ESSENTIALLY NON-OSCILLATORY AND WEIGHTED ESSENTIALLY NON-OSCILLATORY SCHEMES FOR HYPERBOLIC CONSERVATION LAWS

Chi-Wang Shu[1]

Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912, U. S. A.
E-mail: shu@cfm.brown.edu
Phone: (401) 863-2549
FAX: (401) 863-1355

ABSTRACT

In these lecture notes we describe the construction, analysis, and application of ENO (Essentially Non-Oscillatory) and WENO (Weighted Essentially Non-Oscillatory) schemes for hyperbolic conservation laws and related Hamilton-Jacobi equations. ENO and WENO schemes are high order accurate finite difference schemes designed for problems with piecewise smooth solutions containing discontinuities. The key idea lies at the approximation level, where a nonlinear adaptive procedure is used to automatically choose the locally smoothest stencil, hence avoiding crossing discontinuities in the interpolation procedure as much as possible. ENO and WENO schemes have been quite successful in applications, especially for problems containing both shocks and complicated smooth solution structures, such as compressible turbulence simulations and aeroacoustics.

These lecture notes are basically self-contained. It is our hope that with these notes and with the help of the quoted references, the readers can understand the algorithms and code them up for applications. Sample codes are also available from the author.

[1]Research of the author was partially supported by NSF grants DMS-9500814, ECS-9214488, ECS-9627849 and INT-9601084, ARO grants DAAH04-94-G-0205 and DAAG55-97-1-0318, NASA Langley grant NAG-1-1145 and Contract NAS1-19480 while in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-0001, and AFOSR grant F49620-96-1-0150.

Contents

1 Introduction

ENO (Essentially Non-Oscillatory) schemes were started with the classic paper of Harten, Engquist, Osher and Chakravarthy in 1987 [38]. This paper has been cited at least 144 times by early 1997, according to the ISI database. The Journal of Computational Physics decided to republish this classic paper as part of the celebration of the journal's 30th birthday [68].

Finite difference and related finite volume schemes are based on interpolations of discrete data using polynomials or other simple functions. In the approximation theory, it is well known that the wider the stencil, the higher the order of accuracy of the interpolation, *provided the function being interpolated is smooth inside the stencil.* Traditional finite difference methods are based on fixed stencil interpolations. For example, to obtain an interpolation for cell i to third order accuracy, the information of the three cells $i-1$, i and $i+1$ can be used, to build a second order interpolation polynomial. In other words, one always looks one cell to the left, one cell to the right, plus the center cell itself, regardless of where in the domain one is situated. This works well for globally smooth problems. The resulting scheme is linear for linear PDEs, hence stability can be easily analyzed by Fourier transforms (for the uniform grid case). However, fixed stencil interpolation of second or higher order accuracy is necessarily *oscillatory* near a discontinuity, see Fig. 2.1, left, in Sect. 2.2. Such oscillations, which are called the Gibbs phenomena in spectral methods, do not decay in magnitude when the mesh is refined. It is a nuisance to say the least for practical calculations, and often leads to numerical instabilities in nonlinear problems containing discontinuities.

Before 1987, there were mainly two common ways to eliminate or reduce such spurious oscillations near discontinuities. One way was to add an artificial viscosity. This could be tuned so that it was large enough near the discontinuity to suppress, or at least to reduce the oscillations, but was small elsewhere to maintain high-order accuracy. One disadvantage of this approach is that fine tuning of the parameter controlling the size of the artificial viscosity is problem dependent. Another way was to apply limiters to eliminate the oscillations. In effect, one reduced the order of accuracy of the interpolation near the discontinuity (e.g. reducing the slope of a linear interpolant, or using a linear rather than a quadratic interpolant near the shock). By carefully designing such limiters, the TVD (total variation diminishing) property could be achieved for nonlinear scalar one dimensional problems. One disadvantage of this approach is that accuracy necessarily degenerates to first order near *smooth* extrema. We will not discuss the method of adding explicit artificial viscosity or the TVD method in these lecture notes. We refer to the books by Sod [75] and by LeVeque [52], and the references listed therein, for details.

The ENO idea proposed in [38] seems to be the first successful attempt to obtain a self similar (i.e. no mesh size dependent parameter), uniformly high order accurate, yet essentially non-oscillatory interpolation (i.e. the magnitude of the oscillations decays as $O(\Delta x^k)$ where k is the order of accuracy) for piecewise smooth functions. The generic

solution for hyperbolic conservation laws is in the class of piecewise smooth functions. The reconstruction in [38] is a natural extension of an earlier second order version of Harten and Osher [37]. In [38], Harten, Engquist, Osher and Chakravarthy investigated different ways of measuring local smoothness to determine the local stencil, and developed a hierarchy that begins with one or two cells, then adds one cell at a time to the stencil from the two candidates on the left and right, based on the size of the two relevant Newton divided differences. Although there are other reasonable strategies to choose the stencil based on local smoothness, such as comparing the magnitudes of the highest degree divided differences among all candidate stencils and picking the one with the least absolute value, experience seems to show that the hierarchy proposed in [38] is the most robust for a wide range of grid sizes, Δx, both *before* and inside the asymptotic regime.

As one can see from the numerical examples in [38] and in later papers, many of which being mentioned in these lecture notes or in the references listed, ENO schemes are indeed uniformly high order accurate and resolve shocks with sharp and monotone (to the eye) transitions. ENO schemes are especially suitable for problems containing both shocks and complicated smooth flow structures, such as those occurring in shock interactions with a turbulent flow and shock interaction with vortices.

Since the publication of the original paper of Harten, Engquist, Osher and Chakravarthy [38], the original authors and many other researchers have followed the pioneer work, improving the methodology and expanding the area of its applications. ENO schemes based on point values and TVD Runge-Kutta time discretizations, which can save computational costs significantly for multi space dimensions, were developed in [69] and [70]. Later biasing in the stencil choosing process to enhance stability and accuracy were developed in [28] and [67]. Weighted ENO (WENO) schemes were developed, using a convex combination of all candidate stencils instead of just one as in the original ENO, [53], [43]. ENO schemes based on other than polynomial building blocks were constructed in [40], [16]. Sub-cell resolution and artificial compression to sharpen contact discontinuities were studied in [35], [83], [70] and [43]. Multidimensional ENO schemes based on general triangulation were developed in [1]. ENO and WENO schemes for Hamilton-Jacobi type equations were designed and applied in [59], [60], [50] and [45]. ENO schemes using one-sided Jocobians for field by field decomposition, which improves the robustness for calculations of systems, were discussed in [25]. Combination of ENO with multiresolution ideas was pursued in [7]. Combination of ENO with spectral method using a domain decomposition approach was carried out in [8]. On the application side, ENO and WENO have been successfully used to simulate shock turbulence interactions [70], [71], [2]; to the direct simulation of compressible turbulence [71], [80], [49]; to relativistic hydrodynamics equations [24]; to shock vortex interactions and other gas dynamics problems [12], [27], [43]; to incompressible flow problems [26], [31]; to viscoelasticity equations with fading memory [72]; to semi-conductor device simulation [28], [41], [42]; to image processing [59], [64], [73]; etc. This list is definitely incomplete and may be biased by the author's own research experience, but one can already see that ENO and WENO have been applied quite extensively in many different fields. Most of the problems solved by ENO and WENO schemes are of the type in which solutions contain both strong shocks and rich smooth region structures. Lower order methods usually have difficulties for such problems and it is thus attractive and efficient to use high order stable methods such as ENO and WENO to handle them.

Today the study and application of ENO and WENO schemes are still very active. We expect the schemes and the basic methodology to be developed further and to become even more successful in the future.

In these lecture notes we describe the construction, analysis, and application of ENO and WENO schemes for hyperbolic conservation laws and related Hamilton-Jacobi equations. They are basically self-contained. Our hope is that with these notes and with the help of the quoted references, the readers can understand the algorithms and code them up for applications. Sample codes are also available from the author.

2 One Space Dimension

2.1 Reconstruction and Approximation in 1D

In this section we concentrate on the problems of interpolation and approximation in one space dimension.

Given a grid

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < ... < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b, \tag{2.1}$$

We define cells, cell centers, and cell sizes by

$$\begin{aligned} I_i &\equiv \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right], \quad x_i \equiv \tfrac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right), \\ \Delta x_i &\equiv x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad\qquad i = 1, 2, ..., N\,. \end{aligned} \tag{2.2}$$

We denote the maximum cell size by

$$\Delta x \equiv \max_{1 \le i \le N} \Delta x_i\,. \tag{2.3}$$

2.1.1 Reconstruction from cell averages.

The first approximation problem we will face, in solving hyperbolic conservation laws using cell averages (finite volume schemes, see Sect. 2.3.1), is the following *reconstruction* problem [38].

Problem 2.1. One dimensional reconstruction.

Given the cell averages of a function $v(x)$:

$$\overline{v}_i \equiv \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(\xi)\, d\xi, \qquad i = 1, 2, ..., N\,, \tag{2.4}$$

find a polynomial $p_i(x)$, of degree at most $k-1$, for each cell I_i, such that it is a k-th order accurate approximation to the function $v(x)$ inside I_i:

$$p_i(x) = v(x) + O(\Delta x^k), \qquad x \in I_i, \quad i = 1, ..., N\,. \tag{2.5}$$

In particular, this gives approximations to the function $v(x)$ at the cell boundaries

$$v^-_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}), \qquad v^+_{i-\frac{1}{2}} = p_i(x_{i-\frac{1}{2}}), \qquad i = 1, ..., N\,, \tag{2.6}$$

which are k-th order accurate:

$$v^-_{i+\frac{1}{2}} = v(x_{i+\frac{1}{2}}) + O(\Delta x^k), \qquad v^+_{i-\frac{1}{2}} = v(x_{i-\frac{1}{2}}) + O(\Delta x^k), \qquad i = 1, ..., N\,. \tag{2.7}$$

□

The polynomial $p_i(x)$ in Problem 2.1 can be replaced by other simple functions, such as trigonometric polynomials. See Sect. 4.1.3.

We will not discuss boundary conditions in this section. We thus assume that $\overline{v}_i$ is also available for $i \leq 0$ and $i > N$ if needed.

In the following we describe a procedure to solve Problem 2.1.

Given the location I_i and the order of accuracy k, we first choose a "stencil", based on r cells to the left, s cells to the right, and I_i itself if $r, s \geq 0$, with $r + s + 1 = k$:

$$S(i) \equiv \{I_{i-r}, ..., I_{i+s}\}\,. \tag{2.8}$$

There is a unique polynomial of degree at most $k - 1 = r + s$, denoted by $p(x)$ (we will drop the subscript i when it does not cause confusion), whose cell average in each of the cells in $S(i)$ agrees with that of $v(x)$:

$$\frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p(\xi)\, d\xi \;=\; \overline{v}_j, \qquad j = i - r, ..., i + s. \tag{2.9}$$

This polynomial $p(x)$ is the k-th order approximation we are looking for, as it is easy to prove (2.5), see the discussion below, as long as the function $v(x)$ is smooth in the region covered by the stencil $S(i)$.

For solving Problem 2.1, we also need the approximations to the values of $v(x)$ at the cell boundaries, (2.6). Since the mappings from the given cell averages $\overline{v}_j$ in the stencil $S(i)$ to the values $v^-_{i+\frac{1}{2}}$ and $v^+_{i-\frac{1}{2}}$ in (2.6) are linear, there exist constants c_{rj} and $\tilde{c}_{rj}$, which depend on the left shift of the stencil r of the stencil $S(i)$ in (2.8), on the order of accuracy k, and on the cell sizes Δx_j in the stencil S_i, but *not* on the function v itself, such that

$$v^-_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj} \overline{v}_{i-r+j}\,, \qquad v^+_{i-\frac{1}{2}} = \sum_{j=0}^{k-1} \tilde{c}_{rj} \overline{v}_{i-r+j}\,. \tag{2.10}$$

We note that the difference between the values with superscripts $\pm$ at the same location $x_{i+\frac{1}{2}}$ is due to the possibility of different stencils for cell I_i and for cell I_{i+1}. If we identify the left shift r not with the cell I_i but with the point of reconstruction $x_{i+\frac{1}{2}}$, i.e. using the stencil (2.8) to approximate $x_{i+\frac{1}{2}}$, then we can drop the superscripts $\pm$ and also eliminate the need to consider $\tilde{c}_{rj}$ in (2.10), as it is clear that

$$\tilde{c}_{rj} = c_{r-1,j}.$$

We summarize this as follows: given the k cell averages

$$\overline{v}_{i-r},\ \ldots,\ \overline{v}_{i-r+k-1}\,,$$

there are constants c_{rj} such that the reconstructed value at the cell boundary $x_{i+\frac{1}{2}}$:

$$v_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj}\overline{v}_{i-r+j}, \tag{2.11}$$

is k-th order accurate:

$$v_{i+\frac{1}{2}} = v(x_{i+\frac{1}{2}}) + O(\Delta x^k). \tag{2.12}$$

To understand how the constants $\{c_{rj}\}$ are obtained, as well as how the accuracy property (2.5) is proven, we look at the primitive function of $v(x)$:

$$V(x) \equiv \int_{-\infty}^{x} v(\xi)\,d\xi\,, \tag{2.13}$$

where the lower limit $-\infty$ is not important and can be replaced by any fixed number. Clearly, $V(x_{i+\frac{1}{2}})$ can be expressed by the cell averages of $v(x)$ using (2.4):

$$V(x_{i+\frac{1}{2}}) \;=\; \sum_{j=-\infty}^{i} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(\xi)\,d\xi \;=\; \sum_{j=-\infty}^{i} \overline{v}_j \Delta x_j, \tag{2.14}$$

thus with the knowledge of the cell averages $\{\overline{v}_j\}$ we also know the primitive function $V(x)$ at the cell boundaries exactly. If we denote the unique polynomial of degree at most k, which interpolates $V(x_{j+\frac{1}{2}})$ at the following $k+1$ points:

$$x_{i-r-\frac{1}{2}},\ \ldots,\ x_{i+s+\frac{1}{2}}, \tag{2.15}$$

by $P(x)$, and denote its derivative by $p(x)$:

$$p(x) \equiv P'(x)\,, \tag{2.16}$$

then it is easy to verify (2.9):

$$\begin{aligned}
\frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p(\xi)\,d\xi \;&=\; \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} P'(\xi)\,d\xi \;=\; \frac{1}{\Delta x_j}\Big(P(x_{j+\frac{1}{2}}) - P(x_{j-\frac{1}{2}})\Big)\\
&=\; \frac{1}{\Delta x_j}\Big(V(x_{j+\frac{1}{2}}) - V(x_{j-\frac{1}{2}})\Big)\\
&=\; \frac{1}{\Delta x_j}\left(\int_{-\infty}^{x_{j+\frac{1}{2}}} v(\xi)\,d\xi \;-\; \int_{-\infty}^{x_{j-\frac{1}{2}}} v(\xi)\,d\xi\right)\\
&=\; \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(\xi)\,d\xi \;=\; \overline{v}_j, \qquad j = i-r, ..., i+s,
\end{aligned}$$

where the third equality holds because $P(x)$ interpolates $V(x)$ at the points $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$ whenever $j = i-r, ..., i+s$. This implies that $p(x)$ is the polynomial we are looking for. Standard approximation theory (see an elementary numerical analysis book) tells us that

$$P'(x) = V'(x) + O(\Delta x^k), \qquad x \in I_i.$$

This is the accuracy requirement (2.5).

Now let us look at the practical issue of how to obtain the constants $\{c_{rj}\}$ in (2.11). For this we could use the Lagrange form of the interpolation polynomial:

$$P(x) = \sum_{m=0}^{k} V(x_{i-r+m-\frac{1}{2}}) \prod_{\substack{l=0 \\ l \neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}} . \tag{2.17}$$

For easier manipulation we subtract a constant $V(x_{i-r-\frac{1}{2}})$ from (2.17), and use the fact that

$$\sum_{m=0}^{k} \prod_{\substack{l=0 \\ l \neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}} = 1 ,$$

to obtain:

$$\begin{aligned} & P(x) - V(x_{i-r-\frac{1}{2}}) \\ = & \sum_{m=0}^{k} \left(V(x_{i-r+m-\frac{1}{2}}) - V(x_{i-r-\frac{1}{2}}) \right) \prod_{\substack{l=0 \\ l \neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}} . \end{aligned} \tag{2.18}$$

Taking derivative on both sides of (2.18), and noticing that

$$V(x_{i-r+m-\frac{1}{2}}) - V(x_{i-r-\frac{1}{2}}) = \sum_{j=0}^{m-1} \overline{v}_{i-r+j} \Delta x_{i-r+j}$$

because of (2.14), we obtain

$$p(x) = \sum_{m=0}^{k} \sum_{j=0}^{m-1} \overline{v}_{i-r+j} \Delta x_{i-r+j} \left(\frac{\sum^{k}_{\substack{l=0 \\ l \neq m}} \prod^{k}_{\substack{q=0 \\ q \neq m,l}} \left(x - x_{i-r+q-\frac{1}{2}} \right)}{\prod^{k}_{\substack{l=0 \\ l \neq m}} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) . \tag{2.19}$$

Evaluating the expression (2.19) at $x = x_{i+\frac{1}{2}}$, we finally obtain

$$\begin{aligned} & v_{i+\frac{1}{2}} = p(x_{i+\frac{1}{2}}) \\ = & \sum_{j=0}^{k-1} \left(\sum_{m=j+1}^{k} \frac{\sum^{k}_{\substack{l=0 \\ l \neq m}} \prod^{k}_{\substack{q=0 \\ q \neq m,l}} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod^{k}_{\substack{l=0 \\ l \neq m}} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \, \overline{v}_{i-r+j} , \end{aligned}$$

i.e. the constants c_{rj} in (2.11) are given by

$$c_{rj} = \left(\sum_{m=j+1}^{k} \frac{\sum^{k}_{\substack{l=0 \\ l\neq m}} \prod^{k}_{\substack{q=0 \\ q\neq m,l}} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod^{k}_{\substack{l=0 \\ l\neq m}} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \,. \tag{2.20}$$

Although there are many zero terms in the inner sum of (2.20) when $x_{i+\frac{1}{2}}$ is a node in the interpolation, we will keep this general form so that it applies also to the case where $x_{i+\frac{1}{2}}$ is not an interpolation point.

For a nonuniform grid, one would want to pre-compute the constants $\{c_{rj}\}$ as in (2.20), for $0 \le i \le N$, $-1 \le r \le k-1$, and $0 \le j \le k-1$, and store them before solving the PDE.

For a uniform grid, $\Delta x_i = \Delta x$, the expression for c_{rj} does not depend on i or Δx any more:

$$c_{rj} = \sum_{m=j+1}^{k} \frac{\sum^{k}_{\substack{l=0 \\ l\neq m}} \prod^{k}_{\substack{q=0 \\ q\neq m,l}} (r-q+1)}{\prod^{k}_{\substack{l=0 \\ l\neq m}} (m-l)} \,. \tag{2.21}$$

We list in Table 2.1 the constants c_{rj} in this uniform grid case (2.21), for order of accuracy between $k=1$ and $k=7$.

From Table 2.1, we would know, for example, that

$$v_{i+\frac{1}{2}} = -\frac{1}{6}\overline{v}_{i-1} + \frac{5}{6}\overline{v}_i + \frac{1}{3}\overline{v}_{i+1} + O(\Delta x^3) \,.$$

2.1.2 Conservative approximation to the derivative from point values.

The second approximation problem we will face, in solving hyperbolic conservation laws using point values (finite difference schemes, see Sect. 2.3.2), is the following problem in obtaining high order *conservative* approximation to the derivative from point values [69, 70].

Problem 2.2. One dimensional conservative approximation.

Given the point values of a function $v(x)$:

$$v_i \equiv v(x_i), \qquad i = 1, 2, ..., N \,, \tag{2.22}$$

find a numerical flux function

$$\hat{v}_{i+\frac{1}{2}} \equiv \hat{v}(v_{i-r}, ..., v_{i+s}), \qquad i = 0, 1, ..., N \,, \tag{2.23}$$

such that the flux difference approximates the derivative $v'(x)$ to k-th order accuracy:

$$\frac{1}{\Delta x_i} \left(\hat{v}_{i+\frac{1}{2}} - \hat{v}_{i-\frac{1}{2}} \right) = v'(x_i) + O(\Delta x^k), \qquad i = 0, 1, ..., N. \tag{2.24}$$

Table 2.1: The constants c_{rj} in (2.21).

k	r	j=0	j=1	j=2	j=3	j=4	j=5	j=6
1	-1	1						
	0	1						
2	-1	3/2	-1/2					
	0	1/2	1/2					
	1	-1/2	3/2					
3	-1	11/6	-7/6	1/3				
	0	1/3	5/6	-1/6				
	1	-1/6	5/6	1/3				
	2	1/3	-7/6	11/6				
4	-1	25/12	-23/12	13/12	-1/4			
	0	1/4	13/12	-5/12	1/12			
	1	-1/12	7/12	7/12	-1/12			
	2	1/12	-5/12	13/12	1/4			
	3	-1/4	13/12	-23/12	25/12			
5	-1	137/60	-163/60	137/60	-21/20	1/5		
	0	1/5	77/60	-43/60	17/60	-1/20		
	1	-1/20	9/20	47/60	-13/60	1/30		
	2	1/30	-13/60	47/60	9/20	-1/20		
	3	-1/20	17/60	-43/60	77/60	1/5		
	4	1/5	-21/20	137/60	-163/60	137/60		
6	-1	49/20	-71/20	79/20	-163/60	31/30	-1/6	
	0	1/6	29/20	-21/20	37/60	-13/60	1/30	
	1	-1/30	11/30	19/20	-23/60	7/60	-1/60	
	2	1/60	-2/15	37/60	37/60	-2/15	1/60	
	3	-1/60	7/60	-23/60	19/20	11/30	-1/30	
	4	1/30	-13/60	37/60	-21/20	29/20	1/6	
	5	-1/6	31/30	-163/60	79/20	-71/20	49/20	
7	-1	363/140	-617/140	853/140	-2341/420	667/210	-43/42	1/7
	0	1/7	223/140	-197/140	153/140	-241/420	37/210	-1/42
	1	-1/42	13/42	153/140	-241/420	109/420	-31/420	1/105
	2	1/105	-19/210	107/210	319/420	-101/420	5/84	-1/140
	3	-1/140	5/84	-101/420	319/420	107/210	-19/210	1/105
	4	1/105	-31/420	109/420	-241/420	153/140	13/42	-1/42
	5	-1/42	37/210	-241/420	153/140	-197/140	223/140	1/7
	6	1/7	-43/42	667/210	-2341/420	853/140	-617/140	363/140

□

We again ignore the boundary conditions here and assume that v_i is available for $i \leq 0$ and $i > N$ if needed.

The solution of this problem is essential for the high order conservative schemes based on point values (finite difference) rather than on cell averages (finite volume).

This problem looks quite different from Problem 2.1. However, we will see that there is a close relationship between these two. *We assume that the grid is uniform*, $\Delta x_i = \Delta x$. This assumption is, unfortunately, essential in the following development.

If we can find a function $h(x)$, which may depend on the grid size Δx, such that

$$v(x) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi) d\xi \,, \tag{2.25}$$

then clearly

$$v'(x) = \frac{1}{\Delta x} \left[h\left(x + \frac{\Delta x}{2}\right) - h\left(x - \frac{\Delta x}{2}\right) \right] ,$$

hence all we need to do is to use

$$\hat{v}_{i+\frac{1}{2}} = h(x_{i+\frac{1}{2}}) + O(\Delta x^k) \tag{2.26}$$

to achieve (2.24). We note here that it would look like an $O(\Delta x^{k+1})$ term in (2.26) is needed in order to get (2.24), due to the Δx term in the denominator. However, in practice, the $O(\Delta x^k)$ term in (2.26) is usually smooth, hence the difference in (2.24) would give an extra $O(\Delta x)$, just to cancel the one in the denominator.

It is not easy to approximate $h(x)$ via (2.25), as it is only implicitly defined there. However, we notice that the known function $v(x)$ is the cell average of the unknown function $h(x)$, so to find $h(x)$ we just need to use the *reconstruction* procedure described in Sect. 2.1.1. If we take the primitive of $h(x)$:

$$H(x) = \int_{-\infty}^{x} h(\xi) d\xi \,, \tag{2.27}$$

then (2.25) clearly implies

$$H(x_{i+\frac{1}{2}}) = \sum_{j=-\infty}^{i} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} h(\xi) d\xi = \Delta x \sum_{j=-\infty}^{i} v_j \,. \tag{2.28}$$

Thus, given the point values $\{v_j\}$, we "identify" them as cell averages of another function $h(x)$ in (2.25), then the primitive function $H(x)$ is exactly known at the cell interfaces $x = x_{i+\frac{1}{2}}$. We thus use the same reconstruction procedure described in Sect. 2.1.1, to get a k-th order approximation to $h(x_{i+\frac{1}{2}})$, which is then taken as the numerical flux $\hat{v}_{i+\frac{1}{2}}$ in (2.23).

In other words, if the "stencil" for the flux $\hat{v}_{i+\frac{1}{2}}$ in (2.23) is the following k points:

$$x_{i-r}, \ \ldots, \ x_{i+s} \,, \tag{2.29}$$

where $r+s=k-1$, then the flux $\hat{v}_{i+\frac{1}{2}}$ is expressed as

$$\hat{v}_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj} v_{i-r+j}, \tag{2.30}$$

where the constants $\{c_{rj}\}$ are given by (2.21) and Table 2.1.

From Table 2.1 we would know, for example, that if

$$\hat{v}_{i+\frac{1}{2}} = -\frac{1}{6}v_{i-1} + \frac{5}{6}v_i + \frac{1}{3}v_{i+1}\,,$$

then

$$\frac{1}{\Delta x}\left(\hat{v}_{i+\frac{1}{2}} - \hat{v}_{i-\frac{1}{2}}\right) = v'(x_i) + O(\Delta x^3).$$

We emphasize again that, unlike in the reconstruction procedure in Sect. 2.1.1, here the grid *must* be uniform: $\Delta x_j = \Delta x$. Otherwise, it can be proven that no choice of constants c_{rj} in (2.30) (which may depend on the local grid sizes but not on the function $v(x)$) could make the conservative approximation to the derivative (2.24) higher than second order accurate ($k > 2$). The proof is a simple exercise of Taylor expansions. Thus, the high order finite difference (third order and higher) discussed in these lecture notes can apply only to uniform or smoothly varying grids.

Because of this equivalence of obtaining a conservative approximation to the derivative (2.23)-(2.24) and the reconstruction problem discussed in Sect. 2.1.1, we will only need to consider the reconstruction problem in the following sections.

2.1.3 Fixed stencil approximation.

By fixed stencil, we mean that the left shift r in (2.8) or (2.29) is *the same for all locations* i. Usually, for a globally smooth function $v(x)$, the best approximation is obtained either by a central approximation $r=s-1$ for even k (here central is relative to the location $x_{i+\frac{1}{2}}$), or by a one point upwind biased approximation $r=s$ or $r=s-2$ for odd k. For example, if the grid is uniform $\Delta x_i = \Delta x$, then a central 4th order reconstruction for $v_{i+\frac{1}{2}}$, in (2.11), is given by

$$v_{i+\frac{1}{2}} = -\frac{1}{12}\overline{v}_{i-1} + \frac{7}{12}\overline{v}_i + \frac{7}{12}\overline{v}_{i+1} - \frac{1}{12}\overline{v}_{i+2} + O(\Delta x^4)\,,$$

and the two one point upwind biased 3rd order reconstructions for $v_{i+\frac{1}{2}}$ in (2.11), are given by

$$\begin{aligned} & v_{i+\frac{1}{2}} &=& -\frac{1}{6}\overline{v}_{i-1} + \frac{5}{6}\overline{v}_i + \frac{1}{3}\overline{v}_{i+1} + O(\Delta x^3) \\ \textit{or} \qquad & v_{i+\frac{1}{2}} &=& \frac{1}{3}\overline{v}_i + \frac{5}{6}\overline{v}_{i+1} - \frac{1}{6}\overline{v}_{i+2} + O(\Delta x^3)\,. \end{aligned}$$

Similarly, a central 4th order flux (2.30) is

$$\hat{v}_{i+\frac{1}{2}} = -\frac{1}{12}v_{i-1} + \frac{7}{12}v_i + \frac{7}{12}v_{i+1} - \frac{1}{12}v_{i+2}\,,$$

which gives

$$\frac{1}{\Delta x}\left(\hat{v}_{i+\frac{1}{2}} - \hat{v}_{i-\frac{1}{2}}\right) = v'(x_i) + O(\Delta x^4),$$

and the two one point upwind biased 3rd order fluxes (2.30) are given by

$$\begin{aligned} & \hat{v}_{i+\frac{1}{2}} &=& -\frac{1}{6}v_{i-1} + \frac{5}{6}v_i + \frac{1}{3}v_{i+1} \\ \textit{or} \qquad & \hat{v}_{i+\frac{1}{2}} &=& \frac{1}{3}v_i + \frac{5}{6}v_{i+1} - \frac{1}{6}v_{i+2}\,, \end{aligned}$$

which gives

$$\frac{1}{\Delta x}\left(\hat{v}_{i+\frac{1}{2}} - \hat{v}_{i-\frac{1}{2}}\right) = v'(x_i) + O(\Delta x^3).$$

Traditional central and upwind schemes, either finite volume or finite difference, can be derived by these fixed stencil reconstructions or flux differenced approximations to the derivatives.

