux in the corrector step becomes very simple. Furthermore, FCT was defined using the corrector flux

$$b_{j+1/2} = \begin{cases} 0 & if\ \Delta_+ u_j^* \Delta_- u_j^* < 0\ or \\ & \Delta_+ u_{j+1}^* \Delta_- u_{j+1}^* < 0 \\ s \min(|\Delta_- u_j^*|, d_{j+1/2}|\Delta_+ u_j^*|, |\Delta_+ u_{j+1}^*|) & \\ otherwise & \end{cases} \tag{2.8}$$

which in general does not lead to a TVD method.

We have here modified the flux (2.8) with factors $\frac{1}{2}$ in some places, to make the total method TVD for arbitrary TVD predictors. Alternatively a more restrictive CFL condition could have been imposed on the corrector step.

2.1.2 Harten's TVD scheme with simplifications

This method is built on the second order Lax-Wendroff scheme, which has the numerical flux

$$h_{j+1/2}^{LW} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2\lambda}(\lambda a_{j+1/2})^2 \Delta_+ u_j.$$

The crucial observation is that if for any numerical flux function $h_{j+1/2}$ it holds that $h_{j+1/2} = h_{j+1/2}^{LW} + O(\Delta x^2)$ then $h_{j+1/2}$ is second order accurate, too. Here the

leading error term in the $O(\Delta x^2)$ must be differentiable.

Harten's TVD scheme (also called the ULT1 scheme) is defined by taking the first order numerical flux

$$h^1_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2\lambda}Q(\lambda a_{j+1/2})\Delta_+ u_j$$

and apply it to the modified flux function $f(u) + g(u)$. $Q(x)$ is any numerical viscosity which gives TVD. The flux of Harten's TVD scheme is

$$h^M_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j + g_{j+1} + g_j) - \frac{1}{2\lambda}Q(\lambda a^M_{j+1/2})\Delta_+ u_j$$

where $a^M_{j+1/2}$ is computed from the modified flux, i.e.,

$$a^M_{j+1/2} = \frac{f_{j+1} - f_j}{\Delta_+ u_j} + \frac{g_{j+1} - g_j}{\Delta_+ u_j}$$

Harten shows that the conditions

$$\begin{aligned} g_j + g_{j+1} &= 2(h^{LW}_{j+1/2} - h_{j+1/2}) + O(\Delta x^2) \\ g_{j+1} - g_j &= O(\Delta x^2) \end{aligned}$$

implies second order accuracy. Intuitively, this means that in the expression for $h^M_{j+1/2}$ above, the term $(g_j + g_{j+1}) \approx h^{LW}_{j+1/2} - h^1_{j+1/2}$ while the rest of the terms is $h^1_{j+1/2}$, so that the sum is $\approx h^{LW}_{j+1/2}$. The TVD property follows by the standard theorem, (that $\lambda|a_{j+1/2}| \leq Q_{j+1/2} \leq 1$), applied to the flux $h^M_{j+1/2}$. An important condition for making this possible is that $(g_{j+1} - g_j)/\Delta_+ u_j$ is bounded, i.e., that $g_{j+1} = g_j$ whenever $u_{j+1} = u_j$. The choice made in [11] is

$$g_j = \text{minmod}(h^{LW}_{j+1/2} - h_{j+1/2}, h^{LW}_{j-1/2} - h_{j-1/2}) =$$

from which one can prove TVD under the standard CFL restriction $\lambda|a_{j+1/2}| \leq 1$.

The scheme above was later simplified by linearizing parts of the numerical flux. This lead to the class of upwind - Lax-Wendroff weighted methods, which we now describe. In these methods, the numerical flux function which is written as a weighted average of the upwind method and the Lax-Wendroff method,

$$h_{j+1/2} = (1 - w_{j+1/2})h^{UPW}_{j+1/2} + w_{j+1/2}h^{LW}_{j+1/2}.$$

Any first order TVD method can be used instead of the upwind flux, $h^{UPW}_{j+1/2}$. The idea is to have $w_{j+1/2} \approx 1$, when the solution is smooth, and $w_{j+1/2} \approx 0$ near discontinuities. Note that Harten's TVD scheme can not be written in this way, due to the non-linear dependence of $Q_{j+1/2}$ on the modified flux.

For this class of methods, the known results about TVD have mainly been worked out for the linear problem $u_t + au_x = 0$. For this problem we obtain the numerical flux

$$\begin{aligned} h_{j+1/2} &= a(u_{j+1} + u_j)/2 - \tfrac{1}{2}|a|\Delta_+ u_j \\ &+ \tfrac{1}{2\lambda}(\lambda|a| - (\lambda a)^2)w_{j+1/2}\Delta_+ u_j \end{aligned} \tag{2.9}$$

Example of weight functions are

$$w_{j+1/2} = \begin{cases} \phi(r_j) & \text{if } a > 0 \\ \phi(1/r_{j+1}) & \text{if } a < 0 \end{cases} \tag{2.10}$$

given in [31], or

$$w_{j+1/2} = \phi(r_j) + \phi(1/r_{j+1}) - 1 \tag{2.11}$$

given in [7], or the more general [38]

$$w_{j+1/2} = L(r_j, \frac{1}{r_{j+1}})$$

Where we define

$$r_j = \frac{\Delta_- u_j}{\Delta_+ u_j}$$

as a measure of the smoothness of u_j. When u_j is smooth, and does not have an extreme point, $r_j = 1 + \mathcal{O}(\Delta x)$.

The function $\phi(r)$ is called *limiter*. We require that $\phi(1) = 1$, which implies that

$$\phi(r_j) = 1 + \mathcal{O}(\Delta x) \quad \phi(1/r_j) = 1 + \mathcal{O}(\Delta x)$$

and consequently

$$\begin{aligned} h_{j+1/2} &= h^{LW}_{j+1/2} + (1 - w_{j+1/2})(h^{UPW}_{j+1/2} - h^{LW}_{j+1/2}) \\ &= h^{LW}_{j+1/2} + \mathcal{O}(\Delta x)\mathcal{O}(\Delta x) \end{aligned}$$

at smooth non-extreme points, and for the weight functions (2.10), (2.11). $\phi(1) = 1$ thus guarantees second order of accuracy. The following theorem is proved in [31].

Theorem 2.2. *The method with numerical flux (2.9), and limiter (2.10), approximating $u_t + au_x = 0$ is* TVD *if $\phi(r)$ satisfies*

$$0 \leq \phi(r) \leq 2 \quad 0 \leq \phi(r)/r \leq 2$$

For (2.11), the theorem below is given in [7].

Theorem 2.3. *The method with numerical flux (2.9), and limiter (2.11), approximating* $u_t + au_x = 0$ *is* TVD *if* $\phi(r)$ *satisfies*

$$0 \le \phi(r) \le 1 \quad \phi(r) = 0 \ \ for \ \ r \le 0$$

Example of a function satisfying the conditions on $\phi(r)$ in theorem 2.2 is

$$\phi(r) = \begin{cases} \frac{2r}{r+1} & \text{if } r > 0 \\ 0 & \text{otherwise} \end{cases}$$

and for (2.11) the function

$$\phi(r) = \begin{cases} \min(2r, 1) & r > 0 \\ 0 & r \le 0 \end{cases}$$

is often used. There is a special terminology for this class of methods. The scheme with limiter (2.10) is called an *upwind* TVD scheme, and the scheme with the limiter (2.11) a *symmetric* TVD scheme, thus indicating whether the upwind direction is required in the computation of the weight function. Note that in both cases the upwind direction is required when computing the flux $h^{UPW}_{j+1/2}$. The symmetric TVD scheme is simpler than the upwind TVD scheme, but we pay for the simlicity because the TVD analysis for the case (2.11) gives more restrictive conditions on ϕ.

Another version of these methods is obtained if the weighting is based on the entire flux function, as done in [31]. There the weight is defined by

$$w_{j+1/2} = \begin{cases} \phi(\frac{h^{LW}_{j-1/2} - h^{UPW}_{j-1/2}}{h^{LW}_{j+1/2} - h^{UPW}_{j+1/2}}) & \text{if } a_{j+1/2} > 0 \\ \phi(\frac{h^{LW}_{j+3/2} - h^{UPW}_{j+3/2}}{h^{LW}_{j+1/2} - h^{UPW}_{j+1/2}}) & \text{if } a_{j+1/2} < 0 \end{cases}$$

in the case where the low order method is the upwind scheme (i.e., has numerical viscosity $\lambda|a_{j+1/2}|$).

Alternatively, we can generalize by substituting the local wave speed $a_{j+1/2}$ for the constant wave speed a in formula (2.9). This way seems to work best in practice, and has a lower computational cost [38]. The methods described here has in [38] been modified to use centered differences instead of Lax-Wendroff, i.e., the numerical flux function is

$$h_{j+1/2} = (1 - w_{j+1/2})h^{upw}_{j+1/2} + w_{j+1/2}h^{c}_{j+1/2}.$$

The scheme then becomes equivalent to one of the simplified variants of the *outer* TVD scheme, which is described in Section 2.2.2.

2.2 Methods based on centered differences

The point of departure for the schemes in this section is the semi-discrete approximation of the conservation law $u_t + f(u)_x = 0$.

$$\frac{du_j}{dt} = -\frac{1}{\Delta x}\Delta_+ h_{j-1/2}$$

where $h_{j+1/2}$ corresponds to a centered difference approximation. Limiters are introduced in the scheme to assure TVD and second order away from extrema. The analysis is done from the semi discrete form

$$\frac{du_j}{dt} = C_{j+1/2}\Delta_+ u_j - D_{j-1/2}\Delta_- u_j$$

where TVD for the semi discrete problem follows from

$$C_{j+1/2} \ge 0 \quad D_{j+1/2} \ge 0$$

in order to make time discretization possible, we also need the bound

$$C_{j+1/2} + D_{j+1/2} < A$$

where A is a constant independent of Δx.

Time discretization is often made with a Runge-Kutta type method. In [27] Runge-Kutta methods are developed which are especially suited for TVD discretizations. The idea in [27] is to write the Runge-Kutta method as a convex sum of forward Euler steps.

2.2.1 Inner TVD, or the MUSCL scheme

The inner TVD schemes, are based on piecewise linear reconstruction of the grid function u_j. The method is often interpreted as a finite volume scheme, so that the grid function is thought of as representing cell averages rather than point values.

The grid function u_j is interpolated to the cell interfaces $x_{j+1/2}$ from the right and from the left through

$$\begin{aligned} u^R_{j+1/2} &= u_{j+1} - \tfrac{1}{2}\psi(\tfrac{1}{r_{j+1}})\Delta_+ u_{j+1} \\ u^L_{j+1/2} &= u_j + \tfrac{1}{2}\psi(r_j)\Delta_- u_j \end{aligned} \qquad (2.12)$$

where r_j is defined as

$$r_j = \frac{\Delta_+ u_j}{\Delta_- u_j}.$$

This is a one sided interpolation, where the limiter $\psi(r)$ is introduced to supress oscillations.

If $h^1(u_{j+1}, u_j)$ is the numerical flux of a first order TVD scheme, the second order inner TVD scheme is defined by

$$h_{j+1/2} = h^1(u^R_{j+1/2}, u^L_{j+1/2})$$

Second order accuracy follows from comparison with the second order flux $f(u_{j+1/2})$, or

$$\begin{aligned}&|f(u_{j+1/2}) - h^2_{j+1/2}| = \\ &|h^1(u_{j+1/2}, u_{j+1/2}) - h^1(u^R_{j+1/2}, u^L_{j+1/2})| \leq \\ &K_1|u^R_{j+1/2} - u_{j+1/2}| + K_2|u^L_{j+1/2} - u_{j+1/2}|.\end{aligned}$$

We require that $u^R_{j+1/2}, u^L_{j+1/2}$ approximate the values at the cell interface $u_{j+1/2}$ to second order accuracy.

The following theorem from [29] gives general conditions on the limiter.

Theorem 2.4. *If the limiter function ψ is Lipschitz continuous and the following holds for all r*

$$\begin{aligned}&\psi(1) = 1 \\ &m \leq \psi(r) \leq M \\ &M + 2 - 2A \leq \tfrac{\psi(r)}{r} \leq 2 + m\end{aligned}$$

for some constants $m > -1, M, A$, then the second order semi discrete method, obtained by putting (2.12) into a first order numerical flux function is second order accurate and TVD if the first order flux corresponds to a monotone scheme.

The TVD domain for $\psi(r)$ is shown in Fig. 2.2. Some examples of limiter functions are

$$\begin{aligned}&\psi(r) = (r + |r|)/(r + 1) \quad \text{(van Leer)} \\ &\psi(r) = (r^2 + r)/(r^2 + 1) \quad \text{(van Albada)} \\ &\psi(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ \min(1, r) & \text{otherwise} \end{cases} \quad \text{(minmod)} \\ &\psi(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ \max(\min(2r, 1), \min(r, 2)) \end{cases} \quad \text{(superbee)} \\ &\psi(r) = (|r|^\alpha + r)/(|r|^\alpha + 1)\end{aligned}$$

The last limiter contains the first two as the special cases $\alpha = 1$ and $\alpha = 2$. The Superbee limiter is very compressive, and especially suited for linear discontinuities. It is however not advisable to use in the nonlinear fields since there is a risk that the entropy condition is violated.

The limiters are usually implemented by using the associated slope function

$$s(x, y) = \psi(x/y)y.$$

With this notation the general α-limiter above can be written

$$s(x, y) = \frac{y|x|^\alpha + x|y|^\alpha}{|x|^\alpha + |y|^\alpha} = \frac{|x|^\alpha}{|x|^\alpha + |y|^\alpha}y + \frac{|y|^\alpha}{|x|^\alpha + |y|^\alpha}x \tag{2.13}$$

we can clearly interpret this as a weighted average of x and y. In this form there is less problems when $y = 0$, but sometimes it is necessary to introduce a parameter ϵ to avoid division by zero. For (2.13), we can write

$$s(x, y) = \frac{|x|^\alpha + \epsilon}{2\epsilon + |x|^\alpha + |y|^\alpha}y + \frac{|y|^\alpha + \epsilon}{2\epsilon + |x|^\alpha + |y|^\alpha}x$$

In general ϵ should be placed such that the scheme becomes fully second order accurate as $x, y \to 0$.

When $\psi(r) = r\psi(1/r)$, then we can interpret the interpolation (2.12) as a piecewise linear reconstruction in each cell, i.e.,

$$u(x) = u_j + s_j(x - x_j)/\Delta x \quad x_{j-1/2} < x < x_{j+1/2}$$

with $s_j = \psi(r_j)\Delta_- u_j$.

It can be seen by Taylor expansion that in the case of a scheme based on cell averages, third order accuracy can be achieved for $\psi'(1) = 2/3$. However, this is a purely one dimensional effect. Third order is not possible for schemes using point values instead of cell averages, unless the flux is linear ($f''(u) = 0$). Examples of limiters satisfying $\psi'(1) = 2/3$ are

$$\begin{aligned}&\psi(r) = \tfrac{4r^2+2r}{3(r^2+1)} \quad \text{(Speijkrese)} \\ &\psi(r) = \tfrac{2r^2+r}{2r^2-r+2} \quad \text{(Koren-Hemker)} \\ &\psi(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ 2r & \text{if } 0 < r \leq 1/4 \\ (2r+1)/3 & 1/4 < r \leq 5/2 \\ 2 & \text{if } 5/2 < r \end{cases}\end{aligned}$$

The last limiter is new, and designed for linear fields, i.e., it is very compressive, and can be used in place of Superbee when third order is desired.

2.2.2 Outer TVD schemes with simplifications

The outer TVD methods can be understood as similar to inner TVD methods, but based on the second order flux

$$h_{j+1/2} = (f_{j+1} + f_j)/2. \tag{2.14}$$

Alternatively, many of the outer TVD methods, can be obtained from the Lax-Wendroff based schemes, by replacing $h^{LW}_{j+1/2}$ by the centered flux (2.14), in the formulas in section 2.1.2.

The original outer TVD method was defined in [23], and can be written as

$$h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2}\psi(r^+_j)(f(u_{j+1}) - h_{j+1/2}) + \tfrac{1}{2}\psi(r^-_{j+1})(f(u_j) - h_{j+1/2}) \tag{2.15}$$

where we use

$$r^+_j = \frac{f(u_j) - h_{j-1/2}}{f(u_{j+1}) - h_{j+1/2}} \quad r^-_j = \frac{f(u_j) - h_{j+1/2}}{f(u_{j-1}) - h_{j-1/2}}.$$

The following theorem is similar to the inner TVD methods.

Theorem 2.5. *If the limiter function ψ is Lipschitz continuous and the following holds for all r*

$$\begin{aligned} &\psi(1) = 1 \\ &m \le \psi(r) \le M \\ &M - 2 \le \tfrac{\psi(r)}{r} \le 2A - 2 + m \end{aligned}$$

for some constants $m, M < 2, A$, then the second order outer semi discrete method, using the flux (2.15), where $h_{j+1/2}$ corresponds to a first order TVD scheme is second order accurate and TVD.

Thus the limiter functions described in the previous subsection can be used here, too.

By using the viscosity form of the first order flux we can write

$$\begin{aligned} r^+_j &= \tfrac{(\lambda a_{j-1/2} + Q_{j-1/2})\Delta_- u_j}{(\lambda a_{j+1/2} + Q_{j+1/2})\Delta_+ u_j} \\ r^-_j &= \tfrac{(-\lambda a_{j+1/2} + Q_{j+1/2})\Delta_+ u_j}{(-\lambda a_{j-1/2} + Q_{j-1/2})\Delta_- u_j}. \end{aligned}$$

By the notation

$$\begin{aligned} d^-_{j+1/2} &= (-a_{j+1/2} + Q_{j+1/2}/\lambda)/2 \\ d^+_{j+1/2} &= (a_{j+1/2} + Q_{j+1/2}/\lambda)/2 \end{aligned}$$

we can write the scheme as

$$h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2}s(d^+_{j-1/2}\Delta_- u_j, d^+_{j+1/2}\Delta_+ u_j) + \tfrac{1}{2}s(d^-_{j+3/2}\Delta_+ u_{j+1}, d^+_{j+1/2}\Delta_+ u_j)$$

where we have rewritten the limiter expressions in terms of the slope limiter function $s(x, y) = \psi(x/y)y$. In practice [35], one uses often the linearized version of this,

$$h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2}d^+_{j+1/2}s(\Delta_- u_j, \Delta_+ u_j) + \tfrac{1}{2}d^-_{j+1/2}s(\Delta_+ u_{j+1}, \Delta_+ u_j) \tag{2.16}$$

since, it has turned out to work better in practical computations with systems of conservation laws. (2.16) has,furthermore, the advantage of a lower computational cost. (2.15) and (2.16) are clearly a type of upwind-TVD schemes. By lumping together the positive and negative parts, using

$$d^+_{j+1/2} + d^-_{j+1/2} = \frac{1}{\lambda}Q_{j+1/2}$$

we obtain the symmetric TVD scheme

$$h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2\lambda}Q_{j+1/2}(s(\Delta_- u_j, \Delta_+ u_j) + s(\Delta_+ u_{j+1}, \Delta_+ u_j) - \Delta_+ u_j) \tag{2.17}$$

by introducing the weight

$$w_{j+1/2} = \psi(r_j) + \psi(1/r_{j+1}) - 1$$

we can write (2.17) as

$$h^2_{j+1/2} = (1 - w_{j+1/2})h_{j+1/2} + w_{j+1/2}h^c_{j+1/2}$$

where $h^c = (f_{j+1} + f_j)/2$. Thus we have obtained a weighted upwind-centered difference method, compare Section 2.1.2.