2.2 ENO and WENO Approximations in 1D

For solving hyperbolic conservation laws, we are interested in the class of piecewise smooth functions. A piecewise smooth function $v(x)$ is smooth (i.e. it has as many derivatives as the scheme calls for) except for at finitely many isolated points. At these points, $v(x)$ and its derivatives are assumed to have finite left and right limits. Such functions are "generic" for solutions to hyperbolic conservation laws.

For such piecewise smooth functions, the order of accuracy we refer to in these lecture notes are *formal*, that is, it is defined as whatever accuracy determined by the local truncation error in the *smooth regions* of the function.

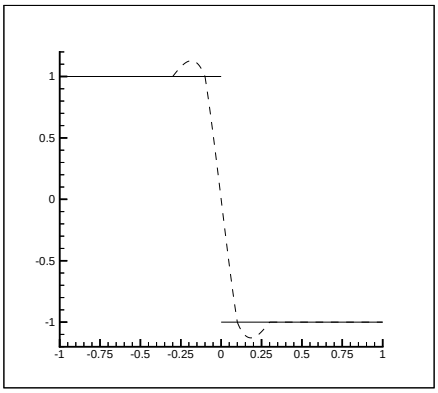
If the function $v(x)$ is only piecewise smooth, a fixed stencil approximation described in Sect. 2.1.3 may not be adequate near discontinuities. Fig. 2.1 (left) gives the 4-th order (piecewise cubic) interpolation with a central stencil for the step function, i.e. the polynomial approximation inside the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ interpolates the step function at the four points $x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}$. Notice the obvious over/undershoots for the cells near the discontinuity.

These oscillations (termed *the Gibbs Phenomena* in spectral methods) happen because the stencils, as defined by (2.15), actually contain the discontinuous cell for x_i close enough to the discontinuity. As a result, the approximation property (2.5) is no longer valid in such stencils.

2.2.1 ENO approximation.

A closer look at Fig. 2.1 (left) motivates the idea of "adaptive stencil", namely, the left shift r changes with the location x_i. The basic idea is to avoid including the discontinuous cell in the stencil, if possible.

To achieve this effect, we need to look at the Newton formulation of the interpolation polynomial.



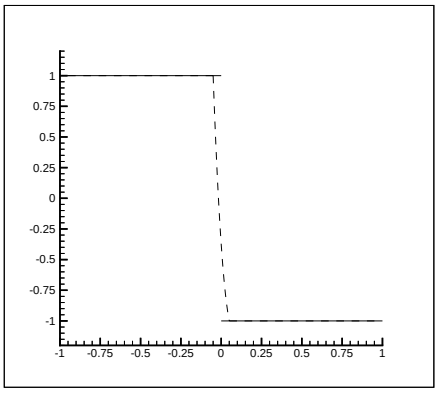


Figure 2.1: Fixed central stencil cubic interpolation (left) and ENO cubic interpolation (right) for the step function. Solid: exact function; Dashed: interpolant piecewise cubic polynomials.

We first review the definition of the Newton divided differences. The 0-th degree divided differences of the function $V(x)$ in (2.13)-(2.14) are defined by:

$$V[x_{i-\frac{1}{2}}] \equiv V(x_{i-\frac{1}{2}}); \tag{2.31}$$

and in general the j-th degree divided differences, for $j \geq 1$, are defined inductively by

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] \equiv \frac{V[x_{i+\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] - V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{3}{2}}]}{x_{i+j-\frac{1}{2}} - x_{i-\frac{1}{2}}}. \tag{2.32}$$

Similarly, the divided differences of the cell averages $\overline{v}$ in (2.4) are defined by

$$\overline{v}[x_i] \equiv \overline{v}_i; \tag{2.33}$$

and in general

$$\overline{v}[x_i, ..., x_{i+j}] \equiv \frac{\overline{v}[x_{i+1}, ..., x_{i+j}] - \overline{v}[x_i, ..., x_{i+j-1}]}{x_{i+j} - x_i}. \tag{2.34}$$

We note that, by (2.14),

$$V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] = \frac{V(x_{i+\frac{1}{2}}) - V(x_{i-\frac{1}{2}})}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} = \overline{v}_i\,, \tag{2.35}$$

i.e. the 0-th degree divided differences of $\overline{v}$ are the first degree divided differences of $V(x)$. We can thus write the divided differences of $V(x)$ of first degree and higher by those of $\overline{v}$ of 0-th degree and higher, using (2.35) and (2.32).

The Newton form of the k-th degree interpolation polynomial $P(x)$, which interpolates $V(x)$ at the $k+1$ points (2.15), can be expressed using the divided differences (2.31)-(2.32) by

$$P(x) \;=\; \sum_{j=0}^{k} V[x_{i-r-\frac{1}{2}}, ..., x_{i-r+j-\frac{1}{2}}] \prod_{m=0}^{j-1} \left(x - x_{i-r+m-\frac{1}{2}}\right)\,. \tag{2.36}$$

We can take the derivative of (2.36) to get $p(x)$ in (2.16):

$$p(x) = \sum_{j=1}^{k} V[x_{i-r-\frac{1}{2}}, ..., x_{i-r+j-\frac{1}{2}}] \sum_{m=0}^{j-1} \prod_{\substack{l=0 \\ l \neq m}}^{j-1} \left(x - x_{i-r+l-\frac{1}{2}}\right) . \tag{2.37}$$

Notice that only first and higher degree divided differences of $V(x)$ appear in (2.37). Hence by (2.35), we can express $p(x)$ completely by the divided differences of $\overline{v}$, without any need to reference $V(x)$.

Let us now recall an important property of divided differences:

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] = \frac{V^{(j)}(\xi)}{j!}, \tag{2.38}$$

for some ξ inside the stencil: $x_{i-\frac{1}{2}} < \xi < x_{i+j-\frac{1}{2}}$, *as long as the function* $V(x)$ *is smooth in this stencil.* If $V(x)$ is discontinuous at some point inside the stencil, then it is easy to verify that

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] = O\left(\frac{1}{\Delta x^j}\right) . \tag{2.39}$$

Thus the divided difference is a measurement of the smoothness of the function inside the stencil.

We now describe the ENO idea by using (2.36). Suppose our job is to find a stencil of $k+1$ consecutive points, which must include $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, such that $V(x)$ is "the smoothest" in this stencil comparing with other possible stencils. We perform this job by breaking it into steps, in each step we only add one point to the stencil. We thus start with the two point stencil

$$\tilde{S}_2(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}, \tag{2.40}$$

where we have used $\tilde{S}$ to denote a stencil for the primitive function V. Notice that the stencil $\tilde{S}$ for V has a corresponding stencil S for $\overline{v}$ through (2.35), for example (2.40) corresponds to a single cell stencil

$$S(i) = \{I_i\}$$

for $\overline{v}$. The linear interpolation on the stencil $\tilde{S}_2(i)$ in (2.40) can be written in the Newton form as

$$P^1(x) = V[x_{i-\frac{1}{2}}] + V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]\left(x - x_{i-\frac{1}{2}}\right).$$

At the next step, we have only two choices to expand the stencil by adding one point: we can either add the left neighbor $x_{i-\frac{3}{2}}$, resulting in the following quadratic interpolation

$$R(x) = P^1(x) + V[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]\left(x - x_{i-\frac{1}{2}}\right)\left(x - x_{i+\frac{1}{2}}\right), \tag{2.41}$$

or add the right neighbor $x_{i+\frac{3}{2}}$, resulting in the following quadratic interpolation

$$S(x) = P^1(x) + V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}]\left(x - x_{i-\frac{1}{2}}\right)\left(x - x_{i+\frac{1}{2}}\right). \tag{2.42}$$

We note that the deviations from $P^1(x)$ in (2.41) and (2.42), are the *same* function

$$\left(x - x_{i-\frac{1}{2}}\right)\left(x - x_{i+\frac{1}{2}}\right)$$

multiplied by two different constants

$$V[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \qquad and \qquad V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}]. \tag{2.43}$$

These two constants are the two second degree divided differences of $V(x)$ in two different stencils. We have already noticed before, in (2.38) and (2.39), that a smaller divided difference implies the function is "smoother" in that stencil. We thus decide upon which point to add to the stencil, by comparing the two relevant divided differences (2.43), and picking the one with a smaller absolute value. Thus, if

$$\left|V[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]\right| < \left|V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}]\right|, \tag{2.44}$$

we will take the 3 point stencil as

$$\tilde{S}_3(i) = \{x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\};$$

otherwise, we will take

$$\tilde{S}_3(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}\}.$$

This procedure can be continued, with one point added to the stencil at each step, according to the smaller of the absolute values of the two relevant divided differences, until the desired number of points in the stencil is reached.

We note that, for the uniform grid case $\Delta x_i = \Delta x$, there is no need to compute the divided differences as in (2.32). We should use undivided differences instead:

$$V < x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} >= V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] = \overline{v}_i \tag{2.45}$$

(see (2.35)), and

$$\begin{aligned} V < x_{i-\frac{1}{2}}, ..., x_{i+j+\frac{1}{2}} > \quad \equiv \quad & V < x_{i+\frac{1}{2}}, ..., x_{i+j+\frac{1}{2}} > -V < x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}} >, \\ & j \geq 1\,. \end{aligned} \tag{2.46}$$

The Newton interpolation formulae (2.36)-(2.37) should also be adjusted accordingly. This both saves computational time and reduces round-off effects.

The FORTRAN program for this ENO choosing process is very simple:

```
* assuming the m-th degree divided (or undivided) differences
* of V(x), with x_i as the left-most point in the arguments,
* are stored in V(i,m), also assuming that "is" is the
* left-most point in the stencil for cell i for a k-th
* degree polynomial

      is=i
      do m=2,k
      if(abs(V(is-1,m)).lt.abs(V(is,m))) is=is-1
      enddo
```

Once the stencil $\tilde{S}(i)$, hence $S(i)$, in (2.8) is found, one could use (2.11), with the prestored values of the constants c_{rj}, (2.20) or (2.21), to compute the reconstructed values at the cell boundary. Or, one could use (2.30) to compute the fluxes. An alternative way is to compute the values or fluxes using the Newton form (2.37) directly. The computational cost is about the same.

We summarize the ENO reconstruction procedure in the following

Procedure 2.1. 1D ENO reconstruction.

Given the cell averages $\{\overline{v}_i\}$ of a function $v(x)$, we obtain a piecewise polynomial reconstruction, of degree at most $k-1$, using ENO, in the following way:

1. Compute the divided differences of the primitive function $V(x)$, for degrees 1 to k, using $\overline{v}$, (2.35) and (2.32).

   If the grid is uniform $\Delta x_i = \Delta x$, at this stage, undivided differences (2.45)-(2.46) should be computed instead.

2. In cell I_i, start with a two point stencil

$$\tilde{S}_2(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$$

   for $V(x)$, which is equivalent to a one point stencil,

$$S_1(i) = \{I_i\}$$

   for $\overline{v}$.

3. For $l = 2, ..., k$, assuming

$$\tilde{S}_l(i) = \{x_{j+\frac{1}{2}}, ..., x_{j+l-\frac{1}{2}}\}$$

   is known, add one of the two neighboring points, $x_{j-\frac{1}{2}}$ or $x_{j+l+\frac{1}{2}}$, to the stencil, following the ENO procedure:

   - If

$$\left|V[x_{j-\frac{1}{2}}, ..., x_{j+l-\frac{1}{2}}]\right| < \left|V[x_{j+\frac{1}{2}}, ..., x_{j+l+\frac{1}{2}}]\right|, \tag{2.47}$$

     add $x_{j-\frac{1}{2}}$ to the stencil $\tilde{S}_l(i)$ to obtain

$$\tilde{S}_{l+1}(i) = \{x_{j-\frac{1}{2}}, ..., x_{j+l-\frac{1}{2}}\};$$

   - Otherwise, add $x_{j+l+\frac{1}{2}}$ to the stencil $\tilde{S}_l(i)$ to obtain

$$\tilde{S}_{l+1}(i) = \{x_{j+\frac{1}{2}}, ..., x_{j+l+\frac{1}{2}}\}.$$

4. Use the Lagrange form (2.19) or the Newton form (2.37) to obtain $p_i(x)$, which is a polynomial of degree at most $k-1$ in I_i, satisfying the accuracy condition (2.5), *as long as $v(x)$ is smooth in I_i.*

We could use $p_i(x)$ to get the approximations at the cell boundaries:

$$v^-_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}), \qquad v^+_{i-\frac{1}{2}} = p_i(x_{i-\frac{1}{2}}).$$

However, it is usually more convenient, when the stencil is known, to use (2.10), with c_{rj} defined by (2.20) for a nonuniform grid, or by (2.21) and Table 2.1 for a uniform grid, to compute an approximation to $v(x)$ at the cell boundaries.

□

For the same piecewise cubic interpolation to the step function, but this time using the ENO procedure with a two point stencil $\tilde{S}_2(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ in the Step 2 of Procedure 2.1, we obtain a non-oscillatory interpolation, in Fig. 2.1 (right).

For a piecewise smooth function $V(x)$, ENO interpolation starting with a two point stencil $\tilde{S}_2(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ in the Step 2 of Procedure 2.1, as was shown in Fig. 2.1 (right), has the following properties [39]:

1. The accuracy condition

   $$P_i(x) = V(x) + O(\Delta x^{k+1}), \qquad x \in I_i$$

   is valid for any cell I_i which does not contain a discontinuity.

   This implies that the ENO interpolation procedure can recover the full high order accuracy right up to the discontinuity.

2. $P_i(x)$ is monotone in any cell I_i which *does* contain a discontinuity of $V(x)$.

3. The reconstruction is TVB (total variation bounded). That is, there exists a function $z(x)$, satisfying

   $$z(x) = P_i(x) + O(\Delta x^{k+1}), \qquad x \in I_i$$

   for any cell I_i, including those cells which contain discontinuities, such that

   $$TV(z) \leq TV(V).$$

Property 3 is clearly a consequence of Properties 1 and 2 (just take $z(x)$ to be $V(x)$ in the smooth cells and take $z(x)$ to be $P_i(x)$ in the cells containing discontinuities). It is quite interesting that Property 2 holds. One would have expected trouble in those "shocked cells", i.e. cells I_i which contain discontinuities, for ENO would not help for such cases as the stencil starts with two points already containing a discontinuity. We will give a proof of Property 2 for a simple but illustrative case, i.e. when $V(x)$ is a step function

$$V(x) = \begin{cases} 0, & x \leq 0; \\ 1, & x > 0. \end{cases}$$

and the k-th degree polynomial $P(x)$ interpolates $V(x)$ at $k+1$ points

$$x_{\frac{1}{2}} < x_{\frac{3}{2}} < ... < x_{k+\frac{1}{2}}$$

containing the discontinuity

$$x_{j_0-\frac{1}{2}} < 0 < x_{j_0+\frac{1}{2}}$$

for some j_0 between 1 and k. For any interval which does not contain the discontinuity 0:

$$[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \qquad j \neq j_0, \tag{2.48}$$

we have

$$P(x_{j-\frac{1}{2}}) = V(x_{j-\frac{1}{2}}) = V(x_{j+\frac{1}{2}}) = P(x_{j+\frac{1}{2}}),$$

hence there is at least one point ξ_j in between, $x_{j-\frac{1}{2}} < \xi_j < x_{j+\frac{1}{2}}$, such that $P'(\xi_j) = 0$. This way we can find $k-1$ distinct zeroes for $P'(x)$, as there are $k-1$ intervals (2.48) which do not contain the discontinuity 0. However, $P'(x)$ is a non-zero polynomial of degree at most $k-1$, hence can have at most $k-1$ distinct zeroes. This implies that $P'(x)$ *does not have any zero inside the shocked interval* $[x_{j_0-\frac{1}{2}}, x_{j_0+\frac{1}{2}}]$, i.e. $P(x)$ is *monotone* in this shocked interval. This proof can be generalized to a proof for Property 2 [39].

2.2.2 WENO approximation.

In this subsection we describe the recently developed WENO (weighted ENO) reconstruction procedure [53, 43]. WENO is based on ENO, of course. For simplicity of presentation, in this subsection we assume the grid is uniform, i.e. $\Delta x_i = \Delta x$.

As we can see from Sect. 2.2.1, ENO reconstruction is uniformly high order accurate right up to the discontinuity. It achieves this effect by adaptively choosing the stencil based on the absolute values of divided differences. However, one could make the following remarks about ENO reconstruction, indicating rooms for improvements:

1. The stencil might change even by a round-off error perturbation near zeroes of the solution and its derivatives. That is, when both sides of (2.47) are near 0, a small change at the round off level would change the direction of the inequality and hence the stencil. In smooth regions, this "free adaptation" of stencils is clearly not necessary. Moreover, this may cause loss of accuracy when applied to a hyperbolic PDE [63, 67].

2. The resulting numerical flux (2.23) is not smooth, as the stencil pattern may change at neighboring points.

3. In the stencil choosing process, k candidate stencils are considered, covering $2k-1$ cells, but only one of the stencils is actually used in forming the reconstruction (2.10) or the flux (2.30), resulting in k-th order accuracy. If all the $2k-1$ cells in the potential stencils are used, one could get $(2k-1)$-th order accuracy in smooth regions.

4. ENO stencil choosing procedure involves many logical "if" structures, or equivalent mathematical formulae, which are not very efficient on certain vector computers such as CRAYs (however they are friendly to parallel computers).

There have been attempts in the literature to remedy the first problem, the "free adaptation" of stencils. In [28] and [67], the following "biasing" strategy was proposed. One first identity a "preferred" stencil

$$\tilde{S}_{pref}(i) = \{x_{i-r+\frac{1}{2}}, ..., x_{i-r+k+\frac{1}{2}}\}, \tag{2.49}$$

which might be central or one-point upwind. One then replaces (2.47) by

$$\left|V[x_{j-\frac{1}{2}},...,x_{j+l-\frac{1}{2}}]\right| < b\,\left|V[x_{j+\frac{1}{2}},...,x_{j+l+\frac{1}{2}}]\right|\,,$$

if

$$x_{j+\frac{1}{2}} > x_{i-r+\frac{1}{2}}\,,$$

i.e. if the left-most point $x_{j+\frac{1}{2}}$ in the current stencil $\tilde{S}_l(i)$ has not reached the left-most point $x_{i-r+\frac{1}{2}}$ of the preferred stencil $S_{pref}(i)$ in (2.49) yet; otherwise, if

$$x_{j+\frac{1}{2}} \le x_{i-r+\frac{1}{2}}\,,$$

one replaces (2.47) by

$$b\,\left|V[x_{j-\frac{1}{2}},...,x_{j+l-\frac{1}{2}}]\right| < \left|V[x_{j+\frac{1}{2}},...,x_{j+l+\frac{1}{2}}]\right|\,.$$

Here, $b > 1$ is the so-called biasing parameter. Analysis in [67] indicates a good choice of the parameter $b = 2$. The philosophy is to stay as close as possible to the preferred stencil, unless the alternative candidate is, roughly speaking, a factor $b > 1$ better in smoothness.

WENO is a more recent attempt to improve upon ENO in these four points. The basic idea is the following: instead of using only one of the candidate stencils to form the reconstruction, one uses a convex combination of all of them. To be more precise, suppose the k candidate stencils

$$S_r(i) = \{x_{i-r},...,x_{i-r+k-1}\}, \qquad r = 0,...,k-1 \tag{2.50}$$

produce k different reconstructions to the value $v_{i+\frac{1}{2}}$, according to (2.11),

$$v^{(r)}_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj}\overline{v}_{i-r+j}, \qquad r = 0,...,k-1\,, \tag{2.51}$$

WENO reconstruction would take a convex combination of all $v^{(r)}_{i+\frac{1}{2}}$ defined in (2.51) as a new approximation to the cell boundary value $v(x_{i+\frac{1}{2}})$:

$$v_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r v^{(r)}_{i+\frac{1}{2}}. \tag{2.52}$$

Apparently, the key to the success of WENO would be the choice of the weights ω_r. We require

$$\omega_r \ge 0, \qquad \sum_{r=0}^{k-1} \omega_r \;=\; 1 \tag{2.53}$$

for stability and consistency.

If the function $v(x)$ is smooth in all of the candidate stencils (2.50), there are constants d_r such that

$$v_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} d_r v^{(r)}_{i+\frac{1}{2}} = v(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1})\,. \tag{2.54}$$

For example, d_r for $1 \le k \le 3$ are given by

$$\begin{aligned} d_0 &= 1, \qquad k=1; \\ d_0 &= \frac{2}{3}, \quad d_1 = \frac{1}{3}, \qquad k=2; \\ d_0 &= \frac{3}{10}, \quad d_1 = \frac{3}{5}, \quad d_2 = \frac{1}{10}, \qquad k=3. \end{aligned}$$

We can see that d_r is always positive and, due to consistency,

$$\sum_{r=0}^{k-1} d_r = 1\,. \tag{2.55}$$

In this smooth case, we would like to have

$$\omega_r = d_r + O(\Delta x^{k-1}), \qquad r = 0, ..., k-1\,, \tag{2.56}$$

which would imply $(2k-1)$-th order accuracy:

$$v_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r v_{i+\frac{1}{2}}^{(r)} = v(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1}) \tag{2.57}$$

because

$$\begin{aligned} \sum_{r=0}^{k-1} \omega_r v_{i+\frac{1}{2}}^{(r)} - \sum_{r=0}^{k-1} d_r v_{i+\frac{1}{2}}^{(r)} &= \sum_{r=0}^{k-1} (\omega_r - d_r) \left(v_{i+\frac{1}{2}}^{(r)} - v(x_{i+\frac{1}{2}}) \right) \\ &= \sum_{r=0}^{k-1} O(\Delta x^{k-1})\, O(\Delta x^k) \;=\; O(\Delta x^{2k-1}) \end{aligned}$$

where in the first equality we used (2.53) and (2.55).

When the function $v(x)$ has a discontinuity in one or more of the stencils (2.50), we would hope the corresponding weight(s) ω_r to be essentially 0, to emulate the successful ENO idea.

Another consideration is that the weights should be smooth functions of the cell averages involved. In fact, the weights designed in [43] and described below are C^∞.

Finally, we would like to have weights which are computationally efficient. Thus, polynomials or rational functions are preferred over exponential type functions.

All these considerations lead to the following form of weights:

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1} \alpha_s}, \qquad r = 0, ..., k-1 \tag{2.58}$$

with

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}\,. \tag{2.59}$$

Here $\epsilon > 0$ is introduced to avoid the denominator to become 0. We take $\epsilon = 10^{-6}$ in all our numerical tests [43]. β_r are the so-called "smooth indicators" of the stencil $S_r(i)$: if the function $v(x)$ is smooth in the stencil $S_r(i)$, then

$$\beta_r = O(\Delta x^2)\,,$$

but if $v(x)$ has a discontinuity inside the stencil $S_r(i)$, then

$$\beta_r = O(1).$$

Translating into the weights ω_r in (2.58), we will have

$$\omega_r = O(1)$$

when the function $v(x)$ is smooth in the stencil $S_r(i)$, and

$$\omega_r = O(\Delta x^4)$$

if $v(x)$ has a discontinuity inside the stencil $S_r(i)$. Emulation of ENO near a discontinuity is thus achieved.

One also has to worry about the accuracy requirement (2.56), which must be checked when the specific form of the smooth indicator β_r is given. For any smooth indicator β_r, it is easy to see that the weights defined by (2.58) satisfies (2.53). To satisfy (2.56), it suffices to have, through a Taylor expansion analysis:

$$\beta_r = D\,(1 + O(\Delta x^{k-1})), \qquad r = 0, ..., k-1, \tag{2.60}$$

where D is a nonzero quantity independent of r (but may depend on Δx).

As we have seen in Sect. 2.2.1, the ENO reconstruction procedure chooses the "smoothest" stencil by comparing a hierarchy of divided or undivided differences. This is because these differences can be used to measure the smoothness of the function on a stencil, (2.38)-(2.39). In [43], after extensive experiments, a robust (for third and fifth order at least) choice of smooth indicators β_r is given. As we know, on each stencil $S_r(i)$, we can construct a $(k-1)$-th degree reconstruction polynomial, which if evaluated at $x = x_{i+\frac{1}{2}}$, renders the approximation to the value $v(x_{i+\frac{1}{2}})$ in (2.51). Since the total variation is a good measurement for smoothness, it would be desirable to minimize the total variation for this reconstruction polynomial inside I_i. Consideration for a smooth flux and for the role of higher order variations leads us to the following measurement for smoothness: let the reconstruction polynomial on the stencil $S_r(i)$ be denoted by $p_r(x)$, we define

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial^l x} \right)^2 dx\,. \tag{2.61}$$

The right hand side of (2.61) is just a sum of the squares of scaled L^2 norms for all the derivatives of the interpolation polynomial $p_r(x)$ over the interval $(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$. The factor Δx^{2l-1} is introduced to remove any Δx dependency in the derivatives, in order to preserve self-similarity when used to hyperbolic PDEs (Sect. 2.3).

We remark that (2.61) is similar to but smoother than the total variation measurement based on the L^1 norm. It also renders a more accurate WENO scheme for the case $k = 2$ and 3.

When $k = 2$, (2.61) gives the following smoothness measurement [53, 43]:

$$\begin{aligned} \beta_0 &= (\overline{v}_{i+1} - \overline{v}_i)^2\,, \\ \beta_1 &= (\overline{v}_i - \overline{v}_{i-1})^2\,. \end{aligned} \tag{2.62}$$

For $k=3$, (2.61) gives [43]:

$$\begin{aligned}
\beta_0 &= \frac{13}{12}(\overline{v}_i - 2\overline{v}_{i+1} + \overline{v}_{i+2})^2 + \frac{1}{4}(3\overline{v}_i - 4\overline{v}_{i+1} + \overline{v}_{i+2})^2\,, \\
\beta_1 &= \frac{13}{12}(\overline{v}_{i-1} - 2\overline{v}_i + \overline{v}_{i+1})^2 + \frac{1}{4}(\overline{v}_{i-1} - \overline{v}_{i+1})^2\,, \\
\beta_2 &= \frac{13}{12}(\overline{v}_{i-2} - 2\overline{v}_{i-1} + \overline{v}_i)^2 + \frac{1}{4}(\overline{v}_{i-2} - 4\overline{v}_{i-1} + 3\overline{v}_i)^2\,.
\end{aligned} \tag{2.63}$$

We can easily verify that the accuracy condition (2.60) is satisfied, even near smooth extrema [43]. This indicates that (2.62) gives a third order WENO scheme, and (2.63) gives a fifth order one.

Notice that the discussion here has a one point upwind bias in the optimal linear stencil, suitable for a problem with wind blowing from left to right. If the wind blows the other way, the procedure should be modified symmetrically with respect to $x_{i+\frac{1}{2}}$.

In summary, we have the following WENO reconstruction procedure:

Procedure 2.2. 1D WENO reconstruction.

Given the cell averages $\{\overline{v}_i\}$ of a function $v(x)$, for each cell I_i, we obtain upwind biased $(2k-1)$-th order approximations to the function $v(x)$ at the cell boundaries, denoted by $v^+_{i-\frac{1}{2}}$ and $v^-_{i+\frac{1}{2}}$, in the following way:

1. Obtain the k reconstructed values $v^{(r)}_{i+\frac{1}{2}}$, of k-th order accuracy, in (2.51), based on the stencils (2.50), for $r=0,...,k-1$;

   Also obtain the k reconstructed values $v^{(r)}_{i-\frac{1}{2}}$, of k-th order accuracy, using (2.10), again based on the stencils (2.50), for $r=0,...,k-1$;

2. Find the constants d_r and $\tilde{d}_r$, such that (2.54) and

$$v_{i-\frac{1}{2}} = \sum_{r=0}^{k-1} \tilde{d}_r v^{(r)}_{i-\frac{1}{2}} = v(x_{i-\frac{1}{2}}) + O(\Delta x^{2k-1})$$

   are valid. By symmetry,

$$\tilde{d}_r = d_{k-1-r}.$$

3. Find the smooth indicators β_r in (2.61), for all $r=0,...,k-1$. Explicit formulae for $k=2$ and $k=3$ are given in (2.62) and (2.63) respectively.

4. Form the weights ω_r and $\tilde{\omega}_r$ using (2.58)-(2.59) and

$$\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\sum_{s=0}^{k-1}\tilde{\alpha}_s}, \qquad \tilde{\alpha}_r = \frac{\tilde{d}_r}{(\epsilon+\beta_r)^2}, \qquad r=0,...,k-1.$$

5. Find the $(2k-1)$-th order reconstruction

$$v^-_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r v^{(r)}_{i+\frac{1}{2}}, \qquad v^+_{i-\frac{1}{2}} = \sum_{r=0}^{k-1} \tilde{\omega}_r v^{(r)}_{i-\frac{1}{2}}\,. \tag{2.64}$$

□

We can obtain weights for higher orders of k (corresponding to seventh and higher order WENO schemes) using the same recipe. However, these schemes of seventh and higher order have not been extensively tested yet.

2.3 ENO and WENO Schemes for 1D Conservation Laws

In this section we describe the ENO and WENO schemes for 1D conservation laws:

$$u_t(x,t) + f_x(u(x,t)) = 0 \tag{2.65}$$

equipped with suitable initial and boundary conditions.

We will concentrate on the discussion of spatial discretization, and will leave the time variable t continuous (the method-of-lines approach). Time discretizations will be discussed in Sect. 4.2.

Our computational domain is $a \le x \le b$. We have a grid defined by (2.1), with the notations (2.2)-(2.3). Except for in Sect. 2.3.3, we do not consider boundary conditions. We thus assume that the values of the numerical solution are also available outside the computational domain whenever they are needed. This would be the case for periodic or compactly supported problems.