It is possible to define a non-linear version of (2.17) by redefining

$$r_j = \frac{Q_{j-1/2}\Delta_- u_j}{Q_{j+1/2}\Delta_- u_j} = \frac{(f_j + f_{j-1})/2 - h_{j-1/2}}{(f_{j+1} + f_j)/2 - h_{j+1/2}}.) \tag{2.18}$$

The simplified method (2.17) above has been described in [38], and it has been used in [15] as well, where a clever modification of (2.17) was made. The flux used in [15] has instead the weight

$$w_{j+1/2} = s(1/r_{j+1}, r_j),$$

where $s(x,y)$ is a standard slope limiter function. The limiting is extended over five points. This permits more generous conditions on the limiter function when proving TVD, see [15].

We thus have two non-linearized outer TVD schemes, (2.15) (upwind TVD) and (2.18) (symmetric TVD). We also have linearized versions (2.17) (symmetric TVD) and (2.16) (upwind TVD). Numerical experiments on these methods have shown that

- The upwind versions usually give somewhat smaller error than the symmetric versions.
- The linearized schemes (2.16) and (2.17) give smaller error than (2.15) and (2.18).
- That the non-linearized schemes can give serious problems with oscillations for systems.
- The symmetric schemes can not be made fully second order accurate by the UNO limiter described in section 2.2.3.

2.2.3 The UNO limiter for full second order accuracy

All the second order TVD schemes in the previous section degenerates to first order near smooth extrema. It is described in [12] how this can be overcome by the UNO scheme.

Consider the slope limiter

$$s(\Delta_+ u_j, \Delta_- u_j),$$

which could be dervied from any of the previously described limiters by $s(x,y) = \psi(x/y)y$. This limiter is converted to an UNO limiter by introducing second differences according to

$$\begin{array}{l} s = s(\Delta_+ u_j - \frac{1}{2} m(\Delta_+\Delta_- u_j, \Delta_+\Delta_- u_{j+1}), \\ \Delta_- u_j + \frac{1}{2} m(\Delta_+\Delta_- u_{j-1}, \Delta_+\Delta_- u_j)) \end{array} \qquad (2.19)$$

where $m(x,y)$ is the minmod function. $s(x,y)$ can be any TVD slope limiter. (2.19) is derived by a piecewise parabolic reconstruction, where the minmod function guarantees that no new extrema are created. The UNO limiter was derived for the inner TVD scheme, but works well for most of the upwind-TVD methods described above.

3 Second order TVD schemes for systems

There is a large variety of first order schemes for hyperbolic systems of conservation laws. They are based on approximate Riemann solvers, flux splitting, characteristic decomposition, etc. We will not go into details about this. Instead we will discuss how to generalize the second order TVD schemes described in Section 2. First order schemes give too poor resolution for practical purposes.

All realistic problems require that a grid transformation is included into the equations. This is no fundamental problem, since the transformed equations of the conservation law

$$u_t + f(u)_x + g(u)_y + h(u)_z = 0$$

are in the form

$$u_t + \tilde{f}(u)_\xi + \tilde{g}(u)_\eta + \tilde{h}(u)_\zeta = 0 \qquad (3.1)$$

where (ξ, η, ζ) are the coordinates of a unit cube. The transformed fluxes contain some metric derivatives, which acts as variable coefficients. If only the metric is discretized such that the flux differences become exactly zero for a constant flow, there are usually no additional difficulties from the grid transformation. The discussion below is one dimensional, but applies without changes to the problem of evaluating one coordinate direction flux in (3.1).

3.1 Generalization through eigendecomposition

A linear system $u_t + Au_x = 0$ can be decoupled into several independent scalar problems by a characteristic decomposition. For non-linear problems this is no longer possible, however most generalizations of methods for scalar conservation laws are based on local eigendecomposition. The most well known and widely used method is Roe's linearization [24]. A local Jacobian matrix at the point $x_{j+1/2}$, $A_{j+1/2} = A(u_{j+1}, u_j)$ is constructed such that it satisfies

$$\begin{array}{l} f_{j+1} - f_j = A(u_{j+1}, u_j)(u_{j+1} - u_j) \\ A(u,u) = \frac{\partial f}{\partial u} \end{array}$$

We will use m to denote the number of equations, thus A is an $m \times m$ matrix. In [24] such a matrix is constructed for the compressible Euler equations of gas dynamics. It has the property that $A_{j+1/2} = A(M(u_{j+1}, u_j))$, i.e., it is the value of the Jacobian evaluated at a special average of u_{j+1} and u_j.

Required for generalizing the scalar methods are the eigenvalues of the Jacobian matrix $a^k_{j+1/2}$, $k = 1, \ldots, m$ which corresponds to the wave speeds $a_{j+1/2}$ in the scalar case. The eigenvectors $r^k_{j+1/2}$, which is the coordinate system in which the method is used, and the coefficients $\alpha^k_{j+1/2}$, defined as the representation of $\Delta_+ u_j$ in the eigenbasis,

$$\Delta_+ u_j = \sum_{k=1}^{m} \alpha^k_{j+1/2} r^k_{j+1/2}.$$

The α-coefficients are used in place of $\Delta_+ u_j$ for the scalar problem, in all limiters etc. As an example the first order upwind scheme

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2}|a_{j+1/2}|\Delta_+ u_j$$

is generalized to the first order Roe's method

$$h_{j+1/2} = \frac{1}{2}(f_{j+1}+f_j) - \frac{1}{2}\sum_{k=1}^{m} |a^k_{j+1/2}|\alpha^k_{j+1/2} r^k_{j+1/2} \quad (3.2)$$

Formulas for the compressible Euler equations of gas dynamics can be found in [39] or [35].

An alternative to the Roe decomposition is to fix the eigen basis at the point u_j and transfer all computations to the characteristic coordinates for the matrix $A(u_j)$. This is often used in interpolation and reconstruction algorithms, e.g., in the flux interpolating ENO scheme, or when doing piecewise linear reconstruction in characteristic variables for the inner TVD scheme. As an example, consider the slope limiter function

$$s_j = s(\Delta_+ u_j, \Delta_- u_j)$$

According to the Roe recipe, we would represent this as

$$s_j = \sum_{k=1}^{m} s(\alpha^k_{j+1/2}, \alpha^k_{j-1/2}) r^k_{j+1/2}$$

in the computation of the flux $h_{j+1/2}$. For the flux $h_{j-1/2}$, s_j would probably be represented in the basis for $A_{j-1/2}$. The second method gives instead the representation

$$s_j = \sum_{k=1}^{m} s(\beta_{j+1} - \beta_j, \beta_j - \beta_{j-1}) r(u_j)$$

where the scalars β_k are the characteristic representation of u_k in the basis determined by u_j.

$$\beta_k = l(u_j)^T u_k$$

and where $r(u_j)$ and $l(u_j)$ are the right and left eigenvectors respectively of the Jacobian matrix $A(u_j)$.

There does not exist any analysis concerning the generalization to systems. Instead we make the following comments based on intuition. The advantage with the generalization based on the Roe decomposition is that the $\alpha_{j+1/2}$, always well represent the true characteristic variables. When using $\beta_k = l(u_j)^T u_k$, if the difference between j and k is large, β_k is no longer a good approximation of the local characteristic variable. On the other hand, one can have doubts about comparing $\alpha_{j+1/2}$ and $\alpha_{j-1/2}$, since they belong to two different coordinate systems. Perhaps that is the reason why scaling of the $\alpha_{j+1/2}$ can significantly affect the computation of hypersonic flows [39] [35]. With the β-generalization we are certain to do all computations in the same coordinate system.

3.2 Lax-Wendroff type methods

These methods are almost always generalized through the Roe eigendecomposition. Thus for example, the numerical flux function

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2\lambda}(\lambda a_{j+1/2})^2 \Delta_+ u_j$$

for a scalar conservation law, where $a_{j+1/2} = (f_{j+1} - f_j)/(u_{j+1} - u_j)$ is generalized to

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2\lambda}\sum_{k=1}^{m} (\lambda a^k_{j+1/2})^2 \alpha^k_{j+1/2} r^k_{j+1/2}$$

for systems. Here $a^k_{j+1/2}$ is the kth eigenvalue of the Roe matrix.

Harten's TVD scheme becomes

$$h^M_{j+1/2} = \tfrac{1}{2}(f_{j+1} + f_j) - \tfrac{1}{2\lambda}\textstyle\sum_{k=1}^{m}(\\ Q(\lambda a^k_{j+1/2} + \lambda b^k_{j+1/2})\alpha^k_{j+1/2} + g^k_j + g^k_{j+1}) r^k_{j+1/2}$$

where

$$g_j^k = \frac{1}{2\lambda}\text{minmod}((Q(\lambda a_{j+1/2}^k) - (\lambda a_{j+1/2}^k)^2)\alpha_{j+1/2}^k, \\ (Q(\lambda a_{j-1/2}^k) - (\lambda a_{j-1/2}^k)^2)\alpha_{j-1/2}^k)$$

and where

$$b_{j+1/2}^k = \begin{cases} (g_{j+1}^k - g_j^k)/\alpha_{j+1/2}^k & \text{when } \alpha_{j+1/2}^k \neq 0 \\ 0 & \text{when } \alpha_{j+1/2}^k = 0 \end{cases}$$

These formulas are obtained from the formulas in Section 2.1.2 by a straightforward generalization. All occurencies of $\Delta_+ u_j$ are replaced by $\alpha_{j+1/2}^k$, and all wave speeds $a_{j+1/2}$ are replaced by eigenvalues $a_{j+1/2}^k$. Then the numerical flux is evaluated field by field, and summed together in the eigenvector basis.

The FCT method should for best performance be implemented in the characteristic variables. However, the scheme is implemented in characteristic variables, only in [37], normally the anti-diffusive corrector step is made in the conserved variables.

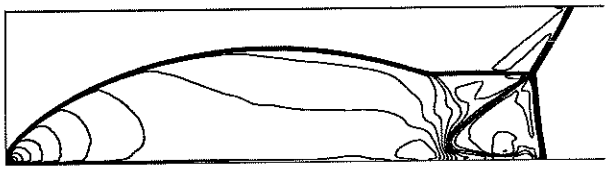

FIG. 3.1. *Minmod limiter in conserved variables.*

3.3 Inner TVD

For the inner TVD scheme, we have to compute the interpolated values on the cell interfaces, $u_{j+1/2}^R, u_{j+1/2}^L$. If we strictly follow the guidlines above, this would mean that the interpolation has to be done in the characteristic variables. However, in order to compute the total flux, a second characteristic decomposition is usually necessary in order to evaluate the first order flux function. It is therefore common to do the interpolation in other variables in order to reduce the cost. The problem is that one sometimes observes oscillations around the contact discontinuities when limiting is done in the conserved variables. For gas dynamics a good compromise is to do the limiting in the density, velocity and pressure variables, which seems to work well in practice.

Some examples of a Mach reflection in compressible gas dynamics are given in Figs. 3.1-3.4. In Fig.3.1 we show the solution using a minmod limiter in the conserved variables. The minmod limiter is diffusive enough to give a good result in these variables. In Fig. 3.2 we show the same computation using the van Leer limiter in pressure-velocity variables. Small oscillations are seen, which dissapears in Fig. 3.3, where the

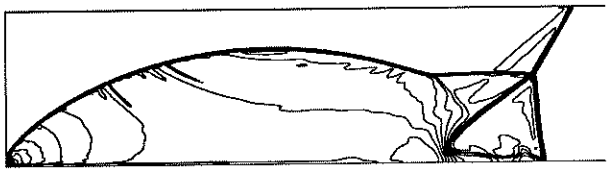

FIG. 3.2. *van Leer limiter in primitive variables.*

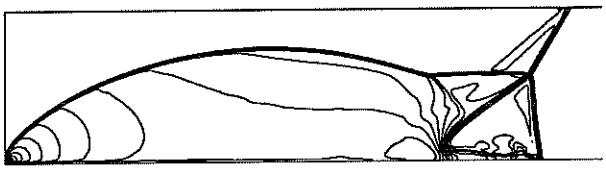

FIG. 3.3. *van Leer limiter in characteristic variables.*

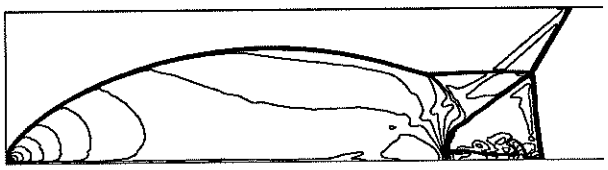

FIG. 3.4. *van Leer limiter in non-linear fields, superbee in linear fields.*

same limiter is used in characteristic variables. When using characteristic variables, we can take the opportunity to use a compressive limiter in the linear fields. Fig. 3.4 shows the superbee limiter applied in the linear characteristic variables. A sharpening of the contact discontinuity is clearly seen. Here the characteristic variables in the limiter were computed in a fixed coordinate system (β decomposition).

The advantage of the inner TVD method, is that it can be built from any first order method. There is a large freedom of choice. Especially in problems which are not strictly non-linear, there will be difficulties to satisfy the entropy condition with many of the more popular first order schemes. In such a case, a scheme based on an exact Riemann solver could be a reasonable choice. Using the inner TVD technique such a scheme would be easy to generalize to second orde accuracy.

3.4 Outer TVD

For the outer TVD scheme, we build on either the non-linear version (2.15) or the linearized (2.16). We here only discuss the upwind limiting. Symmetric limiting is done in exactly the same way. For this method both α and β types of eigendecomposition have been used. The outer TVD method (2.15) can be generalized to systems, by introducing

$$\begin{aligned} d^{-k}_{j+1/2} &= l^k(u_i)^T((f_j - h_{j+1/2}) \\ d^{+k}_{j+1/2} &= l^k(u_i)^T(f_{j+1} - h_{j+1/2})) \end{aligned} \tag{3.3}$$

where $l^k(u_i)$ is the kth left eigenvector of a Jacobian matrix $A(u_i)$. We then define

$$\begin{aligned} &h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2}\textstyle\sum_{k=1}^{m}(\psi(d^{+k}_{j-1/2}/d^{+k}_{j+1/2})d^{+k}_{j+1/2} + \\ &\psi(d^{-k}_{j+3/2}/d^{-k}_{j+1/2})d^{-k}_{j+1/2})r^k_{j+1/2} \end{aligned} \tag{3.4}$$

where the state u_i in (3.3) for the eigenbasis, is an intermediate value $u_{j+1/2}$ for all the d:s appearing in (3.4). Note that, e.g., $d^{+}_{j+1/2}$ thus is linearized differently depending on whether it is used in the computation of $h^2_{j+1/2}$ or $h^2_{j-1/2}$. This is however expensive, since two characteristic decompositions are necessary. One when the first order numerical flux is evaluated, and one for the second order extension.

In [23] the characteristic decomposition is done differently. The linearization is built into the first order method, and it is based on Riemann invariants rather than left eigenvectors.

The more popular Roe decomposition can be used for this scheme, if the first order flux function can be written

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) - \frac{1}{2\lambda}\sum_{k=1}^{m} q^k_{j+1/2}\alpha^k_{j+1/2}r^k_{j+1/2}$$

for some viscosity coefficients $q^k_{j+1/2}$. We then write the scheme (2.15)

$$\begin{aligned} &h^2_{j+1/2} = h_{j+1/2} + \\ &\tfrac{1}{2}\textstyle\sum_{k=1}^{m}(s(d^{+k}_{j-1/2}\alpha^k_{j-1/2}, d^{+k}_{j+1/2}\alpha^k_{j+1/2}) + \\ &s(d^{-k}_{j+3/2}\alpha^k_{j+3/2}, d^{-k}_{j+1/2}\alpha^k_{j+1/2}))r^k_{j+1/2} \end{aligned} \tag{3.5}$$

where we have introduced

$$\begin{aligned} d^{+k}_{j+1/2} &= (a^k_{j+1/2} + q^k_{j+1/2}/\lambda)/2 \\ d^{-k}_{j+1/2} &= (-a^k_{j+1/2} + q^k_{j+1/2}/\lambda)/2 \end{aligned}$$

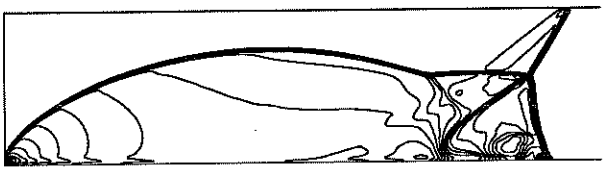

FIG. 3.5. *Symmetric outer TVD with the van Leer limiter.*

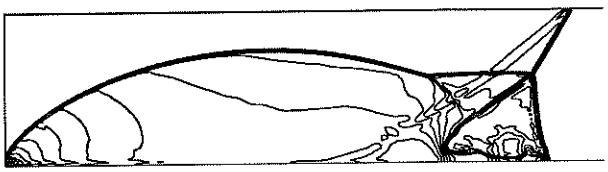

FIG. 3.7. *Upwind outer TVD with the van Leer limiter in the non-linear fields, superbee in linear fields.*

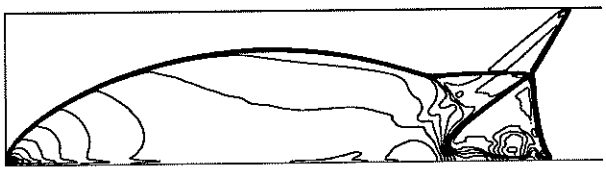

FIG. 3.6. *Upwind outer TVD with the van Leer limiter.*

The linearized (2.16) becomes

$$h^2_{j+1/2} = h_{j+1/2} + \tfrac{1}{2}\sum_{k=1}^{m}(d^{+k}_{j+1/2}s(\alpha^k_{j-1/2},\alpha^k_{j+1/2}) + d^{-k}_{j+1/2}s(\alpha^k_{j+3/2},\alpha^k_{j+1/2}))r^k_{j+1/2}) \tag{3.6}$$

This is a very efficient method. Additional simplifications are done by using the symmetric version (2.17), whose generalization to systems is in analogy with (3.6).

As mentioned previously, the non-linear (3.5) can sometimes give rise to oscillations in the solution, whereas the scheme (3.6) has always shown good performance in practical computations.

Some examples of computations with the scheme (3.6) are shown in Figs. 3.5–3.7. The symmetric outer TVD method in Fig. 3.5 gives almost identical results as the upwind TVD method in Fig. 3.6, when a standard flux limiter is used. Both methods resolve the shock with no spurious oscillations. Fig. 3.7 shows the upwind TVD scheme, but now with an extremely compressive limiter in the linear fields. We see an improvement in the resolution of the contact discontinuity emanating from the triple point.

4 ENO methods

There are several reasons why we would like to increase the accuracy of the TVD schemes previously described. Difference method of higher order of accuracy are usually more efficient, since a fewer number of grid points can be used for the same accuracy. Furthermore, all TVD schemes suffer from degeneracy of accuracy to first order near non-sonic extrema.