2.3.1 Finite volume formulation in the scalar case.

For finite volume schemes, or schemes based on cell averages, we do not solve (2.65) directly, but its integrated version. We integrate (2.65) over the interval I_i to obtain

$$\frac{d\overline{u}(x_i,t)}{dt} = -\frac{1}{\Delta x_i}\left(f(u(x_{i+\frac{1}{2}},t) - f(u(x_{i-\frac{1}{2}},t))\right), \tag{2.66}$$

where

$$\overline{u}(x_i,t) \equiv \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi,t)\,d\xi \tag{2.67}$$

is the cell average. We approximate (2.66) by the following conservative scheme

$$\frac{d\overline{u}_i(t)}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right), \tag{2.68}$$

where $\overline{u}_i(t)$ is the numerical approximation to the cell average $\overline{u}(x_i,t)$, and the numerical flux $\hat{f}_{i+\frac{1}{2}}$ is defined by

$$\hat{f}_{i+\frac{1}{2}} = h\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right) \tag{2.69}$$

with the values $u^{\pm}_{i+\frac{1}{2}}$ obtained by the ENO reconstruction Procedure 2.1, or by the WENO reconstruction Procedure 2.2.

The two argument function h in (2.69) is a monotone flux. It satisfies:

- $h(a,b)$ is a Lipschitz continuous function in both arguments;

- $h(a,b)$ is a nondecreasing function in a and a nonincreasing function in b. Symbolically $h(\uparrow,\downarrow)$;
- $h(a,b)$ is consistent with the physical flux f, that is, $h(a,a) = f(a)$.

Examples of monotone fluxes include:

1. Godunov flux:
$$h(a,b) = \begin{cases} \min_{a\le u\le b} f(u) & \text{if } a \le b \\ \max_{b\le u\le a} f(u) & \text{if } a > b \end{cases}, \tag{2.70}$$
2. Engquist-Osher flux:
$$h(a,b) = \int_0^a \max(f'(u),0)du + \int_0^b \min(f'(u),0)du + f(0). \tag{2.71}$$
3. Lax-Friedrichs flux:
$$h(a,b) = \frac{1}{2}\left[f(a) + f(b) - \alpha(b-a)\right] \tag{2.72}$$
where $\alpha = \max_u |f'(u)|$ is a constant. The maximum is taken over the relevant range of u.

We have listed the monotone fluxes from the least dissipative (less smearing of discontinuities) to the most. For lower order methods (order of reconstruction is 1 or 2), there is a big difference between results obtained by different monotone fluxes. However, this difference becomes much smaller for higher order reconstructions. In Fig. 2.2, we plot the results of a right moving shock for the Burgers' equation ($f(u) = \frac{u^2}{2}$ in (2.65)), with first order reconstruction using Godunov and Lax-Friedrichs monotone fluxes (top), and with fourth order ENO reconstruction using Godunov and Lax-Friedrichs monotone fluxes (bottom). We can clearly see that, while the Godunov flux behaves much better for the first order scheme, the two fourth order ENO schemes behave similarly. We thus use the simple and inexpensive Lax-Friedrichs flux in most of our high order calculations.

We remark that, by the classic Lax-Wendroff theorem [51], the solution to the conservative scheme (2.68), *if converges*, will converge to a weak solution of (2.65).

In summary, to build a finite volume ENO scheme (2.68), given the cell averages $\{\overline{u}_i\}$ (we will often drop the explicit reference to the time variable t), we proceed as follows:

Procedure 2.3. Finite volume 1D scalar ENO and WENO.

1. Follow the Procedure 2.1 in Sect. 2.2.1 for ENO, or the Procedure 2.2 in Sect. 2.2.2 for WENO, to obtain the k-th order reconstructed values $u^-_{i+\frac{1}{2}}$ and $u^+_{i+\frac{1}{2}}$ for all i;
2. Choose a monotone flux (e.g., one of (2.70) to (2.72)), and use (2.69) to compute the flux $\hat{f}_{i+\frac{1}{2}}$ for all i;
3. Form the scheme (2.68).

□

Notice that the finite volume scheme can be applied to arbitrary nonuniform grids.

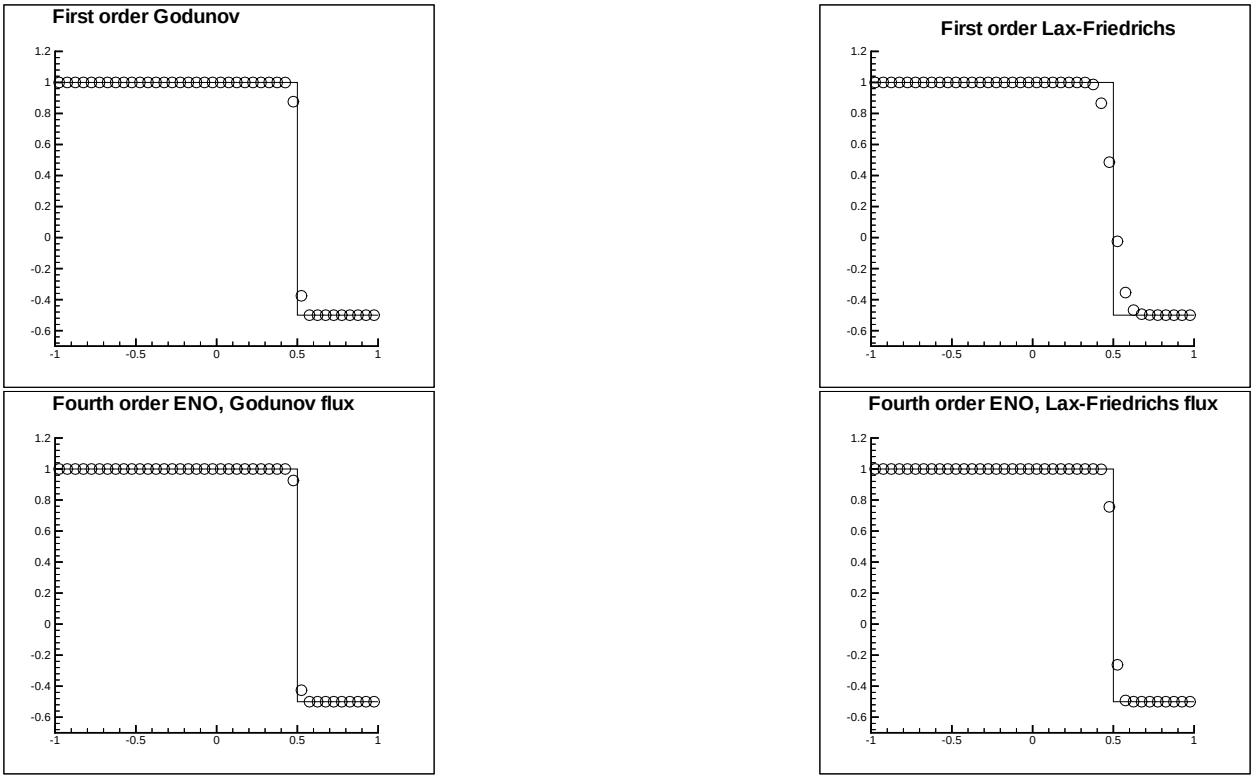


Figure 2.2: First order (top) and fourth order (bottom) ENO schemes for the Burgers equation, with the Godunov flux (left) and the Lax-Friedrichs flux (right). Solid lines: exact solution; Circles: the computed solution at $t = 4$.

2.3.2 Finite difference formulation in the scalar case.

We first assume the grid is uniform and solve (2.65) directly using a conservative approximation to the spatial derivative:

$$\frac{du_i(t)}{dt} = -\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right) \tag{2.73}$$

where $u_i(t)$ is the numerical approximation to the point value $u(x_i, t)$, and the numerical flux

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i-r}, ..., u_{i+s})$$

satisfies the following conditions:

- $\hat{f}$ is a Lipschitz continuous function in all the arguments;
- $\hat{f}$ is consistent with the physical flux f, that is, $\hat{f}(u, ..., u) = f(u)$.

Again the Lax-Wendroff theorem [51] applies. The solution to the conservative scheme (2.73), *if converges*, will converge to a weak solution of (2.65).

The numerical flux $\hat{f}_{i+\frac{1}{2}}$ is obtained by the ENO or WENO reconstruction procedures, Procedure 2.1 or 2.2, *with* $\overline{v}(x) = f(u(x,t))$. For stability, it is important that upwinding is used in constructing the flux. The easiest and the most inexpensive way to achieve upwinding is the following: compute the Roe speed

$$\overline{a}_{i+\frac{1}{2}} \equiv \frac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i}, \tag{2.74}$$

and

- if $\overline{a}_{i+\frac{1}{2}} \geq 0$, then the the wind blows from the left to the right. We would use $v^-_{i+\frac{1}{2}}$ for the numerical flux $\hat{f}_{i+\frac{1}{2}}$;
- if $\overline{a}_{i+\frac{1}{2}} < 0$, then the wind blows from the right to the left. We would use $v^+_{i+\frac{1}{2}}$ for the numerical flux $\hat{f}_{i+\frac{1}{2}}$.

This produces the Roe scheme [62] at the first order level. For this reason, the ENO scheme based on this approach was termed "ENO-Roe" in [70].

In summary, to build a finite difference ENO scheme (2.73) *using the ENO-Roe approach*, given the point values $\{u_i\}$ (we again drop the explicit reference to the time variable t), we proceed as follows:

Procedure 2.4. Finite difference 1D scalar ENO- and WENO-Roe.

1. Compute the Roe speed $\overline{a}_{i+\frac{1}{2}}$ for all i using (2.74);
2. Identify $\overline{v}_i = f(u_i)$ and use the ENO reconstruction Procedure 2.1 or the WENO reconstruction Procedure 2.2, to obtain the cell boundary values $v^-_{i+\frac{1}{2}}$ if $\overline{a}_{i+\frac{1}{2}} \geq 0$, or $v^+_{i+\frac{1}{2}}$ if $\overline{a}_{i+\frac{1}{2}} < 0$;

3. If the Roe speed at $x_{i+\frac{1}{2}}$ is positive

$$\overline{a}_{i+\frac{1}{2}} \geq 0,$$

then take the numerical flux as:

$$\hat{f}_{i+\frac{1}{2}} = v^-_{i+\frac{1}{2}};$$

otherwise, take the the numerical flux as:

$$\hat{f}_{i+\frac{1}{2}} = v^+_{i+\frac{1}{2}};$$

4. Form the scheme (2.73).

□

One disadvantage of the ENO-Roe approach is that entropy violating solutions may be obtained, just like in the first order Roe scheme case. For example, if ENO-Roe is applied to the Burgers equation

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

with the following initial condition

$$u(x,0) = \begin{cases} -1, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0, \end{cases}$$

it will converge to the entropy violating expansion shock:

$$u(x,t) = \begin{cases} -1, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases}$$

Local entropy correction could be used to remedy this [70]. However, it is usually more robust to use a global "flux splitting":

$$f(u) = f^+(u) + f^-(u) \tag{2.75}$$

where

$$\frac{df^+(u)}{du} \geq 0, \qquad \frac{df^-(u)}{du} \leq 0. \tag{2.76}$$

We would need the positive and negative fluxes $f^{\pm}(u)$ to have as many derivatives as the order of the scheme. This unfortunately rules out many popular fluxes (such as those of van Leer [79] and Osher [58]) for high order methods in this framework.

The simplest smooth splitting is the Lax-Friedrichs splitting:

$$f^{\pm}(u) = \frac{1}{2}(f(u) \pm \alpha u) \tag{2.77}$$

where α is again taken as $\alpha = \max_u |f'(u)|$ over the relevant range of u.

We note that there is a close relationship between a flux splitting (2.75) and a monotone flux (2.69). In fact, for any flux splitting (2.75) satisfying (2.76),

$$h(a,b) = f^+(a) + f^-(b) \tag{2.78}$$

is clearly a monotone flux. However, not every monotone flux can be written in the flux split form (2.75). For example, the Godunov flux (2.70) cannot.

With the flux splitting (2.75), we apply the the ENO or WENO reconstruction procedures, Procedure 2.1 or 2.2, with $\overline{v}(x) = f^+(u(x,t))$ and $\overline{v}(x) = f^-(u(x,t))$ separately, to obtain two numerical fluxes $\hat{f}^+_{i+\frac{1}{2}}$ and $\hat{f}^-_{i+\frac{1}{2}}$, and then sum them to get the numerical flux $\hat{f}_{i+\frac{1}{2}}$.

In summary, to build a finite difference (FD) ENO or WENO scheme (2.73) *using the flux splitting approach*, given the point values $\{u_i\}$, we proceed as follows:

Procedure 2.5. FD 1D scalar flux splitting ENO and WENO.

1. Find a smooth flux splitting (2.75), satisfying (2.76);
2. Identify $\overline{v}_i = f^+(u_i)$ and use the ENO or WENO reconstruction procedure, Procedure 2.1 or 2.2, to obtain the cell boundary values $v^-_{i+\frac{1}{2}}$ for all i;
3. Take the positive numerical flux as
$$\hat{f}^+_{i+\frac{1}{2}} = v^-_{i+\frac{1}{2}};$$
4. Identify $\overline{v}_i = f^-(u_i)$ and use the ENO or WENO reconstruction procedures, Procedure 2.1 or 2.2, to obtain the cell boundary values $v^+_{i+\frac{1}{2}}$ for all i;
5. Take the negative numerical flux as
$$\hat{f}^-_{i+\frac{1}{2}} = v^+_{i+\frac{1}{2}};$$
6. Form the numerical flux as
$$\hat{f}_{i+\frac{1}{2}} = \hat{f}^+_{i+\frac{1}{2}} + \hat{f}^-_{i+\frac{1}{2}};$$
7. Form the scheme (2.73).

□

We remark that the finite difference scheme in this section and the finite volume scheme in Sect. 2.3.1 are equivalent for one dimensional, linear PDE with constant coefficients: the only difference is in the initial conditions (one uses point values and the other uses cell averages of the exact initial condition). Notice that the schemes are still nonlinear in this case. However, this equivalence does not hold for a nonlinear PDE. Moreover, we will see later that there are significant differences in efficiency of the two approaches for multidimensional problems.

Table 2.2: Accuracy on $u_t + u_x = 0$ with $u_0(x) = \sin(\pi x)$.

Method	N	L_∞ error	L_∞ order	L_1 error	L_1 order
WENO-5	10	2.98e-2	-	1.60e-2	-
	20	1.45e-3	4.36	7.41e-4	4.43
	40	4.58e-5	4.99	2.22e-5	5.06
	80	1.48e-6	4.95	6.91e-7	5.01
	160	4.41e-8	5.07	2.17e-8	4.99
	320	1.35e-9	5.03	6.79e-10	5.00
CENTRAL-5	10	4.98e-3	-	3.07e-3	-
	20	1.60e-4	4.96	9.92e-5	4.95
	40	5.03e-6	4.99	3.14e-6	4.98
	80	1.57e-7	5.00	9.90e-8	4.99
	160	4.91e-9	5.00	3.11e-9	4.99
	320	1.53e-10	5.00	9.73e-11	5.00

In the following we test the accuracy of the fifth order finite difference WENO schemes on the linear equation:

$$\begin{aligned} u_t + u_x &= 0, \qquad -1 \le x \le 1 \\ u(x,0) &= u_0(x) \qquad \text{periodic.} \end{aligned}$$

In Table 2.2, we show the errors of the fifth order WENO scheme given by the weights (2.58)-(2.59) with the smooth indicator (2.63), at time $t = 1$ for the initial condition $u_0(x) = \sin(\pi x)$, and compare them with the errors of the linear 5^{th} order upstream central scheme (referred to as CENTRAL-5 in the following tables). We can see that fifth order WENO gives the expected order of accuracy starting at about 40 grid points.

In Table 2.3, we show errors for the initial condition $u_0(x) = \sin^4(\pi x)$. The order of accuracy for the fifth order WENO settles down later than in the previous example. Notice that this is the example for which ENO schemes lose their accuracy [63], [67].

We emphasize again that the high order conservative finite difference ENO and WENO schemes of third or higher order accuracy can only be applied to a uniform grid or a smoothly varying grid, i.e. a grid such that a smooth transformation

$$\xi = \xi(x)$$

will result in a uniform grid in the new variable ξ. Here ξ must contain as many derivatives as the order of accuracy of scheme calls for. If this is the case, then (2.65) is transformed to

$$u_t + \xi_x f(u)_\xi = 0$$

and the conservative ENO or WENO derivative approximation is then applied to $f(u)_\xi$. It is proven in [58] that this way the scheme is still conservative, i.e. Lax-Wendroff theorem [51] still applies.

Table 2.3: Accuracy on $u_t + u_x = 0$ with $u_0(x) = \sin^4(\pi x)$.

Method	N	L_∞ error	L_∞ order	L_1 error	L_1 order
WENO-5	20	1.08e-1	-	4.91e-2	-
	40	8.90e-3	3.60	3.64e-3	3.75
	80	1.80e-3	2.31	5.00e-4	2.86
	160	1.22e-4	3.88	2.17e-5	4.53
	320	4.37e-6	4.80	6.17e-7	5.14
	640	9.79e-8	5.48	1.57e-8	5.30
CENTRAL-5	20	5.23e-2	-	3.35e-2	-
	40	2.47e-3	4.40	1.52e-3	4.46
	80	8.32e-5	4.89	5.09e-5	4.90
	160	2.65e-6	4.97	1.60e-6	4.99
	320	8.31e-8	5.00	4.99e-8	5.00
	640	2.60e-9	5.00	1.56e-9	5.00

2.3.3 Boundary conditions.

For periodic boundary conditions, or problems with compact support for the entire computation (not just the initial data), there is no difficulty in implementing boundary conditions: one simply set as many ghost points as needed using either the periodicity condition or the compactness of the solution.

Other types of boundary conditions should be handled according to their type: for reflective or symmetry boundary conditions, one would set as many ghost points as needed, then use the symmetry/antisymmetry properties to prescribe solution values at those ghost points. For inflow or partially inflow (e.g. a subsonic outflow where one of the characteristic waves flows in) boundary conditions, one would usually use the physical inflow boundary condition at the exact boundary (for example, if $x_{\frac{1}{2}}$ is the left boundary and a finite volume scheme is used, one would use the given boundary value u_b as $u^-_{\frac{1}{2}}$ in the monotone flux at $x_{\frac{1}{2}}$; if x_0 is the left boundary and a finite difference scheme is used, one would use the given boundary value u_b as u_0). Apart from that, the most natural way of treating boundary conditions for the ENO scheme is to *use only the available values inside the computational domain when choosing the stencil*. In other words, only stencils completely contained inside the computational domain is used in the ENO stencil choosing process described in Procedure 2.1. In practical implementation, in order to avoid logical structures to distinguish whether a given stencil is completely inside the computational domain, one could set all the ghost values outside the computational domain to be very large with large variations (e.g. setting $u_{-j} = (10j)^{10}$ if x_{-j}, for $j = 1, 2, ...$, are ghost points). This way the ENO stencil choosing procedure will automatically avoid choosing any stencil containing ghost points. Another way of treating boundary conditions is to use extrapolation of suitable order to set the values of the solution in all necessary ghost points. For scalar problems this is actually equivalent to the approach of using only the stencils inside the computational domain in the ENO procedure. WENO can be handled in a similar fashion.

Stability analysis (GKS analysis [30], [76]) can be used to study the linear stability when the boundary treatment described above is applied to a fixed stencil upwind biased scheme. For most practical situations the schemes are linearly stable [3].

2.3.4 Provable properties in the scalar case.

Second order ENO schemes are also TVD (total variation diminishing), hence have at least subsequences which converge to weak solutions. There is no known convergence result for ENO schemes of degree higher than 2, even for smooth solutions.

WENO schemes have better convergence results, mainly because their numerical fluxes are smoother. It is proven [43] that WENO schemes converge for smooth solutions. Also, Jiang and Yu [44] have obtained an existence proof for traveling waves for WENO schemes. This is an important first step towards the proof of convergence for shocked cases.

Even though there are very little theoretical results about ENO or WENO schemes, in practice these schemes are very robust and stable. We caution against any attempts to modify the schemes solely for the purpose of stability or convergence proofs. In [69] we gave a remark about a modification of ENO schemes, which keeps the formal uniform high order accuracy and makes them stable and convergent for general multi dimensional scalar equations. However it was pointed out there that the modification is not computationally useful, hence the convergence result has little value.

The remark in [69] is illustrative hence we reproduce it here. We start with a flux splitting (2.75) satisfying (2.76), and notice that the first order monotone scheme

$$\frac{du_i}{dt} = -\frac{1}{\Delta x_i}\left(f^+(u_i) - f^+(u_{i-1}) + f^-(u_{i+1}) - f^-(u_i)\right) \equiv R_1(u)_i \tag{2.79}$$

is convergent (also for multi space dimensions). We now construct a high order ENO approximation in the following way: starting from the two point stencil $\{x_{i-1}, x_i\}$, we expand it into a $k+1$ point stencil in an ENO fashion using the divided differences of $f^+(u(x))$. We then build the k-th degree polynomial $P^+(x)$ which interpolates $f^+(u(x))$ in this stencil. $P^-(x)$ is constructed in a similar way, starting from the two point stencil $\{x_i, x_{i+1}\}$. The scheme is finally defined as

$$\frac{du_i}{dt} = -\frac{d}{dx}\left(P^+(x) + P^-(x)\right)\Big|_{x=x_i} \equiv R_k(u)_i \tag{2.80}$$

This scheme is clearly k-th order accurate but is not conservative. We now denote the difference between the high order scheme (2.80) and the first order monotone scheme (2.79) by

$$D(u)_i \equiv R_k(u)_i - R_1(u)_i, \tag{2.81}$$

and limit it by

$$\tilde{D}(u)_i = \overline{m}(D(u)_i, M\Delta x^\alpha), \tag{2.82}$$

where $M > 0$ and $0 < \alpha \leq 1$ are constants, and the capping function $\overline{m}$ is defined by

$$\overline{m}(a,b) = \begin{cases} a, & if \ \ |a| \leq b\,; \\ b, & if \ \ a > b\,; \\ -b, & if \ \ a < -b\,. \end{cases}$$

The modified ENO scheme is then defined by

$$\frac{du_i}{dt} = \tilde{R}_k(u)_i \equiv R_1(u)_i + \tilde{D}(u)_i. \tag{2.83}$$

We notice that, in smooth regions, the difference between the first order and high order residues, $D(u)_i$, as defined in (2.81), is of the size $O(\Delta x)$, hence the capping (2.82) does not take effect in such regions, if $\alpha < 1$ or if $\alpha = 1$ and M is large enough, when Δx is sufficiently small. This implies that the scheme (2.83) is uniformly accurate. Moreover, since

$$\left|\tilde{R}_k(u)_i - R_1(u)_i\right| \leq M\Delta x^{\alpha}$$

by (2.82), the high order scheme (2.83) shares every good property of the first order monotone scheme (2.79), such as total variation boundedness, entropy conditions, and convergence. From a theoretical point of view, this is the strongest result one could possibly hope for a high order scheme. However, the mesh size dependent limiting (2.82) renders the scheme highly impractical: the quality of the numerical solution will depend strongly on the choice of the parameters M and α, as well as on the mesh size Δx.

2.3.5 Systems.

We only consider hyperbolic $m \times m$ systems, i.e. the Jocobian $f'(u)$ has m real eigenvalues

$$\lambda_1(u) \leq ... \leq \lambda_m(u) \tag{2.84}$$

and a complete set of independent eigenvectors

$$r_1(u), ..., r_m(u)\,. \tag{2.85}$$

We denote the matrix whose columns are eigenvectors (2.85) by

$$R(u) = (r_1(u), ..., r_m(u)) \tag{2.86}$$

Then clearly

$$R^{-1}(u)\, f'(u)\, R(u) \;=\; \Lambda(u) \tag{2.87}$$

where $\Lambda(u)$ is the diagonal matrix with $\lambda_1(u), ..., \lambda_m(u)$ on the diagonal. Notice that the rows of $R^{-1}(u)$, denoted by $l_1(u), ..., l_m(u)$ (row vectors), are left eigenvectors of $f'(u)$:

$$l_i(u)f'(u) = \lambda_i(u)l_i(u), \qquad i = 1, ..., m\,. \tag{2.88}$$

There are several ways to generalize scalar ENO or WENO schemes to systems.

The easiest way is to apply the ENO or WENO schemes in a component by component fashion. For the finite volume (FV) formulation, this means that we make the reconstruction using ENO or WENO for each of the components of u separately. This produces the left and right values $u^{\pm}_{i+\frac{1}{2}}$ at the cell interface $x_{i+\frac{1}{2}}$. An exact or approximate Riemann solver, $h(u^{-}_{i+\frac{1}{2}}, u^{+}_{i+\frac{1}{2}})$, is then used to build the scheme (2.68)-(2.69). The exact Riemann solver is given by the exact solution of (2.65) with the following step function as initial condition

$$u(x,0) = \begin{cases} u^{-}_{i+\frac{1}{2}}, & x \leq 0; \\ u^{+}_{i+\frac{1}{2}}, & x > 0. \end{cases} \tag{2.89}$$

evaluated at the center $x = 0$. Notice that the solution to (2.65) with the initial condition (2.89) is self-similar, that is, it is a function of the variable $\xi = \frac{x}{t}$, hence is constant along $x = 0$. If we denote this solution by $u_{i+\frac{1}{2}}$, then the flux is taken as

$$h(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}) = f(u_{i+\frac{1}{2}}).$$

In the scalar case, the exact Riemann solver gives the Godunov flux (2.70). Exact Riemann solver can be obtained for many systems including the Euler equations of compressible gas, which is used very often in practice. However, it is usually very costly to get this solution (for Euler equations of compressible gas, an iterative procedure is needed to obtain this solution, see [74]). In practice, approximate Riemann solvers are usually good enough. As in the scalar case, the quality of the solution is usually very sensitive to the choice of approximate Riemann solvers for *lower order* schemes (first or second order), but this sensitivity decreases with an increasing order of accuracy. The simplest approximate Riemann solver (albeit the most dissipative) is again the Lax-Friedrichs solver (2.72), except that now the constant α is taken as

$$\alpha = \max_u \max_{1 \le j \le m} |\lambda_j(u)| \tag{2.90}$$

where $\lambda_j(u)$ are the eigenvalues of the Jacobian $f'(u)$, (2.84). The maximum is again taken over the relevant range of u.

We summarize the procedure in the following

Procedure 2.6. Component-wise FV 1D system ENO and WENO.

1. For each component of the solution $\overline{u}$, apply the scalar ENO Procedure 2.1 or WENO Procedure 2.2 to reconstruct the corresponding component of the solution at the cell interfaces, $u^{\pm}_{i+\frac{1}{2}}$ for all i;

2. Apply an exact or approximate Riemann solver to compute the flux $\hat{f}_{i+\frac{1}{2}}$ for all i in (2.69);

3. Form the scheme (2.68).

□

For the finite difference formulation, a smooth flux splitting (2.75) is again needed. The condition (2.76) now becomes that the two Jacobians

$$\frac{\partial f^+(u)}{\partial u}, \qquad \frac{\partial f^-(u)}{\partial u} \tag{2.91}$$

are still diagonalizable (preferably by the same eigenvectors $R(u)$ as for $f'(u)$), and have only non-negative / non-positive eigenvalues, respectively. We again recommend the Lax-Friedrichs flux splitting (2.77), with α given by (2.90), because of its simplicity and smoothness. A somewhat more complicated Lax-Friedrichs type flux splitting is:

$$f^{\pm}(u) = \frac{1}{2}(f(u) \pm R(u)\,\overline{\Lambda}\,R^{-1}(u)\,u)\,,$$

where $R(u)$ and $R^{-1}(u)$ are defined in (2.86), and

$$\overline{\Lambda} = diag(\overline{\lambda}_1, ..., \overline{\lambda}_m)$$

where $\overline{\lambda}_j = max_u|\lambda_j(u)|$, and the maximum is again taken over the relevant range of u. This way the dissipation is added in each field according to the maximum size of eigenvalues in that field, not globally. One could also use other flux splittings, such as the van Leer splitting for gas dynamics [79]. However, for higher order schemes, the flux splitting must be sufficiently smooth in order to retain the order of accuracy.

With these flux splittings, we can again use the scalar recipes to form the finite difference scheme: just compute the positive and negative fluxes $\hat{f}^+_{i+\frac{1}{2}}$ and $\hat{f}^-_{i+\frac{1}{2}}$ component by component.

We summarize the procedure in the following

Procedure 2.7. Component-wise FD 1D system ENO and WENO.

1. Find a flux splitting (2.75). The simplest example is the Lax-Friedrichs flux splitting (2.77), with α given by (2.90);

2. For each component of the solution u, apply the scalar Procedure 2.5 to reconstruct the corresponding component of the numerical flux $\hat{f}_{i+\frac{1}{2}}$;

3. Form the scheme (2.73).

□

These component by component versions of ENO and WENO schemes are simple and cost effective. They work reasonably well for many problems, especially when the order of accuracy is not high (second or sometimes third order). However, for more demanding test problems, or when the order of accuracy is high, we would need the following more costly, but much more robust characteristic decompositions.

To explain the characteristic decomposition, we start with a simple example where $f(u) = Au$ in (2.65) is linear and A is a constant matrix. In this situation, the eigenvalues (2.84), the eigenvectors (2.85), and the related matrices R, R^{-1} and Λ (2.86)-(2.87), are all constant matrices. If we define a change of variable

$$v = R^{-1}\, u, \tag{2.92}$$

then the PDE (2.65) becomes diagonal:

$$v_t + \Lambda v_x = 0 \tag{2.93}$$

that is, the m equations in (2.93) are decoupled and each one is a scalar linear convection equation of the form

$$w_t + \lambda_j w_x = 0. \tag{2.94}$$

We can thus use the reconstruction or flux evaluation techniques for the scalar equations, discussed in Sections 2.3.1 and 2.3.2, to handle each of the equations in (2.94). After we obtain the results, we can "come back" to the physical space u by using the inverse of (2.92):

$$u = R\, v \tag{2.95}$$

For example, if the reconstructed polynomial for each component j in (2.93) is denoted by $q_j(x)$, then we form

$$q(x) = \begin{pmatrix} q_1(x) \\ \cdot \\ \cdot \\ \cdot \\ q_m(x) \end{pmatrix} \tag{2.96}$$

and obtain the reconstruction in the physical space by using (2.95):

$$p(x) = R\, q(x) \tag{2.97}$$

The flux evaluations for the finite difference schemes can be handled similarly.