Thus, in order to increase the accuracy, we have to abandon the TVD property. We allow a few, very small oscillations, or a very small increase in total variation. Methods with this property are called essentially non-oscillatory (ENO). However, the only instance where such a property has been proved for a uniformly high order method is in [25]. We also mention the piecewise parabolic method (PPM) [5], which is a generalization of the inner TVD scheme to third order. However, PPM degenerates to first order at extrema.

ENO methods are based on an essentially non-oscillatory interpolation procedure. ENO interpolation consists of choosing the interpolation stencil adaptively over a region where the function has smallest variation. This interpolation can then be applied either to the function itself, or to the flux functions. When the in-

terpolation is done on the function, we obtain a method which is a generalization of the inner TVD scheme, and is based on cell averages of the function. When interpolation is done on the fluxes we obtain a generalization of the outer TVD scheme.

The ENO interpolation algorithm is based on Newton's form of the interpolation polynomial. Assume that the function $g(x)$ is known at the points $x_j, j = \ldots, -1, 0, 1, \ldots$. Define the divided differences $[x_i, \ldots, x_{i+r}]g$ recursively by

$$[x_i]g = g(x_i)$$
$$[x_i, \ldots, x_{i+r}]g = \frac{[x_{i+1}, \ldots, x_{i+r}]g - [x_i, \ldots, x_{i+r-1}]g}{x_{i+r} - x_i}$$

Newton's polynomial interpolating g at the points $x_1, \ldots, x_n$ is then given by

$$P^n(x) = \sum_{i=1}^{n} (x - x_1)(x - x_2)\ldots(x - x_{i-1})[x_1, \ldots, x_i]g$$

where $(x - x_i)\ldots(x - x_j) = 1$ if $i > j$. This form is convinient, since if we want to add another point to the interpolation problem, we can immediately update the interpolation polynomial using the formula

$$P^{n+1}(x) = P^n(x) + (x - x_1)\ldots(x - x_n)[x_1, \ldots, x_{n+1}]g$$

We now give an algorithm for constructing a piecewise N degree polynomial continuous interpolant $L(x)$ from the given grid function u_j, with

$$L(x_j) = u_j$$

and which does introduce as small amount of oscillations as possible.

Algorithm 4.1

1. Define the linear polynomial
   $L^1(x) = u_j + (x - x_j)(u_{j+1} - u_j)/\Delta x$
   $x_j \le x < x_{j+1}$
   and indices to bookeep the stencil width
   $k^1_{min} = j \quad k^1_{max} = j + 1$
2. **for** $p = 2$ **to** N **do**
   $a_p = [x_{k^{p-1}_{min}}, \ldots, x_{k^{p-1}_{max}+1}]u$
   $b_p = [x_{k^{p-1}_{min}-1}, \ldots, x_{k^{p-1}_{max}}]u$
   if $|a_p| < |b_p|$ **then**
   $L^p(x) = L^{p-1}(x) + a_p \prod_{k=k^{p-1}_{min}}^{k^{p-1}_{max}} (x - x_k)$
   $k^p_{max} = k^{p-1}_{max} + 1 \quad k^p_{min} = k^{p-1}_{min}$
   else
   $L^p(x) = L^{p-1}(x) + b_p \prod_{k=k^{p-1}_{min}}^{k^{p-1}_{max}} (x - x_k)$
   $k^p_{max} = k^{p-1}_{max} \quad k^p_{min} = k^{p-1}_{min} - 1$
   endif

We thus add one point to the right for a_p and one point to the left for b_p. Next use the smallest difference to update the polynomial.

The numerical approximation u^n_j at (t_n, x_j) can be thought of as an approximation of the point value $u(t_n, x_j)$. Alternatively, we introduce the *cells*, c_j as

$$c_j = \{x | x_{j-1/2} \le x \le x_{j+1/2}\}$$

where $x_{j+1/2} = (x_j + x_{j+1})/2$, and view u^n_j as an approximation to the cell average

$$\frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(t_n, x)\, dx.$$

The situation is depicted in Fig. 4.1.

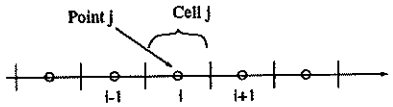

Fig. 4.1. Grid cells and grid points.

The distinction between these two views is not important for methods with accuracy ≤ 2, since

$$u(t_n, x_j) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(t_n, x)\, dx + \mathcal{O}(\Delta x^2)$$

We will here treat higher order of accuracy than two. We consider only the semi-discrete problem. Time discretization can be done by a Runge-Kutta method, as described in [27].

4.1 Finite volume ENO

The *cell average* based higher order schemes are the generalization of the inner TVD schemes described in Section 2.2.1. The schemes starts from the following exact formula for the cell average. Integrate

$$u_t + f(u)_x = 0$$

with respect to x over one cell at t. The result is

$$\frac{d}{dt}\frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(t,x)\,dx + \frac{f(u(t,x_{j+1/2}))-f(u(t,x_{j-1/2}))}{\Delta x} = 0 \tag{4.1}$$

Compare this with the numerical approximation

$$\frac{du_j}{dt} + \frac{h_{j+1/2}-h_{j-1/2}}{\Delta x} = 0 \tag{4.2}$$

If the numerical flux approximates the flux of the exact solution at the cell interface

$$h_{j+1/2} = f(u(t,x_{j+1/2})) + \mathcal{O}(\Delta x^p)$$

then (4.2) is a p th order approximation of the PDE in terms of its *cell averages.*

One usual way to find higher order approximations is to make a piecewise polynomial approximation, $L(x)$ of $u(t_n,x)$ from the given cell averages u_j^n. Inside each cell $u(t_n,x)$ is approximated by a polynomial, and at the cell interfaces, $x_{j+1/2}$, there may be jumps. From this piecewise polynomial the numerical flux is obtained as $h(u_{j+1/2}^R, u_{j+1/2}^L)$, where $h(u_{j+1},u_j)$ is the numerical flux of a first order TVD method and the end values are

$$u_{j+1/2}^R = \lim_{x\to x_{j+1/2}+} L(x)$$
$$u_{j+1/2}^L = \lim_{x\to x_{j+1/2}-} L(x)$$

The grid function u_j^n is given as cell averages, but the reconstruction gives the function itself. Thus the ENO interpolation is not applied directly to u_j^n. To overcome this difficulty, there are two different variants of the method. In the first method, reconstruction by primitive function, we observe that the primitive function

$$U(x_{j+1/2}) = \int_{-\infty}^{x_{j+1/2}} u(t_n,x)\,dx = \sum_{k=-\infty}^{j} u_k^n \Delta x$$

is known at the points $x_{j+1/2}$. The function $U(x)$ is interpolated using Algorithm 4.1. The interpolation polynomial, $L(x)$, is differentiated to get the approximation to $u(t_n,x)$. Thus the left and right values required in the numerical flux are

$$u_{j+1/2}^L = \frac{dL(x_{j+1/2}-)}{dx}$$
$$u_{j+1/2}^R = \frac{dL(x_{j+1/2}+)}{dx}$$

$L(x)$ is continuous, but the derivatives may have different values from the left and from the right at the break points $x_{j+1/2}$.

The second variant is the so called reconstruction by deconvolution, which is based on the formula

$$\overline{u}(x) = \frac{1}{\Delta x}\int_{x-\Delta x/2}^{x+\Delta x/2} u(y)\,dy = \int_{-1/2}^{1/2} u(x+s\Delta x)\,ds \tag{4.3}$$

where thus $\overline{u}(x)$ is the cell average. We interpolate the given cell averages, using Algorithm 4.1, to get an approximation of $\overline{u}(x)$, and then find the approximation of $u(x)$ by inverting ("deconvolute") (4.3). Details can be found in [13].

4.2 Flux ENO

The *point value* based higher order methods starts from the observation that if

$$f(u(x_j)) = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} F(x)\,dx$$

for some function $F(x)$, then

$$f(u(x_j))_x = \frac{F(x_{j+1/2})-F(x_{j-1/2})}{\Delta x}$$

and thus if the numerical flux satisfies

$$h_{j+1/2} = F(x_{j+1/2}) + \mathcal{O}(\Delta x^p)$$

the scheme (4.2) is p th order accurate in terms of point values. The function $F(x)$ can be obtained by interpolation of the grid function

$$G_{j+1/2} = \int_a^{x_{j+1/2}} F(x)\,dx = \sum_{k=a}^{j} f(u_k)\Delta x$$

and then taking the derivative of the interpolation polynomial, $F(x) = dG(x)/dx$. We thus form the interpolant of

$$H_{j+1/2} = \Delta x \sum_{k=a}^{j} f(u_k)$$

by using the Algorithm 4.1, and then define the numerical flux as

$$h_{j+1/2} = \frac{dH(x_{j+1/2})}{dx}.$$

The interpolation is made piecewise polynomial with break points x_j. This direct approach have to be modified somewhat. If we carry out the above scheme we get

$$H^1(x) = \begin{cases} H_{j+1/2} + (x - x_{j+1/2})f_j & \text{if } |f_j| < |f_{j+1}| \\ H_{j+1/2} + (x - x_{j+1/2})f_{j+1} & \text{if } |f_{j+1}| < |f_j| \end{cases}$$

on the interval $x_j < x < x_{j+1}$, which leads to

$$h_{j+1/2} = \begin{cases} f_j & \text{if } |f_j| < |f_{j+1}| \\ f_{j+1} & \text{if } |f_{j+1}| < |f_j| \end{cases}$$

for first order of accuracy. Although this flux is consistent, the resulting method is not TVD. It is crucial that the first order approximation is TVD. From numerical experiments, it is possible to verify that this method is not non oscillatory no matter how high the accuracy of the interpolant. Instead we make the first order version of this method TVD, by taking

$$H^1(x) = \begin{cases} H_{j+1/2} + (x - x_{j+1/2})f_j & \text{if } a_{j+1/2} \geq 0 \\ H_{j+1/2} + (x - x_{j+1/2})f_{j+1} & \text{if } a_{j+1/2} < 0 \end{cases} \tag{4.4}$$

on the interval $x_j < x < x_{j+1}$. The first order method is then the upwind scheme. Continuing the interpolation to higher order leads to a non oscillatory high order scheme.

We obtain a more general way of choosing the starting first order polynomial if we consider a first order TVD flux $h_{j+1/2}$ and split it as

$$h_{j+1/2} = f_j^+ + f_{j+1}^-$$

where f^+ corresponds to positive wave speeds and f^- to negative wave speeds. As an example the Engquist-Osher scheme can be written on this form. Another example is the Lax-Friedrichs scheme, or the modified Lax-Friedrichs scheme where

$$f^+(u) = (f(u) + \alpha u)/2$$
$$f^-(u) = (f(u) - \alpha u)/2$$

with $\alpha = \max|f'(u)|$.

We define the starting polynomials

$$H_-^1(x) = H_{j+1/2} + (x - x_{j+1/2})f_{j+1}^-$$
$$H_+^1(x) = H_{j+1/2} + (x - x_{j+1/2})f_j^+ \quad x_j < x < x_{j+1}$$

and then continue the ENO interpolation of f^+ and f^- respectively through the points x_j to arbitrary order of accuracy, p. Finally

$$h_{j+1/2} = \frac{dH_+^p(x_{j+1/2})}{dx} + \frac{dH_-^p(x_{j+1/2})}{dx}$$

The truncation error for this method will involve differences of the functions f^+ and f^-. Thus to achieve the expected accuracy it is necessary to have $f^+, f^- \in C^p$, p large enough. Because of this, the scheme has mostly been used together with the C^∞ Lax-Friedrichs numerical flux, or the modified Lax-Friedrichs numerical flux. However the Lax-Friedrichs scheme does not always give sufficient shock resolution. Although the higher order versions, obtained as described above, perform much better than the first order Lax-Friedrichs, there is still need for first order TVD methods giving better shock resolution than Lax-Friedrichs and having more derivatives than the upwind or the Engquist-Osher schemes, to be used as building blocks for this method.

The flux ENO method is described in [27], [28].

4.3 Comparison between finite volume ENO and flux ENO

One major difference between the two ENO methods is the treatment of several space dimensions. The point based algorithm is much easier to generalize to more than one space dimension.

For the problem

$$u_t + f(u)_x + g(u)_y = 0$$

the point ENO method can be applied separately in the x- and y- directions to approximate $\partial/\partial x$ and $\partial/\partial y$ respectively. There are no extra complications.

For the cell centered scheme, the two dimensional generalization of formula (4.1) gives an integral around the cell boundary. This integral is required to p th order accuracy, which can be done by a numerical quadrature formula. If, e.g., $p = 4$ this means using two values on each cell side. Thus for each cell, we need a two dimensional reconstruction, which is a non trivial problem in its own right, and then we have 8 flux evaluations to make, two on each side. The cell centered

scheme quickly becomes more computationally expensive than the point centered scheme. However, when no smooth grid transformation is available, the finite volume ENO method maintains full accuracy, whereas the flux ENO requires a smooth grid transformation. Thus the finite volume ENO method is of importance for computations on unstructured grids.

A high formal order of accuracy does not necessarily lead to a high convergence rate. It has been observed, e.g., in [26], that third or fourth order ENO schemes can lead to a first order convergence rate. The explanation is that the ENO interpolation takes the stencil from the direction where the function has smallest variation, however this direction could be the downwind direction, which then leads locally to an unstable approximation. In the stencil-mix of the ENO scheme, stable stencils has to be used sufficiently often. In order to assure this, the modified ENO method in [26] introduces a tunable parameter which give more weight to the stable stencils. With the modified ENO method the expected convergence rate is achieved in numerical computations. However no proof on the convergence of ENO methods exist.

A description of the finite volume ENO method in several space dimensions is given in [2]. A comparison between the different methods can be found in [3].

4.4 ENO for systems

The implementation of the finite volume ENO method for system is analogous to the inner TVD schemes. The reconstruction has to be done in some variables. Usually the characteristic variables are used. In computations, oscillations can appear around contact discontinuities with reconstruction in conserved or primitive variables. The reconstruction is done field by field, and the result is given as input to a first order TVD scheme.

The flux ENO method is usually implemented in a locally fixed characteristic coordinate system, in analogy with the method for the outer TVD method given in (3.3).

The advantage of using the ENO scheme is clearly seen in problems with extrema in the solution. The degeneracy of second order TVD methods to first order at extrema is then a severe problem. We present results from solving the one dimensional Euler equations of compressible gas dynamics. The initial data consist of a constant state, a discontinuity, and an oscillatory density distribution on the other side of the discontinuity. The set up is described in [28].

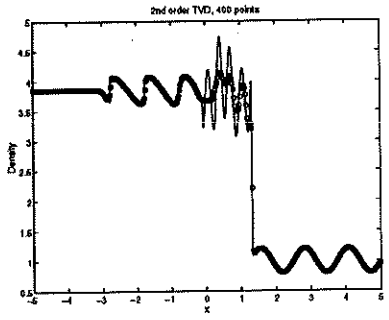

FIG. 4.2. *Second order TVD method.*

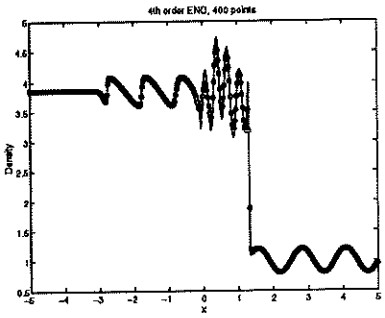

FIG. 4.3. *Fourth order ENO method.*

Fig. 4.2 shows the result from using a second order TVD method. The solid line is a refined solution, which can be considered as exact. The circles show the numerical solution using 400 grid points. The TVD method clips extrema, whereas a computation with a fourth order accurate ENO method, shown in Fig. 4.3, gives a considerably better resolution.

5 Convergence Theory

The mathematically rigorous convergence of nonlinear conservation laws is quite limited. Even if most of the results are scalar problems and rather special classes of algorithms the theory has had an important impact on scheme design. The importance of conservation form, of controlling the total variation and entropy conditions has played a role in the development of higher

order shock capturing methods. We shall here briefly mention some recent results.

The most complete convergence theory is developed for scalar equations and first order methods. The convergence result for monotone schemes discussed in the introduction can be slightly generalized to E-schemes. For these schemes the monotonicity property is replaced by a cell entropy inequality, [9], [32].

There are also convergence results for some classes of methods which are formally of higher order than first. The proofs usually contain a compactness argument and no convergence rate is derived. An important recent paper by Lions and Souganidis, [20], contains convergence proof for second order MUSCL schemes approximating scalar conservation laws in one space dimension.

With specific knowledge about the structure of the solution the error estimates can sometimes be improved. The smooth case, with Strang's result of optimal rate of convergence, was already mentioned in the introduction, [30]. The simplest discontinuous solution is a shock wave with constant states on both sides. There are a few theorems on existence and stability of discrete shock profiles for dissipative finite difference methods, [16], [21]. These results are for one space dimension. In terms of convergence these theorems show that there are approximations in the piece-wise constant shock case with $\mathcal{O}(1)$ errors close to the shock, which converge point-wise faster than algebraically in Δt, at a positive distance from the discontinuity.

In [8] this analysis is generalized to some special cases with isolated shocks and variable states. For the Lax Wendroff scheme and the Burgers flux the optimal $\mathcal{O}(\Delta t^2)$ point-wise convergence is proved at a distance at least $C\log(\Delta x)$ away from the shock.

Even if the numerical method is formally of more than first order and the approximation converges, the rate may still be only first order behind the shock. This can happen for systems where one characteristic may propagate part of the error at a shock into the smooth domain.

As an example of this we show in Fig. 5.1 below the density in a steady state solution of the compressible nozzle flow equations. These equations are on the form

$$u_t + f(u)_x = g(x,u)$$

The vector u consists of the components density momentum and energy. In Fig. 5.2 we show the error in the momentum component, when it has been computed by a third order accurate ENO method, at steady state.

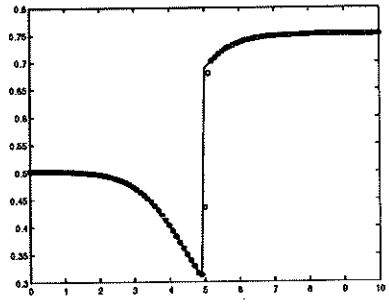

FIG. 5.1. *Density in the nozzle problem.*

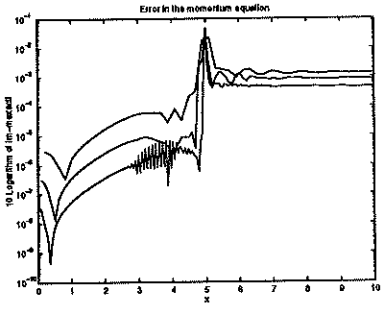

FIG. 5.2 *Error in momentum.*

The solution has been computed on three grids of successive refinements. We see how the third order convergence in front of the shock becomes first order after the shock. In a one dimensional conservation law without lower order terms, the states to the left and to the right of a shock are determined by global conservation, which is computed exactly by a conservative method. For the nozzle flow equations, the low order term causes numerical errors in the conservation relation, which leads to the poor convergence rate.