We now come to the situation where $f'(u)$ is not constant. The trouble is that now all the matrices $R(u)$, $R^{-1}(u)$ $\Lambda(u)$ are dependent upon u. We must "freeze" them locally in order to carry out a similar procedure as in the constant coefficient case. Thus, to compute the flux at the cell boundary $x_{i+\frac{1}{2}}$, we would need an approximation to the Jocobian at the middle value $u_{i+\frac{1}{2}}$. This can be simply taken as the arithmetic mean

$$u_{i+\frac{1}{2}} = \frac{1}{2}\left(u_i + u_{i+1}\right), \tag{2.98}$$

or as a more elaborate average satisfying some nice properties, e.g. the mean value theorem

$$f(u_{i+1}) - f(u_i) = f'(u_{i+\frac{1}{2}})(u_{i+1} - u_i)\,. \tag{2.99}$$

Roe average [62] is such an example for the compressible Euler equations of gas dynamics and some other physical systems. It is also possible to use two different one-sided Jacobians at a higher computational cost [25].

Once we have this $u_{i+\frac{1}{2}}$, we will use $R(u_{i+\frac{1}{2}})$, $R^{-1}(u_{i+\frac{1}{2}})$ and $\Lambda(u_{i+\frac{1}{2}})$ to help evaluating the numerical flux at $x_{i+\frac{1}{2}}$. We thus omit the notation $i+\frac{1}{2}$ and still denote these matrices by R, R^{-1} and Λ, etc. We then repeat the procedure described above for linear systems. The difference here being, the matrices R, R^{-1} and Λ are different at different locations $x_{i+\frac{1}{2}}$, hence the cost of the operation is greatly increased.

In summary, we have the following procedures:

Procedure 2.8. Characteristic-wise FV 1D ENO and WENO.

1. Compute the divided or undivided differences of the cell averages $\overline{u}$, for all i;
2. At each fixed $x_{i+\frac{1}{2}}$, do the following:
   (a) Compute an average state $u_{i+\frac{1}{2}}$, using either the simple mean (2.98) or a Roe average satisfying (2.99);
   (b) Compute the right eigenvectors, the left eigenvectors, and the eigenvalues of the Jacobian $f'(u_{i+\frac{1}{2}})$, (2.84)-(2.87), and denote them by

$$R = R(u_{i+\frac{1}{2}}), \qquad R^{-1} = R^{-1}(u_{i+\frac{1}{2}}), \qquad \Lambda = \Lambda(u_{i+\frac{1}{2}});$$

(c) Transform all those differences computed in Step 1, which are in the potential stencil of the ENO and WENO reconstructions for obtaining $u^{\pm}_{i+\frac{1}{2}}$, to the local characteristic fields by using (2.92). For example,

$$\overline{v}_j = R^{-1}\,\overline{u}_j, \qquad \text{j in a neighborhood of i;}$$

(d) Perform the scalar ENO or WENO reconstruction Procedure 2.3, for each component of the characteristic variables $\overline{v}$, to obtain the corresponding component of the reconstruction $v^{\pm}_{i+\frac{1}{2}}$;

(e) Transform back into physical space by using (2.95):

$$u^{\pm}_{i+\frac{1}{2}} = R\,v^{\pm}_{i+\frac{1}{2}}$$

3. Apply an exact or approximate Riemann solver to compute the flux $\hat{f}_{i+\frac{1}{2}}$ for all i in (2.69); then form the scheme (2.68).

□

Similarly, the procedure to obtain a finite difference ENO-Roe type scheme using the local characteristic variables is:

Procedure 2.9. Characteristic-wise FD 1D system, Roe-type.

1. Compute the divided or undivided differences of the flux $f(u)$ for all i;

2. At each fixed $x_{i+\frac{1}{2}}$, do the following:

   (a) Compute an average state $u_{i+\frac{1}{2}}$, using either the simple mean (2.98) or a Roe average satisfying (2.99);

   (b) Compute the right eigenvectors, the left eigenvectors, and the eigenvalues of the Jacobian $f'(u_{i+\frac{1}{2}})$, (2.84)-(2.87), and denote them by

$$R = R(u_{i+\frac{1}{2}}), \qquad R^{-1} = R^{-1}(u_{i+\frac{1}{2}}), \qquad \Lambda = \Lambda(u_{i+\frac{1}{2}});$$

   (c) Transform all those differences computed in Step 1, which are in the potential stencil of the ENO and WENO reconstructions for obtaining the flux $\hat{f}_{i+\frac{1}{2}}$, to the local characteristic fields by using (2.92). For example,

$$v_j = R^{-1}\,f(u_j), \qquad \text{j in a neighborhood of i;}$$

   (d) Perform the scalar ENO or WENO Roe-type Procedure 2.4, for each component of the characteristic variables v, to obtain the corresponding component of the flux $\hat{v}_{i+\frac{1}{2}}$. The Roe speed $\overline{a}_{i+\frac{1}{2}}$ is replaced by the eigenvalue $\lambda_l(u_{i+\frac{1}{2}})$ for the l-th component of the characteristic variables v;

(e) Transform back into physical space by using (2.95):

$$\hat{f}_{i+\frac{1}{2}} = R\,\hat{v}_{i+\frac{1}{2}}$$

3. Form the scheme (2.73).

□

Finally, the procedure to obtain a finite difference flux splitting ENO or WENO scheme using the local characteristic variables is:

Procedure 2.10. Characteristic-wise FD 1D system, flux splitting.

1. Compute the divided or undivided differences of the flux $f(u)$ and the solution u for all i;

2. At each fixed $x_{i+\frac{1}{2}}$, do the following:

   (a) Compute an average state $u_{i+\frac{1}{2}}$, using either the simple mean (2.98) or a Roe average satisfying (2.99);

   (b) Compute the right eigenvectors, the left eigenvectors, and the eigenvalues of the Jacobian $f'(u_{i+\frac{1}{2}})$, (2.84)-(2.87), and denote them by

   $$R = R(u_{i+\frac{1}{2}}), \qquad R^{-1} = R^{-1}(u_{i+\frac{1}{2}}), \qquad \Lambda = \Lambda(u_{i+\frac{1}{2}});$$

   (c) Transform all those differences computed in Step 1, which are in the potential stencil of the ENO and WENO reconstructions for obtaining the flux $\hat{f}_{i+\frac{1}{2}}$, to the local characteristic fields by using (2.92). For example,

   $$v_j = R^{-1}\,u_j, \qquad g_j = R^{-1}\,f(u_j), \qquad \text{j in a neighborhood of i;}$$

   (d) Perform the scalar flux splitting ENO or WENO Procedure 2.5, for each component of the characteristic variables, to obtain the corresponding component of the flux $\hat{g}^{\pm}_{i+\frac{1}{2}}$. For the most commonly used Lax-Friedrichs flux splitting, we can use, for the l-th component of the characteristic variables, the viscosity coefficient

   $$\alpha = \max_{1\le j\le N} |\lambda_l(u_j)|;$$

   (e) Transform back into physical space by using (2.95):

   $$\hat{f}^{\pm}_{i+\frac{1}{2}} = R\,\hat{g}^{\pm}_{i+\frac{1}{2}}$$

3. Form the flux by taking

   $$\hat{f}_{i+\frac{1}{2}} = \hat{f}^{+}_{i+\frac{1}{2}} + \hat{f}^{-}_{i+\frac{1}{2}}$$

   and then form the scheme (2.73).

There are attempts recently to simplify this characteristic decomposition. For example, for the compressible Euler equations of gas dynamics, Jiang and Shu [43] used smooth indicators based on density and pressure to perform the so-called pseudo characteristic decompositions. There are also second and sometimes third order component ENO type schemes [56], [54], with limited success for higher order methods.

3 Multi Space Dimensions

3.1 Reconstruction and Approximation in Multi Dimensions

In this section we describe how the ideas of reconstruction and approximation in Sect. 2.1 are generalized to multi space dimensions. We will concentrate our discussion in 2D, although things carry over to higher dimensions as well.

In most part of this section we will consider Cartesian grids, that is, the domain is a rectangle

$$[a,b]\times[c,d] \tag{3.1}$$

covered by cells

$$I_{ij}\equiv[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}]\times[y_{j-\frac{1}{2}},y_{j+\frac{1}{2}}],\qquad 1\le i\le N_x,\ 1\le j\le N_y \tag{3.2}$$

where

$$a=x_{\frac{1}{2}}<x_{\frac{3}{2}}<...<x_{N_x-\frac{1}{2}}<x_{N_x+\frac{1}{2}}=b,$$

and

$$c=y_{\frac{1}{2}}<y_{\frac{3}{2}}<...<y_{N_y-\frac{1}{2}}<y_{N_y+\frac{1}{2}}=d.$$

The centers of the cells are

$$(x_i,y_j),\qquad x_i\equiv\frac{1}{2}\left(x_{i-\frac{1}{2}}+x_{i+\frac{1}{2}}\right),\ y_j\equiv\frac{1}{2}\left(y_{j-\frac{1}{2}}+y_{j+\frac{1}{2}}\right), \tag{3.3}$$

and we still use

$$\Delta x_i\equiv x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}},\qquad i=1,2,...,N_x \tag{3.4}$$

and

$$\Delta y_j\equiv y_{j+\frac{1}{2}}-y_{j-\frac{1}{2}},\qquad j=1,2,...,N_y \tag{3.5}$$

to denote the grid sizes. We denote the maximum grid sizes by

$$\Delta x\equiv\max_{1\le i\le N_x}\Delta x_i,\qquad \Delta y\equiv\max_{1\le j\le N_y}\Delta y_j, \tag{3.6}$$

and assume that Δx and Δy are of the same magnitude (their ratio is bounded from above and below during refinement). Finally,

$$\Delta\equiv\max(\Delta x,\Delta y). \tag{3.7}$$

3.1.1 Reconstruction from cell averages.

The approximation problem we will face, in solving hyperbolic conservation laws using cell averages (finite volume schemes, see Sect. 3.3), is still the following *reconstruction* problem.

Problem 3.1. Two dimensional reconstruction.

Given the cell averages of a function $v(x,y)$:

$$\overline{v}_{ij}\equiv\frac{1}{\Delta x_i\Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}v(\xi,\eta)\,d\xi\,d\eta,\qquad i=1,2,...,N_x,\ j=1,2,...,N_y, \tag{3.8}$$

find a polynomial $p_{ij}(x,y)$, preferably of degree at most $k-1$, for each cell I_{ij}, such that it is a k-th order accurate approximation to the function $v(x,y)$ inside I_{ij}:

$$p_{ij}(x,y) = v(x,y) + O(\Delta^k), \qquad (x,y) \in I_{ij},\ i = 1, ..., N_x,\ j = 1, ..., N_y. \tag{3.9}$$

In particular, this gives approximations to the function $v(x,y)$ at the cell boundaries

$$v^-_{i+\frac{1}{2},y} = p_{ij}(x_{i+\frac{1}{2}}, y), \quad v^+_{i-\frac{1}{2},y} = p_{ij}(x_{i-\frac{1}{2}}, y), \quad i = 1, ..., N_x,\ y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}}$$
$$v^-_{x,j+\frac{1}{2}} = p_{ij}(x, y_{j+\frac{1}{2}}), \quad v^+_{x,j-\frac{1}{2}} = p_{ij}(x, y_{j-\frac{1}{2}}), \quad j = 1, ..., N_y,\ x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}}$$

which are k-th order accurate:

$$v^\pm_{i+\frac{1}{2},y} = v(x_{i+\frac{1}{2}}, y) + O(\Delta^k), \qquad i = 0, 1, ..., N_x, \quad y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}} \tag{3.10}$$

and

$$v^\pm_{x,j+\frac{1}{2}} = v(x, y_{j+\frac{1}{2}}) + O(\Delta^k), \qquad j = 0, 1, ..., N_y, \quad x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}}\,. \tag{3.11}$$

□

Again we will not discuss boundary conditions in this section. We thus assume that $\overline{v}_{ij}$ is also available for $i \le 0$, $i > N_x$ and for $j \le 0$, $j > N_y$ if needed.

In the following we describe the general procedure to solve Problem 3.1.

Given the location I_{ij} and the order of accuracy k, we again first choose a "stencil", based on $\frac{k(k+1)}{2}$ neighboring cells, the collection of these cells still being denoted by $S(i,j)$. We then try to find a polynomial of degree at most $k-1$, denoted by $p(x,y)$ (we again drop the subscript ij when it does not cause confusion), whose cell average in each of the cells in $S(i,j)$ agrees with that of $v(x,y)$:

$$\frac{1}{\Delta x_l \Delta y_m} \int_{y_{m-\frac{1}{2}}}^{y_{m+\frac{1}{2}}} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} p(\xi,\eta)\, d\xi\, d\eta \;=\; \overline{v}_{lm}, \qquad if\ \ I_{lm} \in S(i,j). \tag{3.12}$$

We first remark that there are now many more candidate stencils $S(i,j)$ than in the 1D case, More importantly, unlike in the 1D case, here we encounter the following essential difficulties:

- Not all of the candidate stencils can be used to obtain a polynomial $p(x,y)$ of degree at most $k-1$ satisfying condition (3.12).

  For example, it is an easy exercise to show that neither existence nor uniqueness holds, if one wants to reconstruct a first degree polynomial $p(x,y)$ satisfying (3.12) for the three horizontal cells

  $$S(i,j) = \{I_{i-1,j}, I_{ij}, I_{i+1,j}\}\,.$$

  To see this, let's assume that

  $$I_{i-1,j} = [-2\Delta, -\Delta] \times [0, \Delta], \qquad I_{i,j} = [-\Delta, 0] \times [0, \Delta],$$
  $$I_{i+1,j} = [0, \Delta] \times [0, \Delta],$$

and the first degree polynomial $p(x,y)$ is given by

$$p(x,y) = \alpha + \beta x + \gamma y$$

then condition (3.12) implies

$$\left\{ \begin{array}{rcl} \alpha - \frac{3}{2}\Delta\,\beta + \frac{1}{2}\Delta\,\gamma & = & \overline{v}_{i-1,j} \\ \alpha - \frac{1}{2}\Delta\,\beta + \frac{1}{2}\Delta\,\gamma & = & \overline{v}_{i,j} \\ \alpha + \frac{1}{2}\Delta\,\beta + \frac{1}{2}\Delta\,\gamma & = & \overline{v}_{i+1,j} \end{array} \right.$$

which is a singular linear system for α, β and γ.

- Even if one obtains such a polynomial $p(x,y)$, there is no guarantee that the accuracy conditions (3.9) will hold. We again use the same simple example. If we pick the function

  $$v(x,y) = 0,$$

  then one of the polynomials of degree one satisfying the condition (3.12) is

  $$p(x,y) = \Delta - 2y$$

  clearly the difference

  $$v(x,0) - p(x,0) = -\Delta$$

  is not at the size of $O(\Delta^2)$ in $x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}}$, as is required by (3.9).

This difficulty will be more profound for unstructured meshes such as triangles. See, for example, [1].

For rectangular meshes, if we use the tensor products of 1D polynomials, i.e. use polynomials in Q^{k-1}:

$$p(x,y) = \sum_{m=0}^{k-1} \sum_{l=0}^{k-1} a_{lm} x^l y^m$$

then things can proceed as in 1D. We restrict ourselves in the following tensor product stencils:

$$S_{rs}(i,j) = \{I_{lm} : i - r \le l \le i + k - 1 - r,\ j - s \le m \le j + k - 1 - s\}$$

then we can address Problem 3.1 by introducing the two dimensional primitives:

$$V(x,y) = \int_{-\infty}^{y} \int_{-\infty}^{x} v(\xi,\eta) d\xi d\eta\ .$$

Clearly

$$\begin{aligned} V(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) &= \int_{-\infty}^{y_{j+\frac{1}{2}}} \int_{-\infty}^{x_{i+\frac{1}{2}}} v(\xi,\eta) d\xi d\eta \\ &= \sum_{m=-\infty}^{j} \sum_{l=-\infty}^{i} \overline{v}_{lm} \Delta x_l \Delta y_m\ , \end{aligned}$$

hence as in the 1D case, with the knowledge of the cell averages $\overline{v}$ we know the primitive function V exactly at cell corners.

On a tensor product stencil

$$\tilde{S}_{rs}(i,j) = \{(x_{l+\frac{1}{2}}, y_{m+\frac{1}{2}}) : i-r-1 \le l \le i+k-1-r, \quad j-s-1 \le m \le j+k-1-s\}$$

there is a unique polynomial $P(x,y)$ in Q^k which interpolates V at every point in $\tilde{S}_{rs}(i,j)$. We take the mixed derivative of the polynomial P to get:

$$p(x,y) = \frac{\partial^2 P(x,y)}{\partial x \partial y}$$

then $p(x,y)$ is in Q^{k-1}, approximates $v(x,y)$, which is the mixed derivative of $V(x,y)$, to k-th order:

$$v(x,y) - p(x,y) = O(\Delta^k)$$

and also satisfies (3.12):

$$\begin{aligned}
& \frac{1}{\Delta x_l \Delta y_m} \int_{y_{m-\frac{1}{2}}}^{y_{m+\frac{1}{2}}} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} p(\xi,\eta)\, d\xi\, d\eta \\
= & \frac{1}{\Delta x_l \Delta y_m} \int_{y_{m-\frac{1}{2}}}^{y_{m+\frac{1}{2}}} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} \frac{\partial^2 P}{\partial \xi \partial \eta}(\xi,\eta)\, d\xi\, d\eta \\
= & \frac{1}{\Delta x_l \Delta y_m} \Big(P(x_{l+\frac{1}{2}}, y_{m+\frac{1}{2}}) - P(x_{l+\frac{1}{2}}, y_{m-\frac{1}{2}}) \\
& \quad - P(x_{l-\frac{1}{2}}, y_{m+\frac{1}{2}}) + P(x_{l-\frac{1}{2}}, y_{m-\frac{1}{2}}) \Big) \\
= & \frac{1}{\Delta x_l \Delta y_m} \Big(V(x_{l+\frac{1}{2}}, y_{m+\frac{1}{2}}) - V(x_{l+\frac{1}{2}}, y_{m-\frac{1}{2}}) \\
& \quad - V(x_{l-\frac{1}{2}}, y_{m+\frac{1}{2}}) + V(x_{l-\frac{1}{2}}, y_{m-\frac{1}{2}}) \Big) \\
= & \frac{1}{\Delta x_l \Delta y_m} \int_{y_{m-\frac{1}{2}}}^{y_{m+\frac{1}{2}}} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} v(\xi,\eta)\, d\xi\, d\eta \;=\; \overline{v}_{lm}\,, \\
& \qquad i-r \le l \le i+k-1-r,\ j-s \le m \le j+k-1-s.
\end{aligned}$$

This gives us a practical way to perform the reconstruction in 2D. We first perform a one dimensional reconstruction (Problem 2.1), say in the y direction, obtaining one dimensional cell averages of the function v in the other direction (say in the x direction). We then perform a reconstruction in the other direction.

It should be remarked that the cost to do this 2D reconstruction is very high: For each grid point, if the cost to perform a one dimensional reconstruction is c, then we need $2c$ per grid point to perform this 2D reconstruction. In general n space dimensions, the cost grows to nc.

We also remark that to use polynomials in Q^{k-1} is a waste: to get the correct order of accuracy only polynomials in P^{k-1} is needed. However, there is no natural way of utilizing polynomials in P^{k-1} (see the comments above and also the paper of Abgrall [1]).

The reconstruction problem, Problem 3.1, can also be raised for general, non-Cartesian meshes, such as triangles. However, the solution becomes much more complicated. For discussions, see for example [1].

3.1.2 Conservative approximation to the derivative from point values.

The second approximation problem we will face, in solving hyperbolic conservation laws using point values (finite difference schemes, see Sect. 3.2), is again the following problem in obtaining high order conservative approximation to the derivative from point values [69, 70]. As in the 1D case, here we also assume that the grid is uniform in each direction. We again ignore the boundary conditions and assume that v_{ij} is available for $i \leq 0$ and $i > N_x$, and for $j \leq 0$ and $j > N_y$.

Problem 3.2. Two dimensional conservative approximation.

Given the point values of a function $v(x, y)$:

$$v_{ij} \equiv v(x_i, y_j), \qquad i = 1, 2, ..., N_x, \quad j = 1, 2, ..., N_y, \tag{3.13}$$

find numerical flux functions

$$\hat{v}_{i+\frac{1}{2},j} \equiv \hat{v}(v_{i-r,j}, ..., v_{i+k-1-r,j}), \qquad i = 0, 1, ..., N_x \tag{3.14}$$

and

$$\hat{v}_{i,j+\frac{1}{2}} \equiv \hat{v}(v_{i,j-s}, ..., v_{i,j+k-1-s}), \qquad j = 0, 1, ..., N_y \tag{3.15}$$

such that the flux differences approximate the derivatives $v_x(x, y)$ and $v_y(x, y)$ to k-th order accuracy:

$$\frac{1}{\Delta x}\left(\hat{v}_{i+\frac{1}{2},j} - \hat{v}_{i-\frac{1}{2},j}\right) = v_x(x_i, y_j) + O(\Delta x^k), \qquad i = 0, 1, ..., N_x, \tag{3.16}$$

and

$$\frac{1}{\Delta y}\left(\hat{v}_{i,j+\frac{1}{2}} - \hat{v}_{i,j-\frac{1}{2}}\right) = v_y(x_i, y_j) + O(\Delta y^k), \qquad j = 0, 1, ..., N_y, \tag{3.17}$$

□

The solution of this problem is essential for the high order conservative schemes based on point values (finite difference) rather than on cell averages (finite volume).

Having seen the complication of reconstructions in the previous subsection for multi space dimensions, it is a good relieve to see that conservative approximation to the derivative from point values is as simple in multi dimensions as in 1D. In fact, for fixed j, if we take

$$w(x) = v(x, y_j)$$

then to obtain $v_x(x_i, y_j) = w'(x_i)$ we only need to perform the one dimensional procedure in Sect. 2.1, Problem 2.2, to the one dimensional function $w(x)$. Same thing for $v_y(x, y)$.

As in the 1D case, the conservative approximation to derivatives, of third order accuracy or higher, can only be applied to uniform or smoothly varying meshes (curvilinear coordinates). It cannot be applied to general unstructured meshes such as triangles, unless conservative is given up.

3.2 ENO and WENO Approximations in Multi Dimensions

For solving hyperbolic conservation laws in multi space dimensions, we are again interested in the class of piecewise smooth functions. We define a piecewise smooth function $v(x, y)$ to be such that, for each fixed y, the one dimensional function $w(x) = v(x, y)$ is piecewise smooth in the sense described in Sect. 2.2. Likewise, for each fixed x, the one dimensional function $w(y) = v(x, y)$ is also assumed to be piecewise smooth. Such functions are again "generic" for solutions to multi dimensional hyperbolic conservation laws in practice.

In the previous section, we have already discussed the problems of reconstruction and conservative approximations to derivatives in multi space dimensions. At least for the Cartesian type grids, both the reconstruction and the conservative approximation can be obtained from one dimensional procedures.

For the reconstruction, we first use a one dimensional ENO or WENO reconstruction, Procedure 2.1 or 2.2, on the two dimensional cell averages, say in the y direction, to obtain one dimensional cell averages in x only. Then, another one dimensional reconstruction in the remaining direction, say in the x direction, is performed to recover the function itself, again using the one dimensional ENO or WENO methodology, Procedure 2.1 or 2.2.

For the conservative approximation to derivatives, since they are already formulated in a dimension by dimension fashion, one dimensional ENO and WENO procedures can be trivially applied. In effect, the FORTRAN program for the 2D problem is the same as the one for the 1D problem, with an outside "do loop".

What happens to general geometry which cannot be covered by a Cartesian grid?

If the domain is smooth enough, it usually can be mapped *smoothly* to a rectangle (or at least to a union of non-overlapping rectangles). That is, the transformation

$$\xi = \xi(x, y), \qquad \eta = \eta(x, y) \tag{3.18}$$

maps the physical domain Ω where (x, y) belongs, to a rectangular computational domain

$$a \le \xi \le b, \qquad c \le \eta \le d\,. \tag{3.19}$$

We require the transformation functions (3.18) to be smooth (i.e. it has as many derivatives as the accuracy of the scheme calls for). Using chain rule, we could write, for example,

$$v_x = \xi_x v_\xi + \eta_x v_\eta \tag{3.20}$$

We can then use our ENO or WENO approximations on v_ξ and v_η, as they are now defined in rectangular domains. The smoothness of ξ_x and η_x will guarantee that this leads to a high order approximation to v_x as well through (3.20).

If the domain is really ugly, or if one wants to use unstructured meshes for other purposes (e.g. for adaptivity), then ENO and WENO approximations for unstructured meshes must be studied. This is a much less matured subject at present. We refer the readers to [1] for some efforts in this direction.

3.3 ENO and WENO Schemes for Multi Dimensional Conservation Laws

In this section we describe the ENO and WENO schemes for 2D conservation laws:

$$u_t(x, y, t) + f_x(u(x, y, t)) + g_y(u(x, y, t)) = 0 \tag{3.21}$$

again equipped with suitable initial and boundary conditions.

Although we present everything in 2D, most of the discussion is also valid for higher dimensions.

We again concentrate on the discussion of spatial discretizations, and will leave the time variable t continuous (the method-of-lines approach). Time discretizations will be discussed in Sect. 4.2.

In most of the discussion in this section, our computational domain is rectangular, given by (3.1). Our grids will thus be Cartesian, given by (3.2) and (3.3). Unstructured meshes will only be mentioned briefly.

We do not discuss boundary conditions in this section. We thus assume that the values of the numerical solution are also available outside the computational domain whenever they are needed. This would be the case for periodic or compactly supported problems. Two dimensional boundary condition treatments are similar to the one dimensional case discussed in Sect. 2.3.3.

3.3.1 Finite volume formulation in the scalar case.

For finite volume schemes, or schemes based on cell averages, we do not solve (3.21) directly, but its integrated version. We integrate (3.21) over the interval I_{ij} to obtain

$$\begin{aligned}\frac{d\overline{u}_{ij}(t)}{dt} &= -\frac{1}{\Delta x_i \Delta y_j}\left(\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(u(x_{i+\frac{1}{2}},y,t)\,dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(u(x_{i-\frac{1}{2}},y,t))\,dy\right.\\ &\left.+\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(u(x,y_{j+\frac{1}{2}},t)\,dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(u(x,y_{j-\frac{1}{2}},t))\,dx\right)\end{aligned} \tag{3.22}$$

where

$$\overline{u}_{ij}(t) \equiv \frac{1}{\Delta x_i \Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi,\eta,t)\,d\xi\,d\eta \tag{3.23}$$

is the cell average. We approximate (3.22) by the following conservative scheme

$$\frac{d\overline{u}_{ij}(t)}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+\frac{1}{2},j}-\hat{f}_{i-\frac{1}{2},j}\right)-\frac{1}{\Delta y_j}\left(\hat{g}_{i,j+\frac{1}{2}}-\hat{g}_{i,j-\frac{1}{2}}\right), \tag{3.24}$$

where the numerical flux $\hat{f}_{i+\frac{1}{2},j}$ is defined by

$$\hat{f}_{i+\frac{1}{2},j} = \sum_\alpha w_\alpha h\left(u^-_{i+\frac{1}{2},y_j+\beta_\alpha\Delta y_j}, u^+_{i+\frac{1}{2},y_j+\beta_\alpha\Delta y_j}\right), \tag{3.25}$$

where β_α and w_α are Gaussian quadrature nodes and weights, for approximating the integration in y:

$$\frac{1}{\Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(u(x_{i+\frac{1}{2}},y,t)\,dy$$

inside the integral form of the PDE (3.22), and $u^\pm_{i+\frac{1}{2},y}$ are the k-th order accurate reconstructed values obtained by ENO or WENO reconstruction described in the previous section. As before, the superscripts $\pm$ imply the values are obtained within the cell I_{ij} (for

the superscript -) and the cell $I_{i+1,j}$ (for the superscript +), respectively. The flux $\hat{g}_{i,j+\frac{1}{2}}$ is defined similarly by

$$\hat{g}_{i,j+\frac{1}{2}} = \sum_{\alpha} w_\alpha h\left(u^-_{x_i+\beta_\alpha \Delta x_i, j+\frac{1}{2}}, u^+_{x_i+\beta_\alpha \Delta x_i, j+\frac{1}{2}}\right), \tag{3.26}$$

for approximating the integration in x:

$$\frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(u(x, y_{j+\frac{1}{2}}, t)\, dx$$

inside the integral form of the PDE (3.22). $u^{\pm}_{x,j+\frac{1}{2}}$ are again the k-th order accurate reconstructed values obtained by ENO or WENO reconstruction described in the previous section. h is again the one dimensional monotone flux, examples being given in (2.70)-(2.72).

We summarize the procedure to build a finite volume ENO or WENO 2D scheme (3.24), given the cell averages $\{\overline{u}_{ij}\}$ (we again drop the explicit reference to the time variable t), and a one dimensional monotone flux h, as follows:

Procedure 3.1. Finite volume 2D scalar ENO and WENO.

1. Follow the procedures described in Sect. 3.2, to obtain ENO or WENO reconstructed values at the Gaussian points, $u^{\pm}_{i+\frac{1}{2},y_j+\beta_\alpha \Delta y_j}$ and $u^{\pm}_{x_i+\beta_\alpha \Delta x_i, j+\frac{1}{2}}$. Notice that this step involves two one dimensional reconstructions, each one to remove a one dimensional cell average in one of the two directions. Also notice that the optimal weights used in the WENO reconstruction procedure are different for different Gaussian points indexed by α;

2. Compute the flux $\hat{f}_{i+\frac{1}{2},j}$ and $\hat{g}_{i,j+\frac{1}{2}}$ using (3.25) and (3.26);

3. Form the scheme (3.24).

□

We remark that the finite volume scheme in 2D, as described above, is very expensive due to the following reasons:

- A two dimensional reconstruction, at the cost of two one dimensional reconstruction per grid point, is needed. For general n space dimensions, the cost becomes n one dimensional reconstruction per grid point;

- More than one quadrature points are needed in formulating the flux (3.25)-(3.26), for order of accuracy higher than two. Thus, for ENO, although the stencil choosing process needs to be done only once, the reconstruction (2.10) has to be done for each quadrature point used in the flux formulation. For WENO, the optimal weights are also different for each quadrature point. This becomes much more costly for $n > 2$ dimension, as then the fluxes are defined by integrals in $n-1$ dimension and a $n-1$ dimensional quadrature rule must be used.

This is why multidimensional finite volume schemes of order of accuracy higher than 2 are rarely used. For 2D, based on [34], Casper [11] has coded up a fourth order finite volume ENO scheme for Cartesian grids, see also [12]. 3D finite volume ENO code of order of accuracy higher than 2 does not exist yet, to the author's knowledge.

At the second order level, the cost is greatly reduced because:

- There is no need to perform a reconstruction, as the cell average $\overline{u}_{ij}$ agrees with the point value at the center $u(x_i, y_j)$ to second order $O(\Delta^2)$;
- The quadrature rule in defining the flux (3.25)-(3.26) needs only one (mid) point.

One advantage of finite volume ENO or WENO schemes is that they can be defined on arbitrary meshes, provided that an ENO or WENO reconstruction on that mesh is available. See, for example, [1].

3.3.2 Finite difference formulation in the scalar case.

Here we assume a uniform grid and solve (3.21) directly using a conservative approximation to the spatial derivative:

$$\frac{du_{ij}(t)}{dt} = -\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}\right) - \frac{1}{\Delta y}\left(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}\right) \tag{3.27}$$

where $u_{ij}(t)$ is the numerical approximation to the point value $u(x_i, y_j, t)$.

The numerical flux $\hat{f}_{i+\frac{1}{2},j}$ is obtained by the one dimensional ENO or WENO approximation procedure, Procedure 2.4 or 2.5, with $v(x) = f(u(x, y_j, t))$ and with j fixed. Likewise, the numerical flux $\hat{g}_{i,j+\frac{1}{2}}$ is obtained by the one dimensional ENO or WENO approximation procedure, with $v(y) = f(u(x_i, y, t))$ and with i fixed.

All the one dimensional discussions in Sect. 2.3.2, such as upwinding, ENO-Roe, flux splitting, etc., can be applied here dimension by dimension.

The discussion here is also valid for higher spatial dimension n. In effect, it is the same one dimensional conservative derivative approximation applied to each space dimension.

It is a straight forward exercise [13] to show that, in terms of operation count, the finite difference ENO or WENO schemes are about a factor of 4 less than the finite volume counterpart of the same order. In 3D this factor becomes about 9.

We thus strongly recommend the usage of the finite difference version of ENO and WENO schemes (also called ENO and WENO schemes based on point values), whenever possible.

3.3.3 Provable properties in the scalar case.

Second order ENO schemes are also maximum norm non-increasing. Of course, this stability is too weak to imply any convergence. As was mentioned before, there is no known convergence result for ENO schemes of order higher than 2, even for smooth solutions.

WENO schemes have better convergence results also in the current multi-D case, mainly because their numerical fluxes are smoother. It is proven [43] that WENO schemes converge for smooth solutions.

We again emphasize that, even though there are very little theoretical results about ENO or WENO schemes, in practice they are very robust and stable. We once again caution against any attempts to modify the schemes solely for the purpose of stability or convergence proofs. In fact the modification of ENO schemes in [69], presented in Sect. 2.3.4, which keeps the formal uniform high order accuracy, actually produces schemes which are convergent to entropy solutions for general multi dimensional scalar equations. However it was pointed out there that the modification is not computationally useful, hence the convergence result has little value.

3.3.4 Systems.

The advice here is that, when the fluxes are computed along a cell boundary, a one dimensional local characteristic decomposition normal to the boundary is performed. Also, the monotone flux is replaced with a one dimensional exact or approximate Riemann solver. Thus, the discussion in Sect. 2.3.5 can be applied here.

There are discussions in the literature about truly multi-dimensional recipes. However, these tend to become extremely complicated for order of accuracy higher than two, so they have not been used extensively in practice. Another reason to suggest against using such complicated truly multidimensional recipes for order of accuracy higher than two is that, while dimension by dimension schemes as advocated in these lecture notes are not rotationally invariant, the direction related non-symmetry actually diminishes with increased order [13].

4 Further Topics

4.1 Further Topics in ENO and WENO Schemes

In this section we discuss some miscellaneous (but not necessarily unimportant!) topics in ENO and WENO schemes.

4.1.1 Subcell resolution.

This idea was first raised by Harten [35]. The observation is that, since in interpolating the primitive V, *two* points must be included in the initial stencil (see Procedure 2.1), one cannot avoid having at least one cell for each discontinuity, inside which the reconstructed polynomial is not accurate ($O(1)$ error there). We can clearly see this $O(1)$ error in the ENO interpolation in Figure 2.1. The reconstruction in this shocked cell, although inaccurate, will always be monotone (Property 2 in Sect. 2.2.1), so stability will not be a problem. However, it does cause a smearing of the discontinuity (over one cell, initially).

If we are solving a truly nonlinear shock, then characteristics flow into the shock, thus any error one makes during time evolution tends to be absorbed into the shock (we also say that the shock has a self sharpening mechanism). However, we are less lucky with a linear discontinuity, such as a discontinuity carried by the linear equation $u_t + u_x = 0$. Such linear discontinuities are also called contact discontinuities in gas dynamics. The characteristics for such cases are parallel to the discontinuity, hence any numerical smearing tends to accumulate and the discontinuity becomes progressively more smeared with time (Harten argues that

the smearing of the discontinuity is at the rate of $O(\Delta x^{1-\frac{1}{k+1}})$ where k is the order of the scheme. Although higher order schemes have less smearing, when time is large the smearing is still very significant.

Harten [35] makes the following simple observation: in the shocked cell I_i, instead of using the reconstruction polynomial $p_i(x)$, which is highly inaccurate (the only useful information it carries is the cell average in the cell), one could try to find the location of the discontinuity inside the cell I_i, say at x_s, and then use the neighboring reconstructions $p_{i-1}(x)$ extended to x_s from left and $p_{i+1}(x)$ extended to x_s from right. To find the shock location, one could argue that $p_{i-1}(x)$ is a very accurate approximation to $v(x)$ up to the discontinuity x_s from left, and $p_{i+1}(x)$ is a very accurate approximation to $v(x)$ up to the discontinuity x_s from right. We thus extend $p_{i-1}(x)$ from the left into the cell I_i, and extend $p_{i+1}(x)$ from the right into the cell I_i, and require that the cell average $\overline{v}_i$ be preserved:

$$\int_{x_{i-\frac{1}{2}}}^{x_s} p_{i-1}(x)\,dx + \int_{x_s}^{x_{i+\frac{1}{2}}} p_{i+1}(x)\,dx \;=\; \Delta x_i \overline{v}_i. \tag{4.1}$$

It can be proven that under very general conditions, (4.1) has only one root x_s inside the cell I_i, hence one could use Newton iterations to find this root.

Subcell resolution can be applied to both finite volume and finite difference ENO and WENO schemes [35], [70], However, it should be applied only to sharpen contact discontinuities. It is quite dangerous to apply the subcell resolution to a shock, since it might generate entropy violating expansion shocks in the numerical solution.

Another very serious restriction about subcell resolution is that it is very difficult to be applied to 2D. However, see Siddiqi, Kimia and Shu [73], where a geometrical ENO is used to extend the subcell resolution idea to 2D for image processing problems (we termed it geometric ENO, or GENO).

4.1.2 Artificial compression.

Another very useful idea to sharpen a contact discontinuity is the artificial compression, first developed by Harten [32] and further improved by Yang [83]. The idea is to *increase* the magnitude of the slope of a reconstruction, of course subject to certain monotonicity restrictions, near such a discontinuity. Notice that this goes against the idea of limiting, which typically *decreases* the magnitude of the slope of a reconstruction.

Artificial compression can be applied both to finite volume and to finite difference ENO and WENO schemes [83], [70], [43]. Unlike subcell resolution, artificial compression can also be applied easily to multi space dimensions, at least in principle.

4.1.3 Other building blocks.

It is not necessary to stay within polynomial building blocks, although polynomials are the most natural functions to work with. For some applications, other building blocks, such as rational functions, trigonometric polynomials, exponential functions, radial functions, etc., may be more appropriate. The idea of ENO or WENO can be applied also in such situations. The key idea is to find suitable "smooth indicators", similar to the Newton divided differences for the polynomial case, for applying the ENO or WENO idea. See [16] and [40] for some examples.

4.2 Time Discretization

Up to now we have only considered spatial discretizations, leaving the time variable continuous (method of lines). In this section we consider the issue of time discretization. The techniques discussed in this section can also be applied to other types of spatial discretizations using the method of lines approach, such as various TVD and TVB schemes [52, 78, 65] and discontinuous Galerkin methods [18, 19, 20, 21, 17].

4.2.1 TVD Runge-Kutta methods.

A class of TVD (total variation diminishing) high order Runge-Kutta methods is developed in [69] and further in [29].

These Runge-Kutta methods are used to solve a system of initial value problems of ODEs written as:

$$u_t = L(u), \tag{4.2}$$

resulting from a method of lines spatial approximation to a PDE such as:

$$u_t = -f(u)_x. \tag{4.3}$$

We have written the equation in (4.3) as a 1D conservation law, but the discussion which follows apply to general initial value problems of PDEs in any spatial dimensions. Clearly, $L(u)$ in (4.2) is an approximation (e.g. ENO or WENO approximation in these lecture notes), to the derivative $-f(u)_x$ in the PDE (4.3).

If we *assume* that a first order Euler forward time stepping:

$$u^{n+1} = u^n + \Delta t L(u^n) \tag{4.4}$$

is stable in a certain norm:

$$||u^{n+1}|| \leq ||u^n|| \tag{4.5}$$

under a suitable restriction on Δt:

$$\Delta t \leq \Delta t_1, \tag{4.6}$$

then we look for higher order in time Runge-Kutta methods such that the same stability result (4.5) holds, under a perhaps different restriction on Δt:

$$\Delta t \leq c\, \Delta t_1. \tag{4.7}$$

where c is termed *the CFL coefficient* for the high order time discretization.

We remark that the stability condition (4.5) for the first order Euler forward in time (4.4) is easy to obtain in many cases, such as various TVD and TVB schemes in 1D (where the norm is the total variation norm) and in multi dimensions (where the norm is the L^∞ norm), see, e.g. [52, 78, 65].

Originally in [69, 66] the norm in (4.5) was chosen to be the total variation norm, hence the terminology "TVD time discretization".

As it stands, the TVD high order time discretization defined above maintains stability in whatever norm, of the Euler forward first order time stepping, for the high order time discretization, under the time step restriction (4.7). For example, if it is used for multi

dimensional scalar conservation laws, for which TVD is not possible but maximum norm stability can be maintained for high order spatial discretizations plus forward Euler time stepping (e.g. [20]), then the same maximum norm stability can be maintained if TVD high order time discretization is used. As another example, if an entropy inequality can be proved for the Euler forward, then the same entropy inequality is valid under a high order TVD time discretization.

In [69], a general Runge-Kutta method for (4.2) is written in the form:

$$\begin{aligned} u^{(i)} &= \sum_{k=0}^{i-1} \left(\alpha_{ik} u^{(k)} + \Delta t \beta_{ik} L(u^{(k)}) \right), \qquad i = 1, ..., m \\ u^{(0)} &= u^n, \qquad u^{(m)} = u^{n+1}. \end{aligned} \tag{4.8}$$

Clearly, if all the coefficients are nonnegative $\alpha_{ik} \geq 0$, $\beta_{ik} \geq 0$, then (4.8) is just a convex combination of the Euler forward operators, with Δt replaced by $\frac{\beta_{ik}}{\alpha_{ik}} \Delta t$, since by consistency $\sum_{k=0}^{i-1} \alpha_{ik} = 1$. We thus have

Lemma 4.1. [69] The Runge-Kutta method (4.8) is TVD under the CFL coefficient (4.7):

$$c = \min_{i,k} \frac{\alpha_{ik}}{\beta_{ik}}, \tag{4.9}$$

provided that $\alpha_{ik} \geq 0$, $\beta_{ik} \geq 0$.

□

In [69], schemes up to third order were found to satisfy the conditions in Lemma 4.1 with CFL coefficient equal to 1.

The optimal second order TVD Runge-Kutta method is given by [69, 29]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n) \\ u^{n+1} &= \frac{1}{2} u^n + \frac{1}{2} u^{(1)} + \frac{1}{2} \Delta t L(u^{(1)}), \end{aligned} \tag{4.10}$$

with a CFL coefficient $c = 1$ in (4.9).

The optimal third order TVD Runge-Kutta method is given by [69, 29]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n) \\ u^{(2)} &= \frac{3}{4} u^n + \frac{1}{4} u^{(1)} + \frac{1}{4} \Delta t L(u^{(1)}) \\ u^{n+1} &= \frac{1}{3} u^n + \frac{2}{3} u^{(2)} + \frac{2}{3} \Delta t L(u^{(2)}), \end{aligned} \tag{4.11}$$

with a CFL coefficient $c = 1$ in (4.9).

Unfortunately, it is proven in [29] that no four stage, fourth order TVD Runge-Kutta method exists with nonnegative α_{ik} and β_{ik}. We thus have to consider the situation where $\alpha_{ik} \geq 0$ but β_{ik} might be negative. In such situations we need to introduce an adjoint operator $\tilde{L}$. The requirement for $\tilde{L}$ is that it approximates the same spatial derivative(s) as L, but is TVD (or stable in another relevant norm) for first order Euler, backward in time:

$$u^{n+1} = u^n - \Delta t \tilde{L}(u^n) \tag{4.12}$$

This can be achieved, for hyperbolic conservation laws, by solving the backward in time version of (4.3):

$$u_t = f(u)_x. \tag{4.13}$$

Numerically, the only difference is the change of upwind direction. Clearly, $\tilde{L}$ can be computed with the same cost as that of computing L. We then have the following lemma:

Lemma 4.2. [69] The Runge-Kutta method (4.8) is TVD under the CFL coefficient (4.7):

$$c = \min_{i,k} \frac{\alpha_{ik}}{|\beta_{ik}|}, \tag{4.14}$$

provided that $\alpha_{ik} \geq 0$, and L is replaced by $\tilde{L}$ for negative β_{ik}.

□

Notice that, if for the same k, both $L(u^{(k)})$ and $\tilde{L}(u^{(k)})$ must be computed, the cost as well as storage requirement for this k is doubled. For this reason, we would like to avoid negative β_{ik} as much as possible.

An extensive search performed in [29] gives the following preferred four stage, fourth order TVD Runge-Kutta method:

$$\begin{aligned}
u^{(1)} &= u^n + \frac{1}{2}\Delta t L(u^n) \\
u^{(2)} &= \frac{649}{1600}u^{(0)} - \frac{10890423}{25193600}\Delta t\tilde{L}(u^n) + \frac{951}{1600}u^{(1)} + \frac{5000}{7873}\Delta t L(u^{(1)}) \\
u^{(3)} &= \frac{53989}{2500000}u^n - \frac{102261}{5000000}\Delta t\tilde{L}(u^n) + \frac{4806213}{20000000}u^{(1)} \\
&\quad -\frac{5121}{20000}\Delta t\tilde{L}(u^{(1)}) + \frac{23619}{32000}u^{(2)} + \frac{7873}{10000}\Delta t L(u^{(2)}) \\
u^{n+1} &= \frac{1}{5}u^n + \frac{1}{10}\Delta t L(u^n) + \frac{6127}{30000}u^{(1)} + \frac{1}{6}\Delta t L(u^{(1)}) + \frac{7873}{30000}u^{(2)} \\
&\quad +\frac{1}{3}u^{(3)} + \frac{1}{6}\Delta t L(u^{(3)})
\end{aligned} \tag{4.15}$$

with a CFL coefficient $c = 0.936$ in (4.14). Notice that two $\tilde{L}$'s must be computed. The effective CFL coefficient, comparing with an ideal case without $\tilde{L}$'s, is $0.936 \times \frac{4}{6} = 0.624$. Since it is difficult to solve the global optimization problem, we do not claim that (4.15) is the optimal 4 stage, 4th order TVD Runge-Kutta method.

A fifth order TVD Runge-Kutta method is also given in [69].

For large scale scientific computing in three space dimensions, storage is usually a paramount consideration. There are therefore discussions about low storage Runge-Kutta methods [81], [10], which only require 2 storage units per ODE equation. In [29], we considered the TVD properties among such low storage Runge-Kutta methods and found third order low storage TVD Runge-Kutta methods.

The general low-storage Runge-Kutta schemes can be written in the form [81], [10]:

$$\begin{aligned}
du^{(i)} &= A_i du^{(i-1)} + \Delta t L(u^{(i-1)}) \\
u^{(i)} &= u^{(i-1)} + B_i du^{(i)}, \qquad i = 1, ..., m \\
u^{(0)} &= u^n, \;\; u^{(m)} = u^{n+1}, \;\; A_0 = 0
\end{aligned} \tag{4.16}$$

Only u and du must be stored, resulting in two storage units for each variable.

Carpenter and Kennedy [10] have classified all the three stage, third order (m=3) low storage Runge-Kutta methods, obtaining the following one parameter family:

$$\begin{aligned}
z_1 &= \sqrt{36c_2^4 + 36c_2^3 - 135c_2^2 + 84c_2 - 12} \\
z_2 &= 2c_2^2 + c_2 - 2 \\
z_3 &= 12c_2^4 - 18c_2^3 + 18c_2^2 - 11c_2 + 2 \\
z_4 &= 36c_2^4 - 36c_2^3 + 13c_2^2 - 8c_2 + 4 \\
z_5 &= 69c_2^3 - 62c_2^2 + 28c_2 - 8 \\
z_6 &= 34c_2^4 - 46c_2^3 + 34c_2^2 - 13c_2 + 2 \\
B_1 &= c_2 \\
B_2 &= \frac{12c_2(c_2-1)(3z_2 - z_1) - (3z_2 - z_1)^2}{144c_2(3c_2-2)(c_2-1)^2} \\
B_3 &= \frac{-24(3c_2-2)(c_2-1)^2}{(3z_2 - z_1)^2 - 12c_2(c_2-1)(3z_2 - z_1)} \\
A_2 &= \frac{-z_1(6c_2^2 - 4c_2 + 1) + 3z_3}{(2c_2+1)z_1 - 3(c_2+2)(2c_2-1)^2} \\
A_3 &= \frac{-z_4 z_1 + 108(2c_2-1)c_2^5 - 3(2c_2-1)z_5}{24z_1 c_2(c_2-1)^4 + 72c_2 z_6 + 72c_2^6(2c_2-13)}
\end{aligned} \tag{4.17}$$

In [29] we converted this form into the form (4.8), by introducing three new parameters. Then we searched for values of these parameters that would maximize the CFL restriction, by a computer program. The result seems to indicate that

$$c_2 = 0.924574 \tag{4.18}$$

gives an almost best choice, with CFL coefficient $c = 0.32$ in (4.9). This is of course less optimal than (4.11) in terms of CFL coefficients, however the low storage form is useful for large scale calculations.

We end this subsection by quoting the following numerical example [29], which shows that, even with a very nice second order TVD spatial discretization, if the time discretization is by a non-TVD but linearly stable Runge-Kutta method, the result may be oscillatory. Thus it would always be safer to use TVD Runge-Kutta methods for hyperbolic problems.

The numerical example uses the standard minmod based MUSCL second order spatial discretization [79]. We will compare the results of a TVD versus a non-TVD second order Runge-Kutta time discretizations. The PDE is the simple Burgers equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0 \tag{4.19}$$

with a Riemann initial data:

$$u(x,0) = \begin{cases} 1, & \text{if } x \le 0 \\ -0.5, & \text{if } x > 0 \end{cases} \tag{4.20}$$

The nonlinear flux $\left(\frac{1}{2}u^2\right)_x$ in (4.19) is approximated by the conservative difference

$$\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right),$$

where the numerical flux $\hat{f}_{i+\frac{1}{2}}$ is defined by

$$\hat{f}_{i+\frac{1}{2}} = h\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right)$$

with

$$\begin{aligned} u^-_{i+\frac{1}{2}} &= u_i + \frac{1}{2} minmod(u_{i+1} - u_i, u_i - u_{i-1}), \\ u^+_{i+\frac{1}{2}} &= u_{i+1} - \frac{1}{2} minmod(u_{i+2} - u_{i+1}, u_{i+1} - u_i) \end{aligned}$$

The monotone flux h is the Godunov flux defined by (2.70), and the *minmod* function is given by

$$minmod(a, b) = \frac{sign(a) + sign(b)}{2} \min(|a|, |b|).$$

It is easy to prove, by using Harten's Lemma [33], that the Euler forward time discretization with this second order MUSCL spatial operator is TVD under the CFL condition (4.6):

$$\Delta t \leq \frac{\Delta x}{2 \max_j |u^n_j|} \tag{4.21}$$

Thus $\Delta t = \frac{\Delta x}{2 \max_j |u^n_j|}$ will be used in all our calculations. Actually, apart from a slight difference (the *minmod* function is replaced by a minimum-in-absolute-value function), this MUSCL scheme is the same as the second order ENO scheme discussed in Sect. 2.3.1.

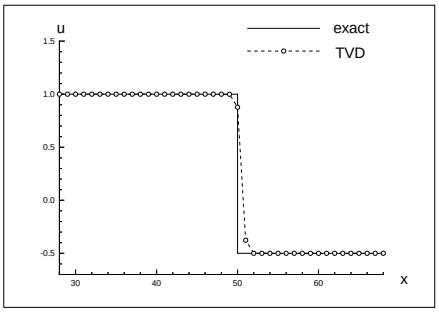
The TVD second order Runge-Kutta method we consider is the optimal one (4.10). The non-TVD method we use is:

$$\begin{aligned} u^{(1)} &= u^n - 20\Delta t L(u^n) \\ u^{n+1} &= u^n + \frac{41}{40}\Delta t L(u^n) - \frac{1}{40}\Delta t L(u^{(1)}). \end{aligned} \tag{4.22}$$

It is easy to verify that both methods are second order accurate in time. The second one (4.22) is however clearly non-TVD, since it has negative β's in both stages (i.e. it partially simulates backward in time with wrong upwinding).

If the operator L is linear (for example the first order upwind scheme applied to a linear PDE), then both Runge-Kutta methods (actually all the two stage, second order Runge-Kutta methods) yield identical results (the two stage, second order Runge-Kutta method for a linear ODE is unique). However, since our L is nonlinear, we may and do observe different results when the two Runge-Kutta methods are used.

In Fig. 4.1 we show the result of the TVD Runge-Kutta method (4.10) and the non-TVD method (4.22), after the shock moves about 50 grids (400 time steps for the TVD method,



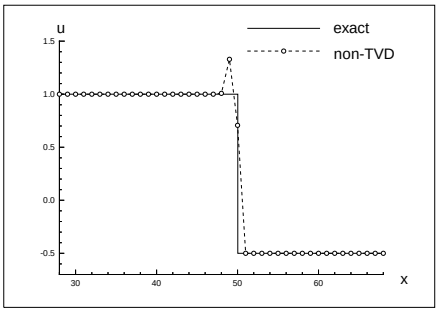


Figure 4.1: Second order TVD MUSCL spatial discretization. Solution after 500 time steps. Left: TVD time discretization (4.10); Right: non-TVD time discretization (4.22).

528 time steps for the non-TVD method). We can clearly see that the non-TVD result is oscillatory (there is an overshoot).

Such oscillations are also observed when the non-TVD Runge-Kutta method coupled with a second order TVD MUSCL spatial discretization is applied to a linear PDE ($u_t + u_x = 0$) (the scheme is still nonlinear due to the *minmod* functions). Moreover, for some Runge-Kutta methods, if one looks at the intermediate stages, i.e. $u^{(i)}$ for $1 \leq i < m$ in (4.8), one observes even bigger oscillations. Such oscillations may render difficulties when physical problems are solved, such as the appearance of negative density and pressure for Euler equations of gas dynamics. On the other hand, TVD Runge-Kutta method guarantees that each middle stage solution is also TVD.

This simple numerical test convinces us that it is much safer to use a TVD Runge-Kutta method for solving hyperbolic problems.

4.2.2 TVD multi-step methods.

If one prefers multi-step methods rather than Runge-Kutta methods, one can use the TVD high order multi-step methods developed in [66]. The philosophy is very similar to the TVD Runge-Kutta methods discussed in the previous subsection. One starts with a method of lines approximation (4.2) to the PDE (4.3), and an assumption that the first order Euler forward in time discretization (4.4) is stable under a certain norm (4.5), with the time step restriction (4.6). One then looks for higher order in time multi-step methods such that the same stability result (4.5) holds, under a perhaps different restriction on Δt in (4.7), where c is again termed *the CFL coefficient* for the high order time discretization.

The general form of the multi-step methods studied in [66] is:

$$u^{n+1} = \sum_{k=0}^{m} \left(\alpha_k u^{n-k} + \Delta t \beta_k L(u^{n-k}) \right), \tag{4.23}$$

Similar to the Runge-Kutta methods in the previous subsection, if all the coefficients are nonnegative $\alpha_k \geq 0$, $\beta_k \geq 0$, then (4.23) is just a convex combination of the Euler forward operators, with Δt replaced by $\frac{\beta_k}{\alpha_k}\Delta t$, since by consistency $\sum_{k=0}^{m} \alpha_k = 1$. We thus have

Lemma 4.3. [66] The multi-step method (4.23) is TVD under the CFL coefficient (4.7):

$$c = \min_k \frac{\alpha_k}{\beta_k}, \tag{4.24}$$

provided that $\alpha_k \geq 0$, $\beta_k \geq 0$.

□

In [66], schemes up to third order were found to satisfy the conditions in Lemma 4.3. Here we list a few examples.

The following three step ($m = 2$) scheme is second order and TVD

$$u^{n+1} = \frac{3}{4}u^n + \frac{3}{2}\Delta t L(u^n) + \frac{1}{4}u^{n-2} \tag{4.25}$$

with a CFL coefficient $c = 0.5$ in (4.24). This translates to the same efficiency as the optimal second order TVD Runge-Kutta scheme (4.10), as here only one residue evaluation is needed per time step. Of course, the storage requirement is bigger here. There is also the problem of the starting values u^1 and u^2.

The following five step ($m = 4$) scheme is third order and TVD

$$u^{n+1} = \frac{25}{32}u^n + \frac{25}{16}\Delta t L(u^n) + \frac{7}{32}u^{n-4} + \frac{5}{16}\Delta t L(u^{n-4}) \tag{4.26}$$

with a CFL coefficient $c = 0.5$ in (4.24). This translates to a better efficiency than the optimal third order TVD Runge-Kutta scheme (4.11), as here only one residue evaluation is needed per time step. Of course, the storage requirement is much bigger here. There is also the problem of the starting values u^1, u^2, u^3 and u^4.

There are many other TVD multi-step methods satisfying the conditions in Lemma 4.3 listed in [66]. It seems that if one uses more storage (larger m) one could get better CFL coefficients.

In [66] we have been unable to find multi-step schemes of order four or higher satisfying the condition of Lemma 4.3. As in the Runge-Kutta case, we can relax the condition $\beta_k \geq 0$ by introducing the adjoint operator $\tilde{L}$. We thus have

Lemma 4.4. [66] The multi-step method (4.23) is TVD under the CFL coefficient (4.7):

$$c = \min_k \frac{\alpha_k}{|\beta_k|}, \tag{4.27}$$

provided that $\alpha_k \geq 0$, and L is replaced by $\tilde{L}$ for negative β_k.

□

Again, notice that, if we have both positive and negative β_k's, then both $L(u^n)$ and $\tilde{L}(u^n)$ must be computed, the cost as well as storage requirement will thus be doubled.

We list here a six step ($m = 5$), fourth order multi-step method which is TVD with a CFL coefficient $c = 0.245$ in (4.24) [66]:

$$\begin{aligned} u^{n+1} &= \frac{747}{1280}u^n + \frac{237}{128}\Delta t L(u^n) \\ &\quad + \frac{81}{256}u^{n-4} + \frac{165}{128}\Delta t L(u^{n-4}) + \frac{1}{10}u^{n-5} - \frac{3}{8}\Delta t \tilde{L}(u^{n-5}) \end{aligned} \tag{4.28}$$

4.2.3 The Lax-Wendroff procedure.

Another way to discretize the time variable is by the Lax-Wendroff procedure [51]. This is also referred to as the Taylor series method for discretizing the ODE (4.2). We will again use the simple 1D scalar conservation law (4.3) as an example to illustrate the procedure, however it applies to more general multidimensional systems.