6 Implementation

In this section we will simplify the notations by writing $r_{j+1/2}$ for the matrix of eigenvectors of the Roe matrix at $x_{j+1/2}$. Similarly $a_{j+1/2}$ will always denote the eigenvalues and $\alpha_{j+1/2}$ the coefficients in the eigenbasis of $\Delta_+ u_j$, as described in Section 3.1.

We first consider the case of one space dimension. All algorithms described will be such that the numerical flux is evaluated at each point, and then added and subtracted to the residual according to Algorithm 6.1.

Algorithm 6.1

for $j := 1, N$
 $res_j := 0$
endfor
for $j := 1, N-1$
 compute $h_{j+1/2}$
 Add to residual:
 $res_j := res_j - h_{j+1/2}$
 $res_{j+1} := res_{j+1} + h_{j+1/2}$
endfor

The problem then boils down to how to compute the numerical flux function. In the algorithms below the add and subtract to the residual is not written out, but implicitly understood. One could evaluate each numerical flux function $h_{j+1/2}$ in a separate code segment, such as an inlined subroutine. This works fine for first order TVD schemes. E.g., for Roe's method we compute the eigendecomposition at $x_{j+1/2}$ and use (3.2) to obtain the flux $h_{j+1/2}$.

However, for second order TVD methods this is ineffcient, since many quantities are common to several fluxes. E.g., the coefficients $\alpha^k_{j+1/2}$ are used in limiters $s(\alpha_{j+1/2}, \alpha_{j-1/2})$ in both $h_{j+1/2}$ and $h_{j-1/2}$. We do not want to do the eigendecomposition more than once for each cell interface. Similarly, for the inner TVD scheme, the slope s_j is required in two neigboring fluxes. We do not want to compute it twice.

The problem is thus that in the computation of $h_{j+1/2}$, the coefficient $\alpha_{j+3/2}$ is needed. We only want to do the eigendecomposition once at each point $x_{j+1/2}$. Thus the eigendecomposition has to be one step ahead, as shown in Algorithm 6.2. All the algorithms in this section can be applied to the inner TVD schemes as well, with the difference that instead of the variable $\alpha_{j+3/2}$, it is the slope s_{j+1} that has to be evaluated one step ahead.

Algorithm 6.2.

compute eigendecomposition at 3/2,
al :=$\alpha_{3/2}$, r:=$r_{3/2}$, a:=$a_{3/2}$
for $j := 1, N-1$
 compute eigendecomposition at $j + 3/2$,
 alp :=$\alpha_{j+3/2}$, rp:=$r_{j+3/2}$, ap:=$a_{j+3/2}$
 compute $h_{j+1/2}$ from alp, al, r, and, a
 al:=alp, r:=rp, a:=ap
endfor

This implementation is memory efficient, and would work well on a superscalar processor. One drawback is that an initial extra step is required. This leads to more serious complications in two and three space dimensions. Furthermore, the loop does not vectorize due to the wrap-around variables. To make the coding simpler, and to make it vectorizes, the loop can be split into two such that the eigendecomposition is made once in each point first, and the flux computation in a second loop. This is shown in Algorithm 6.3.

Algorithm 6.3.

for $j := 1, N-1$
 compute eigendecomposition at $j + 1/2$,
 $\alpha_{j+1/2}$, $r_{j+1/2}$, and $a_{j+1/2}$
endfor
for $j := 1, N-1$
 compute $h_{j+1/2}$ from $\alpha_{j+1/2}$, $\alpha_{j+3/2}$,
 $r_{j+1/2}$, and $a_{j+1/2}$.
endfor

This is very suitable for a vector machine. We use more memory than in Algorithm 6.2, but we can limit the extra work space to one dimension. In three dimensions, one can generalize Algorithm 6.3 to Algorithm 6.4 below.

Algorithm 6.4.

for $k := 1, N_k$
 for $j := 1, N_j$
  for $i := 1, N_i - 1$
   compute eigendecomposition at $i + 1/2, j, k$,
   $\alpha_{i+1/2,j,k}$, $r_{i+1/2,j,k}$, and $a_{i+1/2,j,k}$
  endfor
  for $i := 1, N_i - 1$
   compute $h_{i+1/2,j,k}$ from
 $\alpha_{i+1/2,j,k}$, $\alpha_{i+3/2,j,k}$,
 $r_{i+1/2,j,k}$, and $a_{i+1/2,j,k}$.
  endfor
 endfor
endfor

Algorithm 6.4 computes the i-direction fluxes. In order to compute the entire residual, it is thus necessary to have three subroutines, one for each coordinate direction which are called in sequence. However, it is possible to make the code general, such that the same subroutine can be called three times, once for each coordinate direction. Algorithm 6.5 shows how one can accomplish this, by using the index relation

$$ind = (i-1) + n_i(j-1) + n_i n_j(k-1) + 1$$

which maps the triple $(i,j,k) \in [1..n_i, 1..n_j, 1..n_k]$ onto the one dimensional index space $1..n_i n_j n_k$.

Algorithm 6.5.

if i-fluxes **then**
 $N_{i1} := N_k$, $N_{i2} := N_j$, $N_{i3} := N_i$
 s1:=$N_j N_i$ s2:=$N_i - 1$ s3:=1
elseif j-fluxes **then**
 $N_{i1} := N_k$, $N_{i2} := N_i$, $N_{i3} := N_j$
 s1:=$N_j N_i$ s2:=1 s3:=$N_i - 1$
else
 $N_{i1} := N_i$, $N_{i2} := N_j$, $N_{i3} := N_k$
 s1:=1 s2:=$N_i - 1$ s3:=$N_j N_i$
endif
b := $-N_i - N_i N_j$
for $i1 := 1, N_{i1}$
 for $i2 := 1, N_{i2}$
  for $i3 := 1, N_{i3} - 1$
   ind := s1*i1 + s2*i2 + s3*i3 + b
   indp := s1*i1 + s2*i2 + s3*(i3+1) + b
   compute eigendecomposition
   between u_{ind} and u_{indp},
   al(i3):=α, r(i3):= r, a(i3):=a
  endfor
  for $i3 := 1, N_{i3} - 1$
   compute the numerical flux h from
   al(i3), al(i3+1), r(i3), a(i3)
  endfor
 endfor
endfor

The methodology in Algorithm 6.5 is very convenient to use, when implementing the ENO scheme, since we obtain linewise vectorization. The work space is one dimensional, and we can waste memory in order to make a simple implementation.

This technique is not as favorable on a RISC processor, since the memory is swept through twice in the inner loop, and three times for the coordinate directions. There will be more cache misses than necessary. A more efficient algorithm for superscalar processors, would be to merge all three flux computations into one, as described in Algorithm 6.6.

Algorithm 6.6.

for $j := 1, N_j$
 for $i := 1, N_i$
  compute eigendecomposition at $(i, j, 3/2)$
  alk(i,j) = $\alpha^k_{i,j,3/2}$, etc.
 endfor
endfor
for $k := 1, N_k$
 for $i = 1, N_i$
  compute eigendecomposition at $(i, 3/2, k)$
  alj(i) = $\alpha^j_{i,3/2,k}$
 endfor
 for $j := 1, N_j$
  for $i := 1, N_i$
   compute eigendecomposition for $(i + 3/2, j, k)$,
 alip := $\alpha^i_{i+3/2,j,k}$
   compute $h^i_{i+1/2,j,k}$, using ali, alip
   compute eigendecomposition for $(i, j + 3/2, k)$,
 aljp(i) := $\alpha^j_{i,j+3/2,k}$
   compute $h^j_{i,j+1/2,k}$, using alj, aljp
   compute eigendecomposition for $(i, j, k + 3/2)$,
 alkp(i,j) := $\alpha^k_{i,j,k+3/2}$
   compute $h^k_{i,j,k+1/2}$, using alkp, alk
  ali:=alip
  endfor
  for $i := 1, N_i$
   alj(i) = aljp(i)
  endfor
 endfor
 for $j = 1, N_j$
  for $i = 1, N_i$
   alk(i,j) = alkp(i,j)
  endfor
 endfor
endfor

In Algorithm 6.6, we only write out how the variables α are computed and stored. The other eigendecompo-

sition quantities $r_{j+1/2}, a_{j+1/2}$ are of course evaluated and kept in storage together with $\alpha_{j+1/2}$.

In order to obtain optimal performance, blocking has to be added to Algorithm 6.6. I.e., it is necessary to divide the computational domain into smaller cubes, each of which will fit into the cache memory on the given computer. The Algorithm is then executed once over each subcube.

It would be desirable to have an automatic tool for loop fusion, which could take Algorithm 6.4 or Algorithm 6.5, and merge the three coordinate sweeps into one loop.

Finally, we remark that the formulas for eigenvector multiplication can be hand optimized, by observing that some elements of the eigenvectors for compressible fluid flow are zero or one, some operations in the matrix vector multiplications of type

$$r(1,1)*c(1)+r(1,2)*c(2)+r(1,3)*c(3)+r(1,4)*c(4) \tag{6.1}$$

can be disposed of. Expressions of the type (6.1) arises from the sum in the expression (3.5). A good compiler can do this optimization, and it is not clear how much improvement can be gained. An advantage of not doing the hand optimization, is that we can then consider the multiplication (6.1) as a SAXPY operation, even though this means more arithmetic operations, this could pay on a vector machine, which is very fast on SAXPY's. Note however that RISC processors are not particularly well adapted to SAXPY operations.

6.1 An implementation of the flux ENO method

We describe an efficient implementation of the ENO method. The particular method described in Algorithm 6.7 below is the method defined by equation (4.4). It is known as the ENO/ROE method, and is described in [28]. ENO interpolation is done on the flux functions in a fixed coordinate system.