Starting from a Taylor series expansion in time:

$$u(x,t+\Delta t) = u(x,t) + u_t(x,t)\Delta t + u_{tt}(x,t)\frac{\Delta t^2}{2} + ... \tag{4.29}$$

The expansion is carried out to the desired order of accuracy in time. For example, a second order in time would need the three terms written out in (4.29). We then use the PDE (4.3) to replace the time derivatives by the spatial derivatives:

$$\begin{aligned} u_t(x,t) &= -f(u(x,t))_x = -f'(u(x,t))\,u_x(x,t); \\ u_{tt}(x,t) &= -(f(u(x,t))_{tx} = -(f'(u(x,t)\,u_t(x,t))_x \\ &= ((f'(u(x,t))^2 u_x(x,t))_x \\ &= 2f'(u(x,t))\,f''(u(x,t)\,(u_x(x,t))^2 + (f'(u(x,t)))^2\,u_{xx}(x,t); \end{aligned} \tag{4.30}$$

This little exercise in (4.30) should convince us that it is always possible to write all the time derivatives as functions of the $u(x,t)$ and its spatial derivatives. But the expression could be terribly complicated, especially for multidimensional systems.

Once this is done, we substitute (4.30) into (4.29), and then discretize the spatial derivatives of $u(x,t)$ by whatever methods we use. For example, in the cell averaged (finite volume) ENO schemes discussed in Sect. 2.3.1, we proceed as follows. We first integrate the PDE (2.65) in space-time over the region $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [t^n, t^{n+1}]$ to obtain

$$\overline{u}_i^{n+1} = \overline{u}_i^n - \frac{1}{\Delta x_i}\left(\int_{t^n}^{t^{n+1}} f(u(x_{i+\frac{1}{2}},t))dt \; - \int_{t^n}^{t^{n+1}} f(u(x_{i-\frac{1}{2}},t))dt\right) \tag{4.31}$$

Then, we use a suitable Gaussian quadrature to discretize the time integration for the flux in (4.31):

$$\frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{i+\frac{1}{2}},t))dt \approx \sum_\alpha w_\alpha f(u(x_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t), \tag{4.32}$$

where β_α and w_α are Gaussian quadrature nodes and weights. Next we replace each

$$f(u(x_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t)$$

by a monotone flux:

$$f(u(x_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t) \approx h\left(u(x^-_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t), u(x^+_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t)\right), \tag{4.33}$$

and use the Lax-Wendroff procedure (4.29)-(4.30) to convert

$$u(x^\pm_{i+\frac{1}{2}}, t^n + \beta_\alpha \Delta t)$$

to $u(x^{\pm}_{i+\frac{1}{2}}, t^n)$ and its spatial derivatives also at t^n, which can then be obtained by the reconstructions $p(x)$ inside I_i and I_{i+1}. Notice that the accuracy is just enough in this procedure, as each derivative of the reconstruction $p(x)$ will be one order lower in accuracy, but this is compensated by the Δt in front of it in (4.29).

This Lax-Wendroff procedure, comparing with the method of lines approach coupled with TVD Runge-Kutta or multi-step time discretizations, has the following advantages and disadvantages.

Advantages:

1. This is a truly one step method, hence it is quite compact (a second order method in space and time uses only three cells on time level n to advance to time level $n+1$ for one cell), and there are no complications such as boundary conditions needed in middle stages;

2. It utilizes the PDE more extensively than the method of lines approach. This is also one reason that it can be so compact.

Disadvantages:

1. The algebra is very, very complicated for multi dimensional systems. This also increases operation counts for complicated nonlinear systems;

2. It is more difficult to prove stability properties (e.g. TVD) for higher order methods in this framework;

3. It is difficult and costly to apply this procedure to the conservative finite difference framework established in Sections 2.3 and 3.3.

4.3 Formulation of the ENO and WENO Schemes for the Hamilton-Jacobi Equations

In this section we describe high order ENO and WENO approximations to the Hamilton-Jacobi equation:

$$\begin{cases} \phi_t + H(\phi_x, \phi_y) = 0 \\ \phi(x, y, 0) = \phi^0(x, y) \end{cases} \tag{4.34}$$

where H is a locally Lipschitz continuous Hamiltonian and the initial condition $\phi^0(x, y)$ is locally Lipschitz continuous. We have written the equation (4.34) in two space dimensions, but the discussion is valid for other space dimensions as well.

As is well known, solutions to (4.34) are Lipschitz continuous but may have discontinuous derivatives, regardless of the smoothness of $\phi^0(x, y)$. The non-uniqueness of such generalized solutions also necessitates the definition of viscosity solutions, to single out a unique, practically relevant solution. The viscosity solution to (4.34) is a locally Lipschitz continuous function $\phi(x, y, t)$, which satisfies the initial condition and the following property: for any smooth function $\psi(x, y, t)$, if (x_0, y_0, t_0) is a local maximum point of $\phi - \psi$, then

$$\psi_t(x_0, y_0, t_0) + H(\psi_x(x_0, y_0, t_0) + \psi_y(x_0, y_0, t_0)) \leq 0, \tag{4.35}$$

and, if (x_0, y_0, t_0) is a local minimum point of $\phi - \psi$, then

$$\psi_t(x_0, y_0, t_0) + H(\psi_x(x_0, y_0, t_0) + \psi_y(x_0, y_0, t_0)) \geq 0. \tag{4.36}$$

Of course, the above definition means that whenever $\phi(x, y, t)$ is differentiable, (4.34) is satisfied in the classical sense. Viscosity solution defined this way exists and is unique. For details and equivalent definitions of viscosity solutions, see Crandall and Lions [22].

Hamilton-Jacobi equations are actually easier to solve than conservation laws, because the solutions are typically continuous (only the derivatives are discontinuous).

As before, given mesh sizes Δx, Δy and Δt, we denote the mesh points as $(x_i, y_j, t_n) = (i\Delta x, j\Delta y, n\Delta t)$. The numerical approximation to the viscosity solution $\phi(x_i, y_j, t_n)$ of (4.34) at the mesh point (x_i, y_j, t_n) is denoted by ϕ_{ij}^n. We again use a semi-discrete (discrete in the spatial variables only) formulation as a middle step in designing algorithms. In such cases, the numerical approximation to the viscosity solution $\phi(x_i, y_j, t)$ of (4.34) at the mesh point (x_i, y_j, t) is denoted by $\phi_{ij}(t)$, the temporal variable t is not discretized. We will also use the notations $D_{\pm}^x \phi_{ij} = \frac{\pm(\phi_{i\pm1,j} - \phi_{ij})}{\Delta x}$ and $D_{\pm}^y \phi_{ij} = \frac{\pm(\phi_{i,j\pm1} - \phi_{ij})}{\Delta y}$ to denote the first order forward/backward difference approximations to the left and right derivatives of $\phi(x, y)$ at the location (x_i, y_j).

Since the viscosity solution to (4.34) is usually only Lipschitz continuous but not everywhere differentiable, the *formal* order of accuracy of a numerical scheme is again defined as that determined by the local truncation error in the smooth regions of the solution. Thus, a monotone scheme of the form

$$\phi_{ij}^{n+1} = G(\phi_{i-p,j-r}^n, \cdots, \phi_{i+q,j+s}^n) \tag{4.37}$$

where G is a non-decreasing function of each argument, is called a first order scheme, although the provable order of accuracy in the L_∞ norm is just $\frac{1}{2}$ [23]. In the semi-discrete formulation, a five point monotone scheme (it does not pay to use more points for a monotone scheme because the order of accuracy of a monotone scheme is at most one [36]) is of the form

$$\frac{d}{dt}\phi_{ij}(t) = -\hat{H}(D_+^x \phi_{ij}(t), D_-^x \phi_{ij}(t), D_+^y \phi_{ij}(t), D_-^y \phi_{ij}(t)). \tag{4.38}$$

The numerical Hamiltonian $\hat{H}$ is assumed to be locally Lipschitz continuous, consistent with H: $\hat{H}(u, u, v, v) = H(u, v)$, and is non-increasing in its first and third arguments and non-decreasing in the other two. Symbolically $\hat{H}(\downarrow, \uparrow, \downarrow, \uparrow)$. It is easy to see that, if the time derivative in (4.38) is discretized by Euler forward differencing, the resulting fully discrete scheme, in the form of (4.37), will be monotone when Δt is suitably small. We have chosen the semi-discrete formulation (4.38) in order to apply suitable nonlinearly stable high order Runge-Kutta type time discretization, see Sect. 4.2.

Semi-discrete or fully discrete monotone schemes (4.38) and (4.37) are both convergent towards the viscosity solution of (4.34) [23]. However, monotone schemes are at most first order accurate. As before, we will use the monotone schemes as building blocks for higher order ENO and WENO schemes.

ENO schemes were adapted to the Hamilton-Jacobi equations (4.34) by Osher and Sethian [59] and Osher and Shu [60]. As we know now, the key feature of the ENO algorithm is an adaptive stencil high order interpolation which tries to avoid shocks or high

gradient regions whenever possible. Since the Hamilton-Jacobi equation (4.34) is closely related to the conservation law (3.21), in fact in one space dimension they are exactly the same if one takes $u = \phi_x$, it is not surprising that successful numerical schemes for the conservation laws (3.21), such as ENO and WENO, can be applied to the Hamilton-Jacobi equation (4.34). ENO and WENO schemes, when applied to Hamilton-Jacobi equations (4.34), can produce high order accuracy in the smooth regions of the solution, and sharp, non-oscillatory corners (discontinuities in derivatives).

There are many monotone Hamiltonians [23], [59], [60]. In this section we mainly discuss the following two:

1. For the special case $H(u, v) = f(u^2, v^2)$ where f is a monotone function of both arguments, such as the example $H(u, v) = \sqrt{u^2 + v^2}$, we can use the Osher-Sethian monotone Hamiltonian [59]:

$$\hat{H}^{OS}(u^+, u^-, v^+, v^-) = f(u^2, v^2) \tag{4.39}$$

where, if f is a non-increasing function of u^2, u^2 is implemented by

$$u^2 = (\min(u^-, 0))^2 + (\max(u^+, 0))^2 \tag{4.40}$$

and, if f is a non-decreasing function of u^2, u^2 is implemented by

$$u^2 = (\min(u^+, 0))^2 + (\max(u^-, 0))^2 \tag{4.41}$$

Similarly for v^2. This Hamiltonian is purely upwind (i.e. when $H(u, v)$ is monotone in u in the relevant domain $[u^-, u^+] \times [v^-, v^+]$, only u^- or u^+ is used in the numerical Hamiltonian according to the wind direction), and simple to program. Whenever applicable it should be used. This flux is similar to the Engquist-Osher monotone flux (2.71) for the conservation laws.

2. For the general H we can always use the Godunov type Hamiltonian [5], [60]:

$$\hat{H}^G(u^+, u^-, v^+, v^-) = ext_{u \in I(u^-, u^+)} \; ext_{v \in I(v^-, v^+)} \; H(u, v) \tag{4.42}$$

where the extrema are defined by

$$ext_{u \in I(a,b)} = \begin{cases} \min_{a \le u \le b} & \text{if } a \le b \\ \max_{b \le u \le a} & \text{if } a > b \end{cases} \tag{4.43}$$

Godunov Hamiltonian is obtained by attempting to solve the Riemann problem of the equation (4.34) exactly with piecewise linear initial condition determined by $u^\pm$ and $v^\pm$. It is in general not unique, because in general $\min_u \max_v H(u, v) \neq \max_v \min_u H(u, v)$ and interchanging the order of the two ext's in (4.42) can produce a different monotone Hamiltonian.

Godunov Hamiltonian is purely upwind and is the least dissipative among all monotone Hamiltonians [57]. However, it might be extremely difficult to program, since in general analytical expressions for things like $\min_u \max_v H(u, v)$ can be quite complicated. The readers will be convinced by doing the exercise of obtaining the analytical expression and programming H^G for the ellipse in ellipse case in image processing where $H(u, v) = \sqrt{au^2 + 2buv + cv^2}$. For this case the Osher-Sethian Hamiltonian H^{OS} does not apply.

We are now ready to discuss about higher order ENO or WENO schemes for (4.34). The framework is quite simple: we simply replace the first order scheme (4.38) by:

$$\frac{d}{dt}\phi_{ij}(t) = -\hat{H}(u_{ij}^+(t), u_{ij}^-(t), v_{ij}^+(t), v_{ij}^-(t)) \tag{4.44}$$

where $u_{ij}^{\pm}(t)$ are high order approximations to the left and right x-derivatives of $\phi(x, y, t)$ at (x_i, y_j, t):

$$u_{ij}^{\pm}(t) = \frac{\partial \phi}{\partial x}(x_i^{\pm}, y_j, t) + O(\Delta x^r) \tag{4.45}$$

Similarly for $v_{ij}^{\pm}(t)$. Notice that there is no cell-averaged version now.

The key feature of ENO to avoid numerical oscillations is through the following interpolation procedure to obtain $u_{ij}^{\pm}(t)$ and $v_{ij}^{\pm}(t)$. These are just the same ENO procedure we discussed before in Sect. 2.2. We repeat it here with its own notations:

ENO Interpolation Algorithm: Given point values $f(x_j)$, $j = 0, \pm 1, \pm 2, \cdots$ of a (usually piecewise smooth) function $f(x)$ at discrete nodes x_j, we associate an r-th degree polynomial $P_{j+1/2}^{f,r}(x)$ with each interval $[x_j, x_{j+1}]$, with the left-most point in the stencil as $x_{k_{min}^{(r)}}$, constructed inductively as follows:

(1) $P_{j+1/2}^{f,1}(x) = f[x_j] + f[x_j, x_{j+1}](x - x_j)$, $k_{min}^{(1)} = j$;

(2) If $k_{min}^{(l-1)}$ and $P_{j+1/2}^{f,l-1}(x)$ are both defined, then let

$$\begin{aligned} a^{(l)} &= f[x_{k_{min}^{(l-1)}}, \cdots, x_{k_{min}^{(l-1)}+l}] \\ b^{(l)} &= f[x_{k_{min}^{(l-1)}-1}, \cdots, x_{k_{min}^{(l-1)}+l-1}] \end{aligned}$$

and

(i) If $|a^{(l)}| \geq |b^l|$, then $c^{(l)} = b^{(l)}$ and $k_{min}^{(l)} = k_{min}^{(l-1)} - 1$; otherwise $c^{(l)} = a^{(l)}$ and $k_{min}^{(l)} = k_{min}^{(l-1)}$;

(ii) $P_{j+1/2}^{f,l}(x) = P_{j+1/2}^{f,l-1}(x) + c^{(l)} \prod_{i=k_{min}^{(l-1)}}^{k_{min}^{(l-1)}+l-1}(x - x_i)$.

□

In the above procedure $f[\cdot, \cdots, \cdot]$ are the standard Newton divided differences, inductively defined as $f[x_1, x_2, \cdots, x_{k+1}] = \frac{f[x_2, \cdots, x_{k+1}] - f[x_1, \cdots, x_k]}{x_{k+1} - x_1}$ with $f[x_1] = f(x_1)$.

ENO Interpolation Algorithm starts with a first degree polynomial $P_{j+1/2}^{f,1}(x)$ interpolating the function $f(x)$ at the two grid points x_j and x_{j+1}. If we stop here, we would obtain the first order monotone scheme. When higher order is desired, we will in each step add just one point to the existing stencil, chosen from the two immediate neighbors by the size of the two relevant divided differences, which measures the local smoothness of the function $f(x)$.

The approximations to the left and right x-derivatives of ϕ are then taken as

$$u_{ij}^{\pm} = \frac{\partial}{\partial x} P_{i\pm 1/2,j}^{\phi,r}(x_i). \tag{4.46}$$

where $P_{i\pm 1/2,j}^{\phi,r}(x)$ is obtained by the ENO Interpolation Algorithm in the x-direction, with $y = y_j$ and t both fixed. $v_{ij}^{\pm}$ are obtained in a similar fashion. The resulting ODE (4.44)

is then discretized by an r-th order TVD Runge-Kutta time discretization in Sect. 4.2 to guarantee nonlinear stability. More specifically, the high order Runge-Kutta method we use in Sect. 4.2 will maintain TVD (total-variation-diminishing) or other stability properties, if these properties are valid for the simple first order Euler forward time discretization of the ODE (4.44). Notice that this is different from the usual linear stability requirement for the ODE solver. We thus obtain both nonlinear stability and high order accuracy in time. The second order ($r = 2$) and third order ($r = 3$) methods we use which has this stability property are given by (4.10) and (4.11), respectively.

Time step restriction is taken as

$$\Delta t \left(\frac{1}{\Delta x} \max_{u,v} \left| \frac{\partial}{\partial u} H(u,v) \right| + \frac{1}{\Delta y} \max_{u,v} \left| \frac{\partial}{\partial v} H(u,v) \right| \right) \leq 0.6 \tag{4.47}$$

where the maximum is taken over the relevant ranges of u, v. Here 0.6 is just a convenient number used in practice. This number should be chosen between 0.5 and 0.7 according to our numerical experience.

WENO schemes can be used in a similar fashion for Hamilton-Jacobi equations [45]. We will not present the details here.

5 Applications

5.1 Applications to Compressible Gas Dynamics

One of the main application areas of ENO and WENO schemes is compressible gas dynamics.

In 3D, Euler equations of gas dynamics are written as

$$U_t + f(U)_x + g(U)_y + h(U)_z = 0$$

where

$$\begin{aligned} U &= (\rho, \rho u, \rho v, \rho w, E), \\ f(U) &= (\rho u, \rho u^2 + P, \rho uv, \rho uw, u(E+P)), \\ g(U) &= (\rho v, \rho uv, \rho v^2 + P, \rho vw, v(E+P)), \\ h(U) &= (\rho w, \rho uw, \rho vw, \rho w^2 + P, w(E+P)). \end{aligned}$$

Here ρ is density, (u, v, w) is the velocity, E is the total energy, P is the pressure, related to the total energy E by

$$E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2)$$

with $\gamma = 1.4$ for air.

For the form of Navier-Stokes equations, for the eigenvalues and eigenvectors needed for the characteristic-wise ENO and WENO schemes, and for those equations appearing in curvilinear coordinates, see, e.g. [71].

We mention the following applications of ENO and WENO schemes for compressible flow calculations:

1. Shock tube problem. This is a standard problem for testing codes for shock calculations. However, it is not the best test case for high order methods, as the solution structure is relatively simple (basically piecewise linear). The set-up is a Riemann type initial data:

$$U(x,0) = \begin{cases} U_L & \text{if } x \le 0 \\ U_R & \text{if } x > 0 \end{cases}$$

The two standard test cases are the Sod's problem:

$$(\rho_L, q_L, P_L) = (1, 0, 1); \quad (\rho_R, q_R, P_R) = (0.125, 0, 0.1)$$

and the Lax's problem:

$$(\rho_L, q_L, P_L) = (0.445, 0.698, 3.528); \quad (\rho_R, q_R, P_R) = (0.5, 0, 0.571)\,.$$

We show the results of WENO (third order and fifth order) schemes for the Lax problem, in Fig. 5.1. Notice that "PS" in the pictures means a way of treating the system cheaper than the local characteristic decompositions (for details, see [43]). "A" stands for Yang's artificial compression [83] applied to these cases [43].

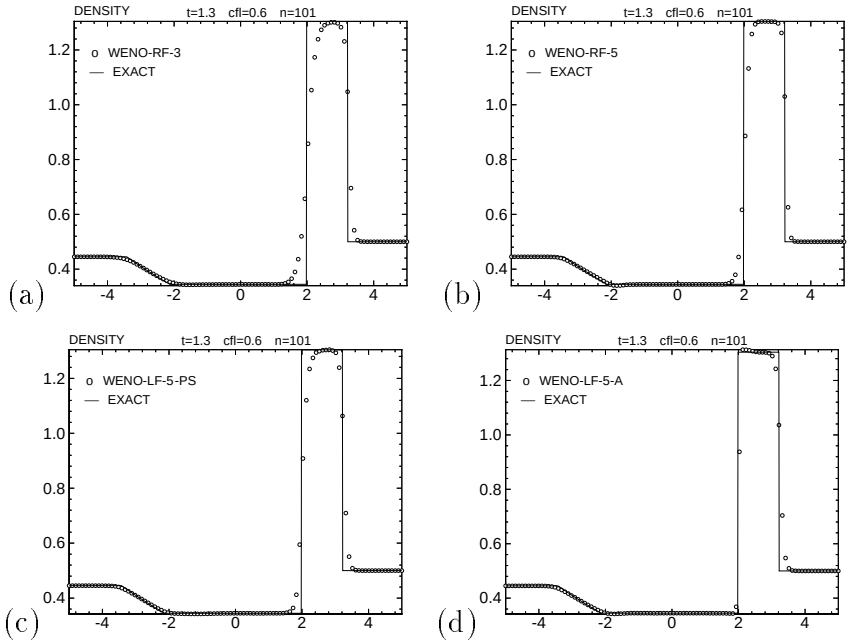


Figure 5.1: Shock tube, Lax problem, density. (a): third order WENO; (b): fifth order WENO; (c): fifth order WENO with cheaper characteristic decomposition; (d): fifth order WENO with artificial compression.

We can see from Fig. 5.1 that WENO perform reasonably well for these shock tube problem. The contact discontinuity is smeared more than the shock, as expected. Artificial compression helps sharpening contacts. For this problem, which is not the most demanding, the less expensive "PS" version of WENO work quite well.

ENO schemes on this test case perform similarly. We will not give the pictures here. See [70].

2. Shock entropy wave interactions. This problem is very suitable for high order ENO and WENO schemes, because both shocks and complicated smooth flow feature co-exist. In this example, a moving shock interacting with an entropy wave of small amplitude. On a domain $[0, 5]$, the initial condition is:

$$\begin{array}{llll} \rho = 3.85714; & u = 2.629369; & P = 10.33333; & \text{when} \quad x < 0.5 \\ \rho = e^{-\epsilon \sin(kx)}; & u = 0; & P = 1; & \text{when} \quad x \geq 0.5 \end{array}$$

where ϵ and k are the amplitude and wave number of the entropy wave, respectively. The mean flow is a pure right moving Mach 3 shock. If ϵ is small compared to the shock strength, the shock will march to the right at approximately the non-perturbed shock speed and generate a sound wave which travels along with the flow behind the shock. At the same time, the perturbing entropy wave, after "going through" the shock, is compressed and amplified and travels approximately at the speed of $u + c$ where u and c are the velocity and speed of the sound of the mean flow left to the shock. The amplification factor for the entropy wave can be obtained by linear analysis.

Since the entropy wave here is set to be very weak relative to the shock, any numerical oscillation might pollute the generated waves (e.g. the sound waves) and the amplified entropy waves. In our tests, we take $\epsilon = 0.01$ and $k = 13$. The amplitude of the amplified entropy waves predicted by the linear analysis is 0.08690716 (shown in the following figures as horizontal solid lines).

In Fig. 5.2, we show the result (entropy) when 12 waves have passed through the shock. It is clear that a lower order method (more dissipative) will damp the magnitude of the transmitted wave more seriously, especially when the waves are traveling more and more away from the shock. We can see that, while fifth order WENO with 800 points already resolves the passing waves well, and with 1200 points resolves the waves excellently, a second order TVD scheme (which is a good one among second order schemes) with 2000 points still shows excessive dissipation downstream. If we agree that fifth order WENO with 800 points behaves similarly as second order TVD with 2000 points, then there is a saving of a factor of 2.5 in grid points. This factor is *per dimension*, hence for a 3D time dependent problem the saving of the number of space-time grids will be a factor of $2.5^4 \approx 40$, a significant saving even after factoring in the extra cost per grid point for the higher order WENO method.

ENO schemes behave similarly for this problem.

There is a two dimensional version of this problem, when the entropy wave can make an angle with the shock. The simulation results again show an advantage in using a higher order method, in Fig. 5.3. Several curves are clustered in Fig. 5.3 around the exact solution, belonging to various fourth and fifth order ENO or WENO schemes. The circles correspond to a second order TVD scheme, which dissipates the amplitude of the transmitted entropy wave much more rapidly.

3. Steady state calculations. This is important both in gas dynamics and in other fields of applications, such as in semiconductor device simulation, Sect. 5.3. For ENO or TVD

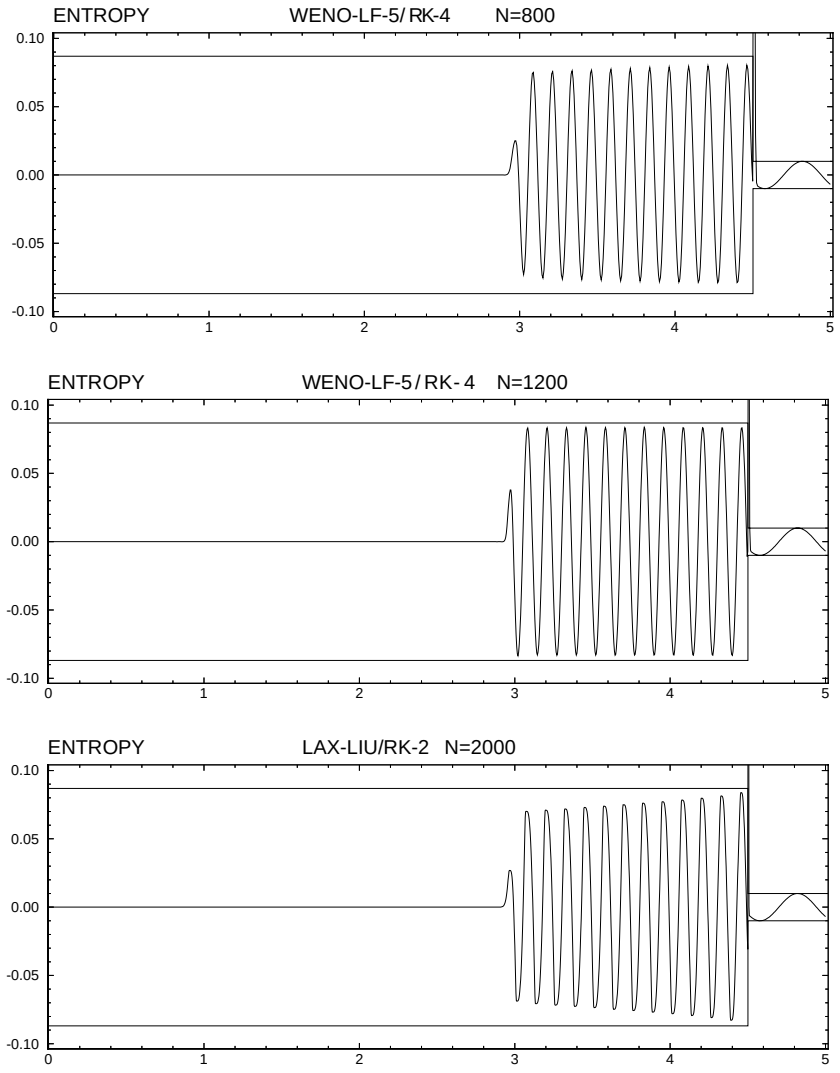


Figure 5.2: 1D shock entropy wave interaction. Entropy. Top: fifth order WENO with 800 points; middle: fifth order WENO with 1200 points; bottom: second order TVD with 2000 points.

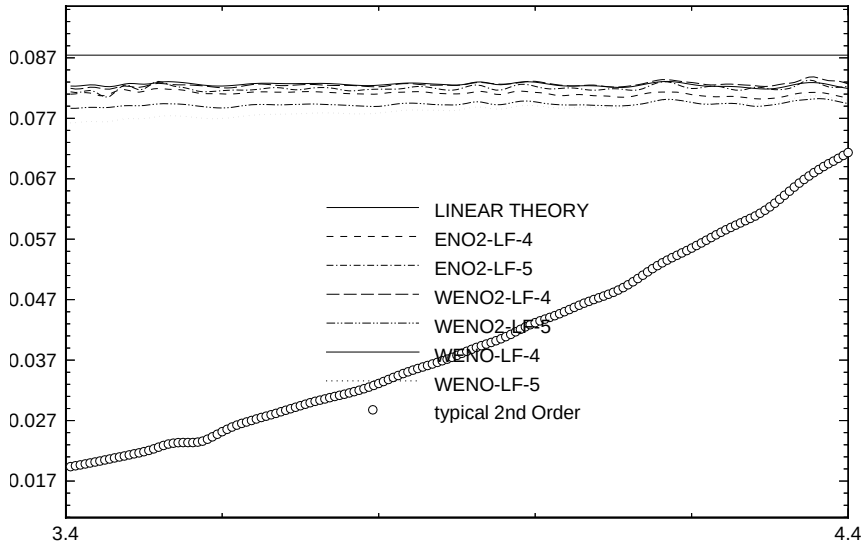


Figure 5.3: 2D shock entropy wave interaction. Amplitude of amplified entropy waves. 800 points (about 20 points per entropy wave length).

schemes, the residue does not settle down to machine zero during the time evolution. It will decay first and then hang at the level of the local truncation errors. Presumably this is due to the fact that the numerical flux is not smooth enough (it is only Lipschitz continuous but not C^1). Although this is not satisfactory, it does not seem to affect the final solution (up to the truncation error level, which is how accurate the solution will be anyway).

WENO schemes are much better in getting the residues to settle down to machine zeroes, due to the smoothness of their fluxes.

In Fig. 5.4 we show the result of a one dimensional nozzle calculation. The residue in this case settles down nicely to machine zeros. Both fourth and fifth order WENO results are shown.

4. Forward facing step problem. This is a standard test case for high resolution schemes [82]. However, second order methods usually already work well. High order methods might have some advantage in resolving the slip lines.

   The set up of the problem is the following: the wind tunnel is 1 length unit wide and 3 length units long. The step is 0.2 length units high and is located 0.6 length units from the left-hand end of the tunnel. The problem is initialized by a right-going Mach 3 flow. Reflective boundary conditions are applied along the walls of the tunnel and in-flow and out-flow boundary conditions are applied at the entrance (left-hand end) and the exit (right-hand end). For the treatment of the singularity at the corner of the step, we adopt the same technique used in [82], which is based on the assumption of a nearly steady flow in the region near the corner.

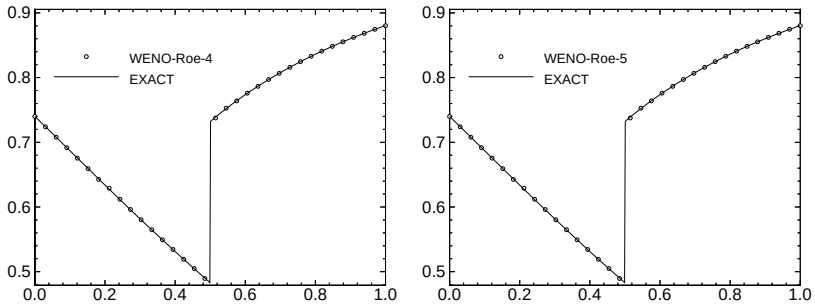


Figure 5.4: Density. Steady quasi-1D nozzle flow. 34 points. Left: fourth order WENO; right: fifth order WENO.

In Fig. 5.5 we present the results of fifth order WENO and fourth order ENO with 242×79 grid points.

5. Double Mach reflection. This is again a standard test case for high resolution schemes [82]. However, second order methods usually again already work well. High order methods might have some advantage in resolving the flow below the Mach stem.

   The computational domain for this problem is chosen to be $[0,4] \times [0,1]$, although only part of it, $[0,3] \times [0,1]$, is shown [82]. The reflecting wall lies at the bottom of the computational domain starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = \frac{1}{6}$ and a reflective boundary condition is used for the rest. At the top boundary of our computational domain, the flow values are set to describe the exact motion of the Mach 10 shock. See [82] for a detailed description of this problem.

   In Fig. 5.6 we present the results of fifth order WENO and fourth order ENO with 480×119 grid points.

6. 2D shock vortex interactions. High order methods have some advantages in this case, as it resolves the vortex and the interaction better.

   The model problem we use describes the interaction between a stationary shock and a vortex. The computational domain is taken to be $[0,2] \times [0,1]$. A stationary Mach 1.1 shock is positioned at $x = 0.5$ and normal to the x-axis. Its left state is $(\rho, u, v, P) = (1, \sqrt{\gamma}, 0, 1)$. A small vortex is superposed to the flow left to the shock and centers at $(x_c, y_c) = (0.25, 0.5)$. We describe the vortex as a perturbation to the velocity (u, v), temperature $(T = \frac{P}{\rho})$ and entropy $(S = \ln \frac{P}{\rho^\gamma})$ of the mean flow and denote it by the tilde values:

$$\tilde{u} = \epsilon \tau e^{\alpha(1-\tau^2)} \sin\theta \tag{5.1}$$

$$\tilde{v} = -\epsilon \tau e^{\alpha(1-\tau^2)} \cos\theta \tag{5.2}$$

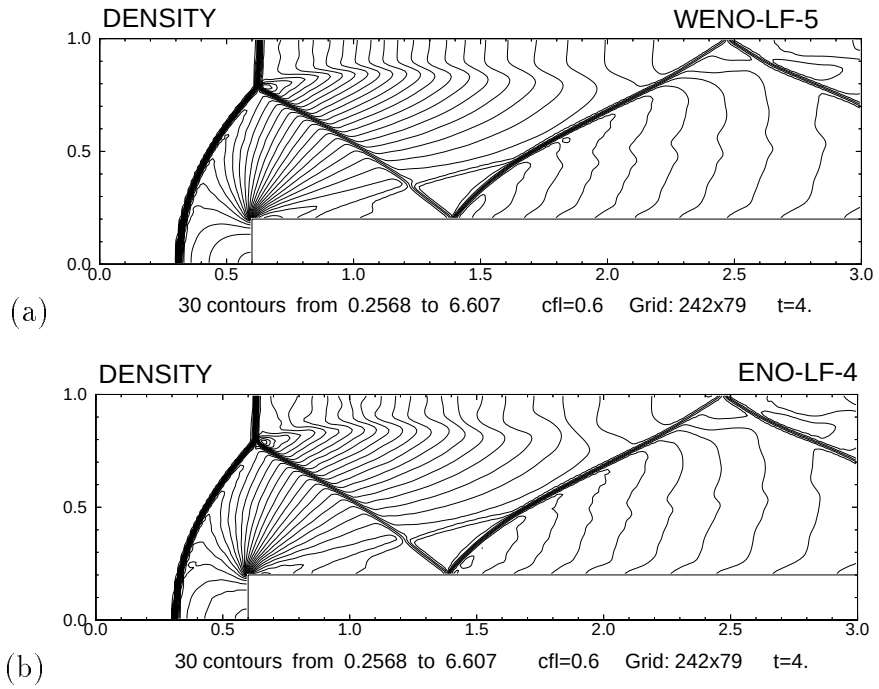


Figure 5.5: Flow past a forward facing step. Density: 242×79 grid points. Top: fifth order WENO; bottom: fourth order ENO.

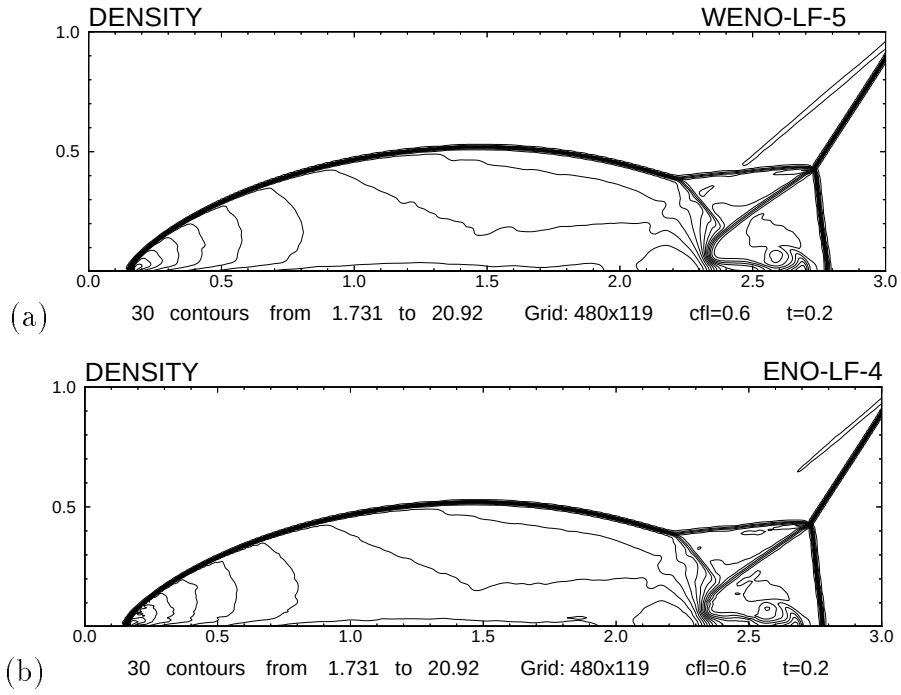


Figure 5.6: Double Mach reflection. Density: 480×119 grid points. Top: fifth order WENO; bottom: fourth order ENO.

$$\tilde{T} = -\frac{(\gamma-1)\epsilon^2 e^{2\alpha(1-\tau^2)}}{4\alpha\gamma} \tag{5.3}$$

$$\tilde{S} = 0 \tag{5.4}$$

where $\tau = \frac{r}{r_c}$ and $r = \sqrt{(x-x_c)^2+(y-y_c)^2}$. Here ϵ indicates the strength of the vortex, α controls the decay rate of the vortex and r_c is the critical radius for which the vortex has the maximum strength. In our tests, we choose $\epsilon = 0.3, r_c = 0.05$ and $\alpha = 0.204$. The above defined vortex is a steady state solution to the 2D Euler equation.

We use a grid of 251×100 which is uniform in y but refined in x around the shock. The upper and lower boundaries are intentionally set to be reflective. The results (pressure contours) are shown in Fig. 5.7 for a fifth order WENO with the cheap "PS" way of treating characteristic decomposition for the system.

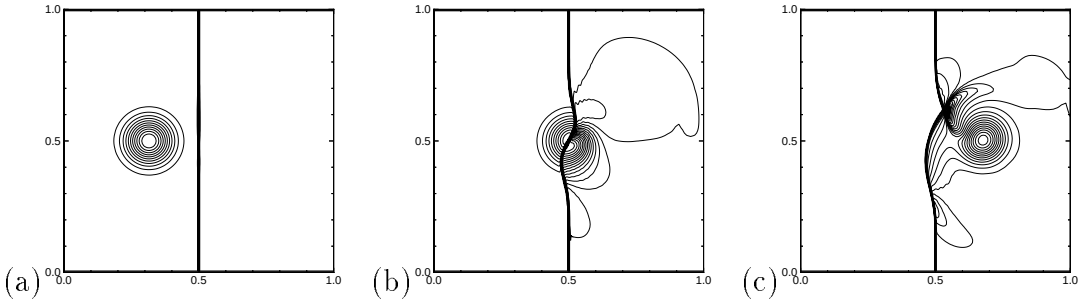


Figure 5.7: 2D shock vortex interaction. Pressure. Fifth order WENO-LF-5-PS. 30 contours. (a) t=0.05. (b) t=0.20. (c) t=0.35.

In [27], interaction of a shock with a longitudinal vortex is also investigated by the ENO method.

7. How does the finite difference version of ENO and WENO handle non-rectangular domain? As we mentioned before, as long as the domain can be *smoothly* transformed to a rectangle, the schemes can be handily applied.

   We consider, as an example, the problem of a supersonic flow past a cylinder. In the physical space, a cylinder of unit radius is positioned at the origin on a $x-y$ plane. The computational domain is chosen to be $[0,1]\times[0,1]$ on $\xi-\eta$ plane. The mapping between the computational domain and the physical domain is:

$$x = (R_x - (R_x-1)\xi)\cos(\theta(2\eta-1)) \tag{5.5}$$

$$y = (R_y - (R_y-1)\xi)\sin(\theta(2\eta-1)) \tag{5.6}$$

   where we take $R_x = 3, R_y = 6$ and $\theta = \frac{5\pi}{12}$. Fifth order WENO and a uniform mesh of 60×80 in the computational domain are used.

   The problem is initialized by a Mach 3 shock moving toward the cylinder from the left. Reflective boundary condition is imposed at the surface of the cylinder, i.e. $\xi = 1$, inflow

boundary condition is applied at $\xi = 0$ and outflow boundary condition is applied at $\eta = 0, 1$,

We present an illustration of the mesh in the physical space (drawing every other grid line), and the pressure contour, in Fig. 5.8. Similar results are obtained by the ENO schemes but are not shown here.

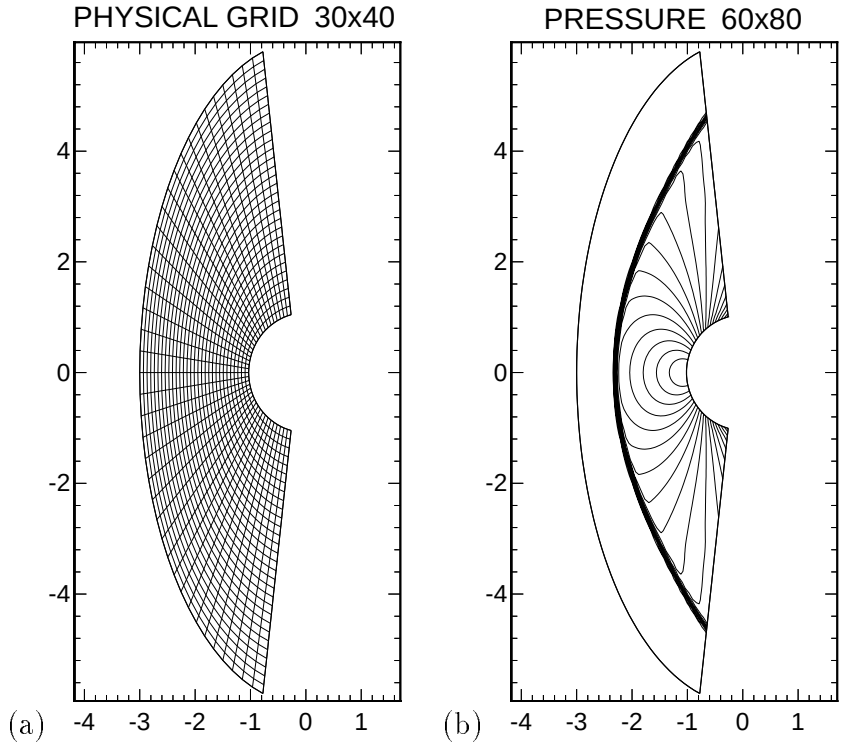


Figure 5.8: Flow past a cylinder. (a) Physical grid. (b) Pressure. WENO-LF-5. 20 contours

8. Finally, we use the following problem to illustrate more clearly the power of high order methods. Consider the following idealized problem for the Euler equations in 2D: The mean flow is $\rho = 1$, $P = 1$, and $(u, v) = (1, 1)$ (diagonal flow). We add, to this mean flow, an isentropic vortex (perturbations in (u, v) and the temperature $T = \frac{P}{\rho}$, no perturbation in the entropy $S = \frac{P}{\rho^\gamma}$):

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(-\overline{y}, \overline{x})$$

$$\delta T = -\frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \qquad \delta S = 0,$$

where $(\overline{x}, \overline{y}) = (x - 5, y - 5)$, $r^2 = \overline{x}^2 + \overline{y}^2$, and the vortex strength $\epsilon = 5$.

Since the mean flow is in the diagonal direction, the vortex movement is not aligned with the mesh direction.

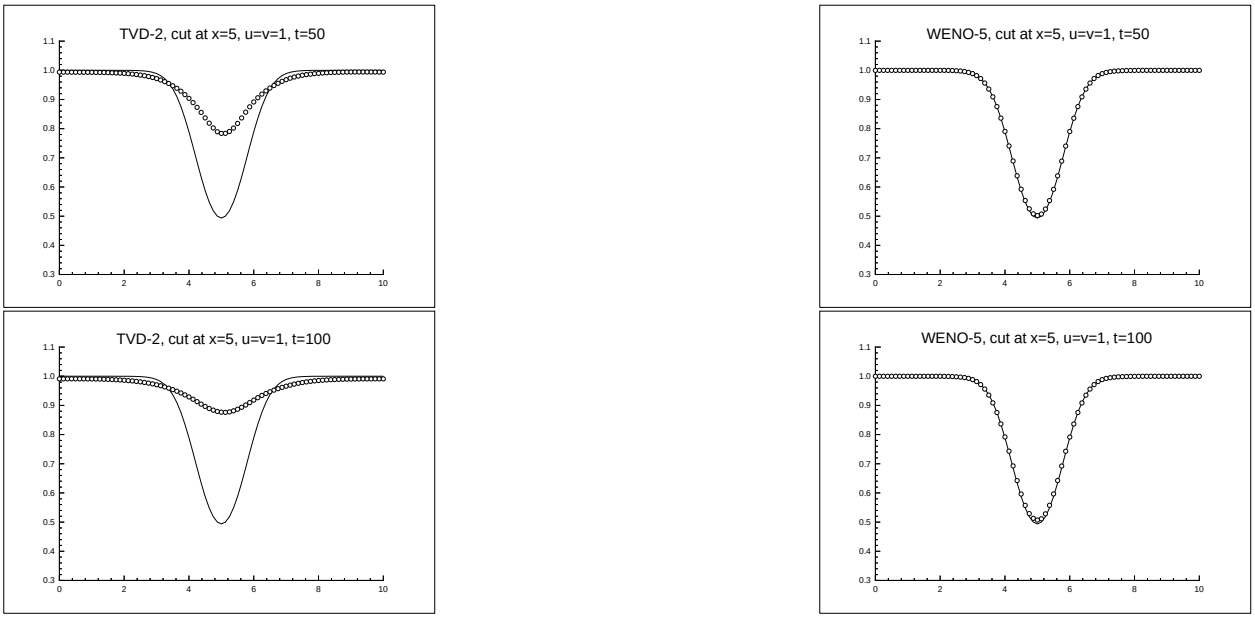


Figure 5.9: Vortex evolution. Cut at $x = 5$. Solid: exact solution; circles: computed solution. Top: $t = 50$ (after 5 time periods); bottom: $t = 100$ (after 10 time periods). Left: second order TVD scheme; right: fifth order WENO scheme.

The computational domain is taken as [0,10]×[0,10], *extended periodically in both directions*. This allows us to perform long time simulation without having to deal with a large domain. As we will see, the advantage of the high order methods are more obvious for long time simulations.

It is clear that the exact solution of the Euler equation with the above initial and boundary conditions is just the passive convection of the vortex with the mean velocity.

A grid of 80^2 points is used. The simulation is performed until $t = 100$ (10 periods in time). As can be seen from Fig. 5.9, fifth order WENO has a much better resolution than a second order TVD scheme, especially for the larger time $t = 100$.

5.2 Applications to Incompressible Flows

In this section we consider numerically solving the incompressible Navier-Stokes or Euler equations

$$\begin{aligned} u_t + uu_x + vu_y &= \mu(u_{xx} + u_{yy}) - p_x \\ v_t + uv_x + vv_y &= \mu(v_{xx} + v_{yy}) - p_y \\ u_x + v_y &= 0 \end{aligned} \tag{5.7}$$

or their equivalent conservative form

$$\begin{aligned} u_t + (u^2)_x + (uv)_y &= \mu(u_{xx} + u_{yy}) - p_x \\ v_t + (uv)_x + (v^2)_y &= \mu(v_{xx} + v_{yy}) - p_y \\ u_x + v_y &= 0 \end{aligned} \tag{5.8}$$

where (u, v) is the velocity vector, p is the pressure, $\mu > 0$ for the Navier-Stokes equations and $\mu = 0$ for the Euler equations, using ENO and WENO schemes. We do not discuss the issue of boundary conditions here, thus the equation is defined on the box $[0, 2\pi] \times [0, 2\pi]$ with periodic boundary conditions in both directions. We choose two space dimensions for easy presentation, although our method is also applicable for three space dimensions.

In some sense equations (5.7) are easier to solve numerically than their compressible counter-parts in Sect. 5.1, because the latter have solutions containing possible discontinuities (for example shocks and contact discontinuities). However, the solution to (5.7), even if for most cases smooth mathematically, may evolve rather rapidly with time t and may easily become too complicated to be fully resolved on a feasible grid. Traditional linearly stable schemes, such as spectral methods and high-order central difference methods, are suitable for the cases where the solution can be fully resolved, but typically produce signs of instability such as oscillations when small scale features of the flow, such as shears and roll-ups, cannot be adequately resolved on the computational grid. Although in principle one can always overcome this difficulty by refining the grid, today's computer capacity seriously restricts the largest possible grid size.

As we know, the high resolution "shock capturing" schemes such as ENO and WENO are based on the philosophy of giving up fully resolving rapid transition regions or shocks, just to "capture" them in a stable and somehow globally correct fashion (e.g., with correct shock speed), but at the same time to require a high resolution for the smooth part of the solution. The success of such an approach for the conservation laws is documented by many examples in these lecture notes and the references. One example is the one and two dimensional shock interaction with vorticity or entropy waves [70], [71]. The shock is captured sharply and certain key quantities related to the interaction between the shock and the smooth part of the flow, such as the amplification and generation factors when a wave passes through a shock, are well resolved. Another example is the homogeneous turbulence for compressible Navier-Stokes equations studied in [71]. In one of the test cases, the spectral method can resolve all the scales using a 256^2 grid, while third order ENO with just 64^2 points can adequately resolve certain interesting quantities although it cannot resolve local quantities achieved inside the rapid transition region such as the minimum divergence. The conclusion seems to be that, when fully resolving the flow is either impossible or too costly, a "capturing" scheme such as ENO can be used on a coarse grid to obtain at least some partial information about the flow.

We thus expect that, also for the incompressible flow, we can use high-order ENO or WENO schemes on a coarse grid, without fully resolving the flow, but still get back some useful information.

A pioneer work in applying shock capturing compressible flow techniques to incompressible flow is by Bell, Colella and Glaz [6], in which they considered a second order Godunov type discretization, investigated the projection into divergence-free velocity fields for general boundary conditions, and discussed accuracy of time discretizations. Higher order ENO and WENO schemes for incompressible flows are extensions of such methods.

We solve (5.8) in its equivalent projection form

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = \mathbf{P}\left[-\begin{pmatrix} u^2 \\ uv \end{pmatrix}_x - \begin{pmatrix} uv \\ v^2 \end{pmatrix}_y + \mu\left(\begin{pmatrix} u \\ v \end{pmatrix}_{xx} + \begin{pmatrix} u \\ v \end{pmatrix}_{yy} \right) \right] \tag{5.9}$$

where $\mathbf{P}$ is the Hodge projection into divergence-free fields, i.e., if $\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \mathbf{P}\begin{pmatrix} u \\ v \end{pmatrix}$, then $\tilde{u}_x + \tilde{v}_y = 0$ and $\tilde{v}_y - \tilde{u}_x = v_y - u_x$. See, e.g., [6]. For the current periodic case the additional condition to obtain a unique projection $\mathbf{P}$ is that the mean values of u and v are preserved, i.e., $\int_0^{2\pi}\int_0^{2\pi} \tilde{u}(x,y)dxdy = \int_0^{2\pi}\int_0^{2\pi} u(x,y)dxdy$ and $\int_0^{2\pi}\int_0^{2\pi} \tilde{v}(x,y)dxdy = \int_0^{2\pi}\int_0^{2\pi} v(x,y)dxdy$.

We use N_x and N_y (even numbers) equally spaced grid points in x and y, respectively. The grid sizes are denoted by $\Delta x = \frac{N_x}{2\pi}$ and $\Delta y = \frac{N_y}{2\pi}$, and the grid points are denoted by $x_i = i\Delta x$ and $y_j = j\Delta y$. The approximated numerical values of u and v at the grid point (x_i, y_j) are denoted by u_{ij} and v_{ij}.

We first describe the numerical implementation of the projection $\mathbf{P}$. In the periodic case this is easily achieved in the Fourier space. We first expand u and v using Fourier collocation:

$$u_N(x,y) = \sum_{l=-\frac{N_y}{2}}^{\frac{N_y}{2}} \sum_{k=-\frac{N_x}{2}}^{\frac{N_x}{2}} \hat{u}_{kl} e^{I(kx+ly)}, \qquad v_N(x,y) = \sum_{l=-\frac{N_y}{2}}^{\frac{N_y}{2}} \sum_{k=-\frac{N_x}{2}}^{\frac{N_x}{2}} \hat{v}_{kl} e^{I(kx+ly)} \tag{5.10}$$

where $I = \sqrt{-1}$, $\hat{u}_{kl}$ and $\hat{v}_{kl}$ are the Fourier collocation coefficients which can be computed from the point values u_{ij} and v_{ij}, using either FFT or matrix-vector multiplications. The detail can be found in, e.g., [9]. Derivatives, either by spectral method or by central differences, involve only multiplications by factors d_k^x or d_l^y in (5.10) because $e^{I(kx+ly)}$ are eigenfunctions of such derivative operators. For example,

$$d_k^x = Ik, \qquad d_l^y = Il \tag{5.11}$$

for spectral derivatives;

$$d_k^x = \frac{2I\sin(\frac{k\Delta x}{2})}{\Delta x}, \qquad d_l^y = \frac{2I\sin(\frac{l\Delta y}{2})}{\Delta y} \tag{5.12}$$

for the second order central differences which, when used twice, will produce the second order central difference approximation $\frac{w_{i+1}-2w_i+w_{i-1}}{\Delta x^2}$ for w_{xx}, and

$$\begin{aligned} d_k^x &= \frac{2I\sqrt{(1-\cos(k\Delta x))(7-\cos(k\Delta x))}}{\Delta x}, \\ d_l^y &= \frac{2I\sqrt{(1-\cos(l\Delta y))(7-\cos(l\Delta y))}}{\Delta y} \end{aligned} \tag{5.13}$$

for the fourth order central differences which, when used twice, will produce the fourth order central difference approximation $\frac{16(w_{i+1}+w_{i-1})-(w_{i+2}+w_{i-2})-30w_i}{12\Delta x^2}$ for w_{xx}. High order filters, such as the exponential filter [55], [46]:

$$\sigma_k^x = e^{-\alpha(\frac{k}{N_x})^{2p}}, \qquad \sigma_l^y = e^{-\alpha(\frac{l}{N_y})^{2p}} \tag{5.14}$$

where $2p$ is the order of the filter and α is chosen so that $\epsilon^{-\alpha}$ is machine zero, can be used to enhance the stability while keeping at least $2p$-th order of accuracy. This is especially helpful when the projection $\mathbf{P}$ is used for the under-resolved coarse grid with ENO methods. We use the fourth order projection (5.13) and the filter (5.14) with $2p = 8$ in our calculations. This will guarantee third order accuracy (fourth order in L_1) of the ENO scheme. We will denote this combination (the fourth order projection plus the eighth order filtering) by $\mathbf{P}_4$. To be precise, if $\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \mathbf{P}_4 \begin{pmatrix} u \\ v \end{pmatrix}$ and $\hat{u}_{kl}$ and $\hat{v}_{kl}$ are Fourier collocation coefficients of u and v, then the Fourier collocation coefficients of $\tilde{u}$ and $\tilde{v}$ are given by

$$\hat{\tilde{u}} = \sigma_k^x \sigma_l^y \frac{d_l^y (d_l^y \hat{u} - d_k^x \hat{v})}{(d_k^x)^2 + (d_l^y)^2}, \qquad \hat{\tilde{v}} = \sigma_k^x \sigma_l^y \frac{-d_k^x (d_l^y \hat{u} - d_k^x \hat{v})}{(d_k^x)^2 + (d_l^y)^2} \tag{5.15}$$

where σ_k^x and σ_l^y are defined by (5.14) with $2p = 8$, and d_k^x and d_l^y are defined by (5.13).

Next we shall describe the ENO scheme for (5.8). Since (5.8) is equivalent to the non-conservative form (5.7), it is natural to implement upwinding by the signs of u and v, and to implement ENO equation by equation (the component version described in Sect. 2.3.5). The r-th order ENO approximation of, e.g., $(u^2)_x$ is thus carried out using the ENO Procedure 4.2. We mention a couple of facts needing attention:

1. Take $f(x) = u^2(x, y)$ with y fixed. We start with the point values $f_i = f(x_i)$;
2. The stencil of the reconstruction is determined adaptively by upwinding and smoothness of $f(x)$. It starts with either x_j or x_{j+1} according to whether $u \geq 0$ or $u < 0$.

There are two ways to handle the second derivative terms for the Navier-Stokes equations. One can absorb them into the convection part and treat them using ENO. For example, $f(x) = u^2(x, y)$ can be replaced by $f(x) = u^2(x, y) - \mu u(x, y)_x$, where $u(x, y)_x$ itself can be obtained using either ENO or central difference of a suitable order. The remaining procedure for computing $f(x)_x$ would be the same as described above. Another simpler possibility is just to use standard central differences (of suitable order) to compute the double derivative terms. Our experience with compressible flow is that there is little difference between the two approaches, especially when the viscosity μ is small.

In the above we have described the discretization for the spatial derivatives

$$L_{ij} \approx \left[-\begin{pmatrix} u^2 \\ uv \end{pmatrix}_x - \begin{pmatrix} uv \\ v^2 \end{pmatrix}_y + \mu \left(\begin{pmatrix} u \\ v \end{pmatrix}_{xx} + \begin{pmatrix} u \\ v \end{pmatrix}_{yy} \right) \right]_{\substack{x = x_i \\ y = y_j}} . \tag{5.16}$$

We then use the third order TVD (total variation diminishing) Runge-Kutta method (4.11) to discretize the resulting ODE:

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = \mathbf{P}_4 L_{ij} \tag{5.17}$$

obtaining:

$$\begin{pmatrix} u \\ v \end{pmatrix}^{(1)} = \mathbf{P}_4 \left[\begin{pmatrix} u \\ v \end{pmatrix}^n + \Delta t L_{ij}^n \right]$$

$$\begin{pmatrix} u \\ v \end{pmatrix}^{(2)} = \mathbf{P}_4 \left[\frac{3}{4} \begin{pmatrix} u \\ v \end{pmatrix}^{n} + \frac{1}{4} \begin{pmatrix} u \\ v \end{pmatrix}^{(1)} + \frac{1}{4} \Delta t L_{ij}^{(1)} \right] \tag{5.18}$$

$$\begin{pmatrix} u \\ v \end{pmatrix}^{n+1} = \mathbf{P}_4 \left[\frac{1}{3} \begin{pmatrix} u \\ v \end{pmatrix}^{n} + \frac{2}{3} \begin{pmatrix} u \\ v \end{pmatrix}^{(2)} + \frac{2}{3} \Delta t L_{ij}^{(2)} \right]$$

Notice that we have used the property $P_4 \circ P_4 = P_4$ in obtaining the discretization (5.18) from (5.17).

This explicit time discretization is expected to be nonlinearly stable under the CFL condition

$$\Delta t \left[\max_{i,j} \left(\frac{|u_{ij}|}{\Delta x} + \frac{|v_{ij}|}{\Delta y} \right) + 2\mu \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right] \leq 1 \tag{5.19}$$

For small μ (which is the case we are interested in) this is not a serious restriction on Δt.