In the algorithm below we keep the flux differences in a table, $tab(m,s,i) = \Delta_+^{s-1} f_i$ where m is the component of the PDE, e.g., density, momentum, energy, in a characteristic coordinate system fixed at $x_{i+1/2}$. The difference table is only updated with new information when the stencil becomes wider. In this way we only need the characteristic decomposition for the points that are actually used in the adaptive stencil. The coefficients in the interpolation polynomial are precomputed to save aritmetic operations.

Algorithm 6.7.

Computes numerical fluxes of the ENO/ROE method. r is the order of accuracy.

Compute the coefficients $cof(m,k)$.
for $i := 1, N$
 compute the flux function, f_i
endfor
for $i := 1, N-1$
 compute the Roe decomposition, and store a
 factorized form the left eigenvector matrix $L_{i+1/2}$
 such that $L^{-1}_{i+1/2} v$ is inexpensive to compute.
for $m := 1, 3$
 if $a^m_{i+1/2} > 0$ **then**
  $kmin := i$
 else
  $kmin := i+1$
 endif
 $pol(m) = -\frac{1}{2}|a^m_{i+1/2}|\alpha^m_{i+1/2}$
 for $l = 2, r$
  if $kmin \le tableft$ **then**
   $tab(:,2,kmin-1) = L^{-1}_{i+1/2}(f_{kmin} - f_{kmin-1})$
   $tableft = kmin - 1$
  endif
  for $s = 2, l-1$
   $tab(m, s+1, kmin-1) :=$
    $tab(m,s,kmin) - tab(m,s,kmin-1)$
  endfor
  $kmax := kmin + l$
  if $kmax > tabright$ **then**
   $tab(:,2,kmax) = L^{-1}_{i+1/2}(f_{kmax+1} - f_{kmax})$
   $tabright = kmax$
  endif
  for $s = 2, l-1$
   $tab(m, s+1, kmax-s+1) :=$
    $tab(m,s,kmax-s+2) - tab(m,s,kmax-s+1)$
  endfor
  if weighting for modified ENO is
  required, do it here.
  if $|tab(m,l,kmin-1)| < tab(m,l,kmin)|$ **then**

$pol(m) := pol(m) +$
$cof(l-1, kmin) * tab(m, l, kmin-1)$
$kmin := kmin - 1$
else
$pol(m) := pol(m) +$
$cof(l-1, kmin) * tab(m, l, kmin)$
endif
endfor
endfor
Use the matrix $L_{i+1/2}$ to transform
back $pol(1), pol(2), pol(3)$ to standard variables:
$h_{i+1/2} = (f_{i+1} + f_i)/2 + R_{i+1/2} pol$
endfor

In Algorithm 6.7, we use the notations of Section 3.1. The eigenvalues of the Roe matrix are denoted $a^k_{j+1/2}$, the coefficients of $\Delta_+ u_j$ in the eigenvector basis are denoted $\alpha^k_{j+1/2}$, etc..

7 On line information

Some software, and additional lecture notes are available through ftp by the command
`ftp ftp.tdb.uu.se`

log in as anonymous, and do

`cd pub/numerical/tvd`

The following files can be obtained

`tvdnotes.tar.gz` - A set of lecture notes giving more details and proofs of some of the theorems given here.
`cfdlib.tar.gz` - A set of solver routines for the compressible Euler in two space dimensions on curvilinear grids. An implementation of several TVD methods.
`xeuler1d.tar.gz` - A demo program which solves one dimensional Riemann problems in compressible gas dynamics. The program is written for XWindows using the motif widget set.

References

[1] J.P. Boris and D.L. Book (1973) Flux-Corrected Transport I.SHASTA, A Fluid Transport Algorithm That Works, J. Comp. Phys., vol.11 pp.38-69

[2] J. Casper (1992) Finite-Volume Implementation of High-Order Essentially Nonoscillatory Schemes in Two Dimensions, AIAA journal, vol.30 pp.2829-2835

[3] J. Casper, C-W. Shu, and H. Atkins, A Comparison of Two Formulations for High-Order Accurate Essentially Non-Oscillatory Schemes, AIAA 93-3338-CP

[4] P. Colella (1982) Glimm's Method for Gas Dynamics, SIAM J. Sci. Stat. Comput., vol.3 pp.76-110

[5] P. Colella and P. Woodward (1984) The Piecewise-Parabolic Method for Gas-dynamical Simulations, J. Comp. Phys., vol.54 pp.174-201

[6] M. Crandall and A. Majda (1980) Monotone Difference Approximations for Scalar Conservation Laws, Math. Comp., vol.34 pp.1-21

[7] S.
Davis (1988) Simplified Second-Order Godunov-Type Methods, SIAM J. Sci. Stat. Comput., vol.9 pp.445-473

[8] B. Engquist and S.-H. Yu (1995) Convergence of Discrete Shock Profiles to Piecewise Smooth Solutions, to appear.

[9] E. Godlewski and P.-A.. Raviart (1991) Hyperbolic Systems of Conservation Laws, Mathematique and Applications, No. 3/4, Ellipses Publ.

[10] A. Harten (1977) The Artificial Compression Method for Computation of Shocks and Contact Discontinuities, Comm. Pure Appl. Math., vol.30 pp.611-638

[11] A. Harten High Resolution Schemes for Hyperbolic Conservation Laws, J. Comput. Phys., vol.49 pp.357-393

[12] A. Harten and S. Osher Uniformly High-Order Nonoscillatory Schemes I, SIAM J. Numer. Anal., 24 pp.279-309

[13] A. Harten, S. Osher, B. Engquist, and S. Chakravarthy (1986) Some Results on Uniformly High-Order Accurate Essentially Nonoscillatory Schemes, Applied Numerical Mathematics, vol.2 pp.347-377

[14] A. Jameson (1988) Computational Transonics, Comm. Pure Appl. Math., vol.41 pp.507-549

[15] A. Jameson, Artifical Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows, AIAA paper 93-3359, AIAA 11th Computational Fluid Dynamics Conference, Orlando FL, 1993.

[16] G. Jennings (1974) Discrete Shocks, Comm. Pure Appl. Math., vol. 27, pp. 25-37.

[17] N.N. Kuznetsov (1976) Accuracy of Some Approximate Methods for Computing the Weak Solutions of a First-Order Quasi-Linear Equation, USSR Comp. Math. and Math. Phys. vol.16, no. 6, pp.105-119.

[18] P. Lax and B. Wendroff (1960) Systems of Conservation Laws, Comm. Pure Appl. Math, vol.13 pp.217-237

[19] A.LeRoux (1977) A numerical conception of entropy for quasi-linear equations, Math. Comp., vol.31 pp.848-872

[20] P.-L. Lions and E. Souganides (1995) Convergence of MUSCL and Filtered Schemes for Scalar Conservation Laws and Hamilton-Jacobi Equations, to appear.

[21] A. Majda and J. Ralston (1979) Discrete Shock Profiles for Systems of Conservation Laws, Comm. Pure Appl. Math, vol.32 pp.445-482

[22] S. Osher (1984) Riemann Solvers, the Entropy Condition, and Difference Approximations, SIAM J. Numer. Anal., vol.21 pp.217-235

[23] S. Osher and S. Chakravarthy (1984) High resolution schemes and the entropy condition, SIAM J. Numer. Anal., vol.21 pp.955-984

[24] P.L. Roe (1981) Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, J. Comput. Phys., vol.43 pp.357-372

[25] R. Sanders (1988) A Third-Order Accurate Variation Nonexpansive Difference Scheme for Single Nonlinear Conservation Laws, Math. Comp., vol.51 pp.535-558

[26] C-W.Shu (1990) Numerical Experiments on the Accuracy of ENO and Modified ENO schemes, ICASE Report No. 90-55

[27] C-W. Shu and S. Osher (1988) Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, J. Comput. Phys., vol.77 pp.439-471

[28] C-W. Shu and S. Osher (1989) Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes II, J. Comput. Phys., vol.83 pp.32-78

[29] S. Spekreijse (1987) Multigrid solution of Monotone Second-Order Discretizations of Hyperbolic Conservations Laws, Math. Comp., vol.49 pp.135-155

[30] W. G. Strang (1964) Accurate Partial Difference Methods II: Non-linear Problems, Numer. Math., vol.6 p.37

[31] P. Sweby, High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws, SIAM J. Numer. Anal., vol.21 pp.995-1010

[32] E. Tadmor (1987) The Numerical Viscosity of Entropy Stable Schemes for Systems of Conservativation Laws I, Math. Comp., vol.49 pp.91-104

[33] B. van Leer (1984) On the Relation Between the Upwind-Differencing Schemes of Godunov, Engquist-Osher and Roe, SIAM J. Sci. Stat. Comput., vol.5 pp.1-20

[34] B. van Leer (1979) Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov's method, J. Comput. Phys., vol.32 pp.101-136

[35] Y. Wada and H. Kubota (1992) Numerical Simulation of Re-Entry Flow Around the Space Shuttle with Finite-Rate Chemistry, J. Aircraft, vol.29 pp.1049-1056

[36] J. von Neumann and R.D. Richtmyer (1950) A Method for the Numerical Calculations of Hydrodynamical Shocks, J. Appl. Phys, vol.21, p.232

[37] P.R. Woodward and P. Colella (1984) The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys., vol.54 pp.115-173

[38] H.C. Yee (1987) Construction of Explicit and Implicit Symmetric TVD Schemes and Their Applications, J. Comput. Phys., vol.68 pp.151-179

[39] H.C. Yee, A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods, NASA TM-101088, February 1989 (unpublished)

[40] C. Johnson (1992) Streamline Diffusion Finite-Element Methods for Compressible and Incompressible Fluid Flow, Finite Elements in Fluids, vol.8 pp.75-96