We present some numerical examples in the following.

Example 5.1: This example is used to check the third order accuracy of our ENO scheme for smooth solutions. We first take the initial condition as

$$u(x, y, 0) = -\cos(x)\sin(y), \qquad v(x, y, 0) = \sin(x)\cos(y) \tag{5.20}$$

which was used in [6]. The exact solution for this case is known:

$$u(x, y, t) = -\cos(x)\sin(y)e^{-2\mu t}, \qquad v(x, y, t) = \sin(x)\cos(y)e^{-2\mu t} \tag{5.21}$$

We take $\Delta x = \Delta y = \frac{1}{N}$ with $N = 32, 64, 128$ and 256. The solution is computed up to $t = 2$ and the L_2 error and numerical order of accuracy are listed in Table 5.1. For the $\mu = 0.05$ case, we list results both with fourth order central approximation to the double derivative terms (central) and with ENO to handle the double derivative terms by absorbing them into the convection part (ENO). We can clearly observe fully third order accuracy (actually better in many cases because the spatial ENO is fourth order in the L_1 sense) in this table.

Example 5.2: This is our test example to study resolution of ENO schemes when the grid is coarse. It is a double shear layer taken from [6]:

$$u(x, y, 0) = \begin{cases} \tanh((y - \pi/2)/\rho) & y \leq \pi \\ \tanh((3\pi/2 - y)/\rho) & y > \pi \end{cases} \qquad v(x, y, 0) = \delta \sin(x) \tag{5.22}$$

where we take $\rho = \pi/15$ and $\delta = 0.05$. The Euler equations ($\mu = 0$) are used for this example. The solution quickly develops into roll-ups with smaller and smaller scales, so on any fixed grid the full resolution is lost eventually. For example, the expensive run we performed using 512^2 points for the spectral collocation code (with a 18-th order filter (5.14)) is able to resolve the solution fully up to $t = 8$, Fig. 5.10, top left, as verified by the spectrum of the solution (not shown here), but begins to lose resolution as indicated by the wriggles in the vorticity contour at $t = 10$ (not shown here). On the other hand, the ENO runs with 64^2 and 128^2 points produces smooth, stable results Fig. 5.10, top right and bottom left. In Fig. 5.10, bottom right, we show a cut at $x = \pi$ for v at $t = 8$. This gives a better feeling about the resolution in physical space. Apparently with these coarse grids the full structure of the roll-up is not resolved. However, when we compute the total circulation

$$c_\Omega = \int_\Omega \omega(x, y) dx dy = \int_{\partial\Omega} u dx + v dy \tag{5.23}$$

Table 5.1: Accuracy of ENO Schemes for (12.2).

N	$\mu = 0$		$\mu = 0.05$, central		$\mu = 0.05$, ENO	
	L_2 error	order	L_2 error	order	L_2 error	order
32	9.10(-4)		5.28(-4)		4.87(-4)	
64	5.73(-5)	3.99	3.20(-5)	4.04	3.09(-5)	3.98
128	3.62(-6)	3.98	1.93(-6)	4.05	1.89(-6)	4.03
256	2.28(-7)	3.99	1.18(-7)	4.03	1.16(-7)	4.03

N	$\mu = 0$			$\mu = 0.05$, central			$\mu = 0.05$, ENO		
	L_2 diff	order	error	L_2 diff	order	error	L_2 diff	order	error
32	1.14(-1)			3.20(-2)			3.60(-2)		
64	1.40(-2)	3.02	1.96(-3)	2.78(-3)	3.52	2.66(-4)	2.93(-3)	3.62	2.60(-4)
128	1.46(-3)	3.26	1.69(-4)	1.81(-4)	3.94	1.26(-5)	1.80(-4)	4.02	1.18(-5)
256	1.11(-4)	3.77	8.78(-6)	1.09(-5)	4.06	6.91(-7)	1.10(-5)	4.04	7.15(-7)

Table 5.2: Resolution of the Total Circulation.

t	2	4	6	8	10
ENO 64^2	0.87300	3.07100	7.16889	9.88063	10.90122
ENO 128^2	0.87452	2.97810	7.30999	10.34414	11.79418
spectral 512^2	0.87433	2.98029	7.28308	10.46212	11.85875

around the roll-up by taking $\Omega = [\frac{\pi}{2}, \frac{3\pi}{2}] \times [0, 2\pi]$ and using the rectangular rule (which is infinite order accurate for the periodic case) on the line integrals at the right-hand-side of (5.13), we can see that this number is resolved much better than the roll-up itself, Table 5.2.

As an application of ENO scheme for incompressible flow, we consider the motion of an incompressible fluid, in two and three dimensions, in which the vorticity is concentrated on a lower dimensional set [31]. Prominent examples are vortex sheets and vortex filaments in three dimensions, and vortex sheets, vortex dipole sheets and point vortices in two dimensions.

In three dimensions, the equations are written in the form

$$\begin{aligned} \xi_t + v\nabla\xi - \nabla v\ \xi &= 0 \\ \nabla \times v &= \xi \\ \nabla \cdot v &= 0 \end{aligned} \tag{5.24}$$

where $\xi(x, y, z, t)$ is the vorticity vector, and $v(x, y, z, t)$ is the velocity vector.

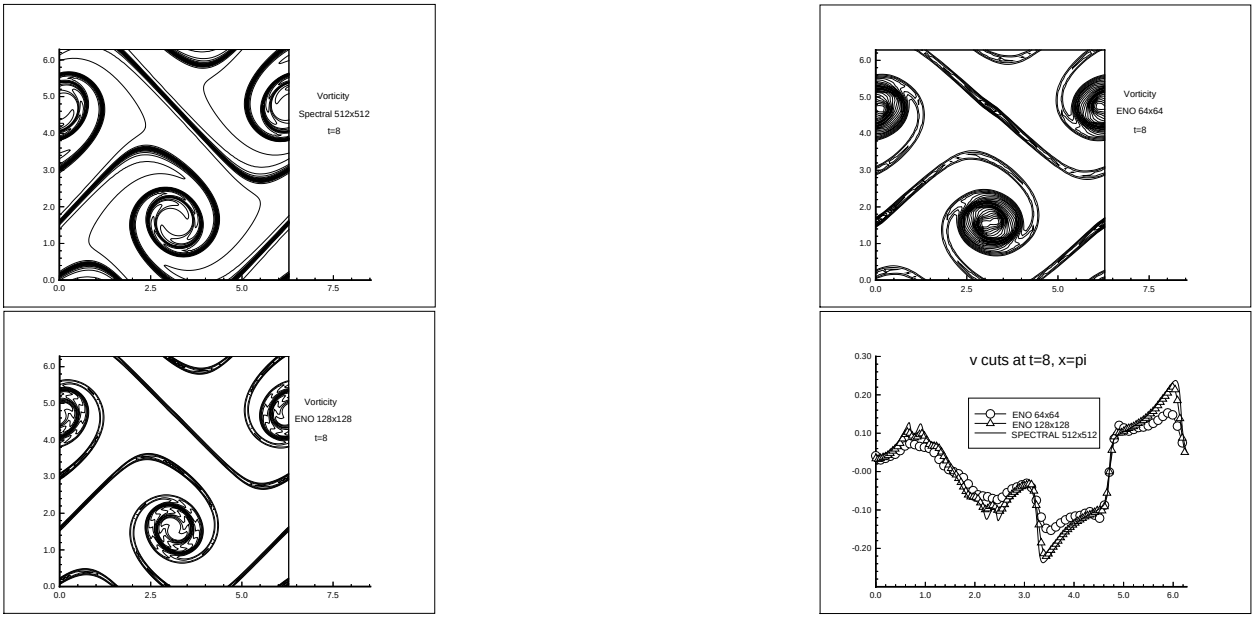


Figure 5.10: Double shear layer. Contours of vorticity. $t = 8$. Top left: spectral with 512^2 points; top right: ENO with 64^2 points; bottom left: ENO with 128^2 points; bottom right: the cut at $x = \pi$ of v, spectral method with 512^2 points, ENO method with 64^2 and with 128^2 points.

In a vortex sheet, ξ is a singular measure concentrated on a two dimensional surface, while in a vortex filament, ξ is a function concentrated on a tubular neighborhood of a curve.

We use an Eulerian, fixed grid, approach, that works in general in two and three dimensions. In the particular case of the two dimensional vortex sheet problem in which the vorticity does not change sign, the approach yields a very simple and elegant formulation.

The basic observation involves a variant of the level set method for capturing fronts, developed in [59].

The formulation we use here regularizes general ill-posed problems via the level set approach, using the idea that a simple closed curve which is the level set of a function cannot change its index, i.e. there is an automatic topological regularization. This is very helpful for numerical calculations. The regularization is automatically accomplished through the use of dissipative schemes, which has the effect of adding a small curvature term (which vanishes as the grid size goes to zero) to the evolution of the interface. The formulation allows for topological changes, such as merging of surfaces.

The main idea is to decompose ξ into a product of the form

$$\xi = P(\varphi)\eta \tag{5.25}$$

where P is a scalar function, typically an approximate δ function. The variable φ is a scalar function whose zero level set represents the points where vorticity concentrates, and η represents the vorticity strength vector. This decomposition is performed at time zero and is of course not unique.

The observation is that once a decomposition is found, the following system of equations yields a solution to the Euler equations, replacing the original set of equations (5.24).

$$\begin{aligned} \varphi_t + v\nabla\varphi &= 0 \\ \eta_t + v\nabla\eta - \nabla v\ \eta &= 0 \\ \nabla \times v &= P(\varphi)\eta \\ \nabla \cdot v &= 0 \end{aligned} \tag{5.26}$$

These equations have initial conditions

$$\begin{aligned} \varphi(0,\cdot) &= \varphi_0 \\ \eta(0,\cdot) &= \eta_0 \end{aligned}$$

where φ_0, η_0 and P are chosen so that (5.25) holds at time $t = 0$. Notice that (5.25) and (5.26) imply that $\nabla\varphi$ is orthogonal to η, and $div(\eta) = 0$. This is enforced in the initial condition and is maintained automatically by (5.25) and (5.26).

When P is a distribution, such as a δ function, approaching P with a sequence of smooth mollifiers P_ϵ yields a sequence of approximating solutions. This is the approach used in numerical calculations, since the δ function can only be represented approximately on a finite grid. The parameter ϵ is usually chosen to be proportional to the mesh size.

The advantage of this formulation, is that it replaces a possibly singular and unbounded vorticity function ξ, by bounded, smooth (at least uniformly Lipschitz) functions φ and η. Therefore, while it is not feasible to compute solutions of (5.24) directly, it is very easy to compute solutions of (5.26).

In two dimensions, the vorticity is given by

$$\xi = \begin{pmatrix} 0 \\ 0 \\ \omega(t,x,y) \end{pmatrix}$$

and hence the Euler equations are given by

$$\begin{aligned} \omega_t + v\nabla\omega &= 0 \\ curl(v) &= \omega \qquad (5.27) \\ div(v) &= 0 \qquad (5.28) \end{aligned}$$

Our formulation (5.26), becomes

$$\begin{aligned} \varphi_t + v\nabla\varphi &= 0 \\ \eta_t + v\nabla\eta &= 0 \\ curl(v) &= P(\varphi)\eta \\ div(v) &= 0 \end{aligned} \tag{5.29}$$

where η is now a scalar.

If the vortex sheet strength η does not change sign along the curve, it can be normalized to $\eta \equiv 1$ and the equations take on a particularly simple and elegant form:

$$\varphi_t + v(\varphi)\nabla\varphi = 0 \tag{5.30}$$

where the velocity $v(\varphi)$ is given by

$$v = -\begin{pmatrix} -\partial_y \\ \partial_x \end{pmatrix} \triangle^{-1} P(\varphi) \tag{5.31}$$

In this case, the vortex sheet strength along the curve is given by $\frac{1}{|\nabla\varphi|}$ (see (5.33)).

Example 5.3: *Vortex Sheets in 2D.* We consider the periodic vortex sheet in two dimensions, i.e. $P(\varphi) = \delta(\varphi)$ in (5.31). The three dimensional case is defined in detail later. The evolution of the vortex sheet in the Lagrangian framework has been considered by various authors. Krasny [47], [48] has computed vortex sheet roll-up using vortex blobs and point vortices with filtering. Baker and Shelley [4] have approximated the vortex sheet by a layer of constant vorticity which they computed by Lagrangian methods. In the context of our approach, their approximation corresponds to approximating the δ function by a step function.

In our framework, we use a fixed Eulerian grid, and approximate (5.30) by the third order upwind ENO finite difference scheme with a third order TVD Runge-Kutta time stepping. At every time step, the velocity v is first obtained by solving the Poisson equation for the stream function Ψ:

$$\Delta\Psi = -P(\varphi)$$

with boundary conditions

$$\Psi(x, \pm 1) = 0$$

and periodic in x. This is done by using a second order elliptic solver FISHPAK. Once Ψ is obtained, the velocity is recovered by $v = (-\Psi_y, \Psi_x)$ by using either ENO or central difference approximations (we do not observe major difference among the two: the results shown are those obtained by central difference). Once v is obtained, upwind biased ENO is easily applied to (5.30).

The initial conditions are similar to the ones in [48], i.e given by a sinusoidal perturbation of a flat sheet:

$$\varphi_0(x, y) = y + 0.05\sin(\pi x)$$

The boundary condition for φ are periodic, of the form:

$$\varphi(t, -1, y) = \varphi(t, 1, y)$$

$$\varphi(t, x, -1) = \varphi(t, x, 1) - 2$$

The δ function is approximated as in [61],[77] by

$$\delta_\epsilon(\phi) = \begin{cases} \frac{1}{2\epsilon}(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right)) & \text{if } |\varphi| < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{5.32}$$

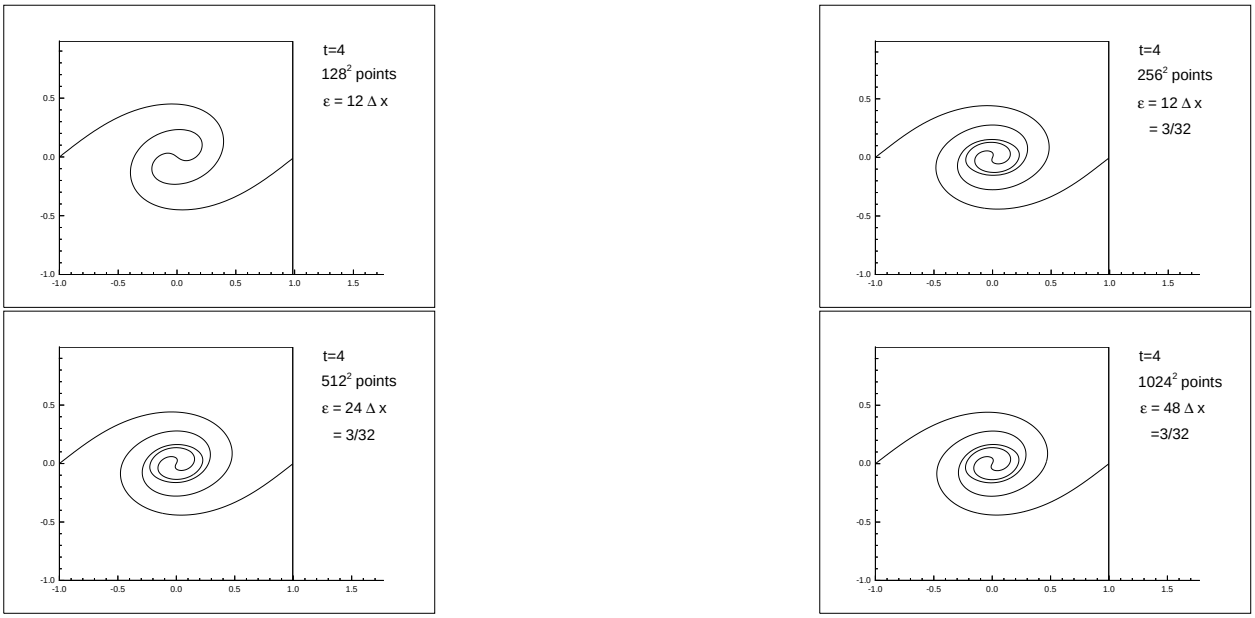


Figure 5.11: Two dimensional vortex sheet simulation. $t = 4$. Top left: ENO with 128^2 points, δ function width $\epsilon = 12\Delta x = \frac{3}{16}$; Top right: ENO with 256^2 points, δ function width $\epsilon = 12\Delta x = \frac{3}{32}$; Bottom left: ENO with 512^2 points, δ function width $\epsilon = 24\Delta x = \frac{3}{32}$; Bottom right: ENO with 1024^2 points, δ function width $\epsilon = 48\Delta x = \frac{3}{32}$.

For fixed ϵ, there is convergence as $\Delta x \to 0$ to a smooth solution. One can then take $\epsilon \to 0$. This two step limit is very costly to implement numerically. Our numerical results show that one can take ϵ to be proportional to Δx, but convergence is difficult to establish theoretically.

In Fig. 5.11, top left, we present the result at $t = 4$, of using ENO with 128^2 grid points with the parameter ϵ in the approximate δ function chosen as $\epsilon = 12\Delta x$. We use the graphic package TECPLOT to draw the level curve of $\varphi = 0$. Next, we keep $\epsilon = 12\Delta x$ but double the grid points in each direction to 256^2, the result of $t = 4$ is shown in Fig. 5.11, top right. Comparing with Fig. 5.11, top left, we can see that there are more turns in the core at the same physical time when the grid size is reduced and the δ function width ϵ is kept proportional to Δx. One might wonder whether the core structure of Fig. 5.11, top right, is distorted by numerical error. To verify that this is not the case, we keep $\epsilon = 12 \times \frac{2}{256} = \frac{3}{32}$ *fixed*, and reduce Δx, Fig. 5.11, bottom left and right. The three pictures overlay very well, the bottom two pictures in Fig. 5.11 are indistinguishable, indicating that the core structure is a resolved solution to the problem and convergence is obtained with fixed ϵ. By reducing ϵ for the more refined grids, more turns in the core can be obtained in shorter time (pictures not shown).

The smoothing of the δ function, and the third order truncation error in the advection step and the second order error in the inverse Laplacian are the only smoothing steps in our method.

We now give the same example in three dimensions. We first sketch the algorithm for initializing and computing a periodic 3D vortex sheet, using (5.26).

We let $P(\varphi) = \delta(\varphi)$ (in practice δ is replaced by an approximation). The zero level set of φ is the vortex sheet $\Gamma(s)$, parameterized by surface area s. The variable η_0 is chosen to

fit the initial vortex sheet strength. For instance, given any smooth test function g

$$\begin{aligned}\langle \xi, g\rangle &= \langle \eta_0 \delta(\varphi_0), g\rangle \\ &= \int \eta_0(\Gamma_0(s)) g(\Gamma_0(s)) \frac{1}{|\nabla \varphi_0|} ds\end{aligned}$$

Thus, the initial vortex sheet strength is given by

$$\frac{\eta_0}{|\nabla \varphi_0|} \tag{5.33}$$

To obtain the velocity vector, one introduces the vector potential A, where

$$v = \nabla \times A, \quad div(A) = 0$$

and solves the Poisson equation

$$\triangle A = -P(\varphi)\eta \tag{5.34}$$

To ensure that $div(A) = 0$, we require that $div(\eta) = 0$ and that $\nabla\varphi \cdot \eta = 0$ initially. It is easy to see that these equalities are maintained as t increases.

The boundary conditions for the velocity are $v_2(x, \pm 1, z) = 0$ and periodic in x and z. To obtain the boundary conditions for $A = (A_1, A_2, A_3)$, we use the divergence free condition on A in addition to the velocity boundary condition. Thus,

$$\begin{aligned} A_1(x, \pm 1, z) = A_3(x, \pm 1, z) &= 0 \\ \partial_y A_2(x, \pm 1, z) &= 0 \end{aligned} \tag{5.35}$$

and periodic in x, z. The Neumann condition requires the following compatibility condition

$$\int \xi_2(x, y, z, 0) dx dy dz = 0$$

Three dimensional runs are much more expensive than two dimensional runs, not only because the number of grid points increases, but also because there are now four evolution equations (for φ and η), and three potential equations. We still use the third order ENO scheme coupled with the second order elliptic solver FISHPAK, with 64^3 grid points, and ϵ is chosen as $6\Delta x$, which is the same in magnitude as that used in Fig. 5.11 of Example 5.3. The boundary conditions for φ are similar to the ones in two dimensions: periodic in all directions (module the linear term in y). The vortex sheet strength vector η is periodic in all directions.

We first verify whether we can recover the two dimensional results with the three dimensional setting. We use the initial condition

$$\varphi_0(x, y, z) = y + 0.05 \sin(\pi x)$$

which is the same as that for Example 5.3, and choose a constant initial condition for η as $\eta_0(x, y, z) = (0, 0, 1)$. We observe exact agreement with our two dimensional results in

Figure 5.12: Three dimensional vortex sheet simulation. $t = 5$. ENO with 64^3 points. δ function width $\epsilon = 6\Delta x$. Left: three dimensional level surface; Right: $z = 0$ plane cut.

Example 5.3, Fig. 5.11. Next, we consider the truly three dimensional problem with the initial condition chosen as

$$\varphi_0(x, y, z) = y + 0.05\sin(\pi x) + 0.1\sin(\pi z)$$

and η is chosen as $\eta_0(x, y, z) = (0, -0.1\pi\cos(\pi z), 1)$ which satisfies the divergence free condition as well as the condition to be orthogonal to $\nabla\varphi$. In Fig. 5.12, left, we show the level set of $\varphi = 0$ for $t = 5$. We can clearly see the roll up process and the three dimensional features. The cut at the constants $z = 0$ plane is shown in Fig. 5.12, right.

5.3 Applications in Semiconductor Device Simulation

An interesting application area for ENO and WENO schemes is the equations in semiconductor device simulations. During the last decade, semiconductor device modeling has attempted to incorporate general carrier heating, velocity overshoot, and various small device features into carrier simulation. The popular wisdom emerging from such concentrated study holds that global dependence of critical quantities, such as mobilities, on energy and/or temperature, is essential if such phenomena are to be modeled adequately.

This gives rise to the various energy transport models, including the hydrodynamic model and the ET model, see, e.g. [41]. Unlike the earlier drift-diffusion models, which are basically parabolic, these new models contain significant transport effects [42], thus calling for discretization techniques suitable for hyperbolic problems.

In this section we present two of such models.

The first one is the hydrodynamic model. It is obtained by taking the first three moments of the Boltzmann equation. In the conservative format the hydrodynamic model is written as follows. Define the vector of dependent variables as

$$u = (n, \sigma, \tau, W), \tag{5.36}$$

where n is the electron concentration, $p = (\sigma, \tau)$ is the momenta, and W is the total energy. The equations, in two dimensions, take the form

$$u_t + f_1(u)_x + f_2(u)_y = c(u) + G(u, \phi) + (0, 0, 0, \nabla\cdot(\kappa\nabla T)). \tag{5.37}$$

where

$$f_1(u) = (\frac{\sigma}{m},\ \frac{2}{3}(\frac{\sigma^2}{mn} + W - \frac{\tau^2}{2mn}),\ \frac{\sigma\tau}{mn},\ \frac{5\sigma W}{3mn} - \sigma\frac{\sigma^2+\tau^2}{3m^2n^2}), \tag{5.38}$$

$$f_2(u) = (\frac{\tau}{m},\ \frac{\sigma\tau}{mn},\ \frac{2}{3}(\frac{\tau^2}{mn} + W - \frac{\sigma^2}{2mn}),\ \frac{5\tau W}{3mn} - \tau\frac{\sigma^2+\tau^2}{3m^2n^2}), \tag{5.39}$$

$$\begin{aligned} c(u) &= (0, -\frac{\sigma}{\tau_p}, -\frac{\tau}{\tau_p}, -\frac{W-W_0}{\tau_w}), & (5.40)\\ G(u) &= (0, -enF_1, -enF_2, -enF\cdot v). & (5.41) \end{aligned}$$

Here, F is the electric field, obtained by solving a Poisson's equation:

$$\begin{aligned} F &= -\nabla\phi, & (5.42)\\ \nabla\cdot(\epsilon\nabla\phi) &= -e\,n - n_d. & (5.43) \end{aligned}$$

where n_d is the doping (a given function which is typically discontinuous).

The second model is the the energy transport model, written as

$$u_t + f(u)_x = g(u)_{xx} + h(u). \tag{5.44}$$

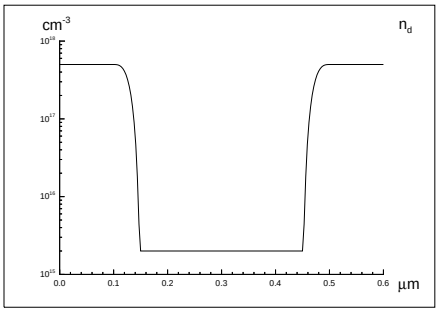
In equation (5.44),

$$\begin{aligned} u &= (en, \frac{nE}{m}), & (5.45)\\ f(u) &= \phi' n\ (e\mu(E),\ \mu^E(E) + D(E)), & (5.46)\\ g(u) &= (nD(E),\ nD^E(E)), & (5.47)\\ h(u) &= (0,\ en\mu(E)(\phi')^2 + \frac{e}{\epsilon}(n-n_d)nD(E) - n\langle\frac{\partial E}{\partial t}\ |_{coll}\rangle). & (5.48) \end{aligned}$$

It can be shown that the left hand side defines a hyperbolic system, since the eigenvalues of $f'(u)$ are real, for all positive n and T.

We first present one dimensional numerical results. The one dimensional n^+-n-n^+ channel we simulate is a standard silicon diode with a length of $0.6\mu m$, with a doping defined by $n_d = 5\times10^{17}cm^{-3}$ in $[0, 0.1]$ and in $[0.5, 0.6]$, and $n_d = 2\times10^{15}cm^{-3}$ in $[0.15, 0.45]$, joined by smooth junctions (Fig. 5.13, left). The lattice temperature is taken as $T_0 = 300$ K. We apply a voltage bias of $vbias = 1.5$V. We use the full HD model; the relevant parameters can be found in [41]. In Fig. 5.13, right, we present the simulated velocity using the HD model. The dashed line shows the result computed with a reduced HD model by ignoring the transport effects. This type of reduced HD models are used quite often in engineering, as they tend to reduce the numerical difficulty when standard (not high resolution) schemes are used. However, we can see here that there is significant difference in the simulated results.

We now present numerical simulation results for one carrier, two dimensional MESFET devices. The third order ENO shock-capturing algorithm with Lax-Friedrichs building blocks, as described elsewhere in these lecture notes, is applied to the hyperbolic part (the left hand side) of Equations (5.37) and (5.44). The TVD third order Runge-Kutta time discretization (4.11) is used for the time evolution towards steady states. The forcing terms



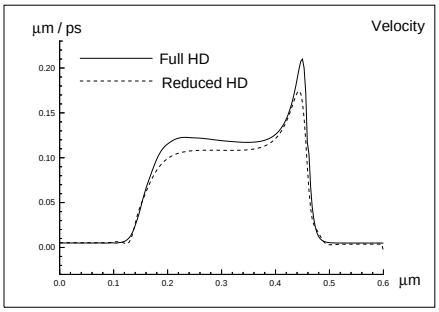


Figure 5.13: The one dimensional n^+-n-n^+ channel. Left: the doping n_d; Right: the velocity v, comparison of the HD model and the reduced HD model.

on the right hand side of (5.37) and (5.44) are treated in a time consistent way in the Runge-Kutta time stepping. The double derivative terms on the right hand side of (5.37) and (5.44) are approximated by standard central differences owing to their dissipative nature. The Poisson equation (5.43) is solved by direct Gauss elimination for one spatial dimension and by Successive Over-Relaxation (SOR) or the Conjugate Gradient (CG) method for two spatial dimensions. Initial conditions are chosen as $n = n_d$ for the concentration, $T = T_0$ for the temperature, and $u = v = 0$ for the velocities. A continuation method is used to reach the steady state: the voltage bias is taken initially as zero and is gradually increased to the required value, with the steady state solution of a lower biased case used as the initial condition for a higher one.

We simulate a two dimensional MESFET of the size $0.6 \times 0.2\mu m^2$. The source and the drain each occupies $0.1\mu m$ at the upper left and the upper right, respectively, with the gate occupying $0.2\mu m$ at the upper middle (Fig. 5.14, top left). The doping is defined by $n_d = 3 \times 10^{17} cm^{-3}$ in $[0, 0.1] \times [0.15, 0.2]$ and in $[0.5, 0.6] \times [0.15, 0.2]$, and $n_d = 1 \times 10^{17} cm^{-3}$ elsewhere, with abrupt junctions (Fig. 5.14, top right). A uniform grid of 96×32 points is used. Notice that even if we may not have shocks in the solution, the initial condition $n = n_d$ is discontinuous, and the final steady state solution has a sharp transition around the junction. With the relatively coarse grid we use, the non-oscillatory shock capturing feature of the ENO algorithm is essential for the stability of the numerical procedure.

We apply, at the source and drain, a voltage bias $vbias = 2V$. The gate is a Schottky contact, with a negative voltage bias $vgate = -0.8V$ and a very low concentration value $n = 3.9 \times 10^5 cm^{-3}$. The lattice temperature is taken as $T_0 = 300°K$. The numerical boundary conditions are summarized as follows (where $\Phi_0 = \frac{k_b T}{e} \ln\left(\frac{n_d}{n_i}\right)$ with $k_b = 0.138 \times 10^{-4}$, $e = 0.1602$, and $n_i = 1.4 \times 10^{10} cm^{-3}$ in our units):

- At the source ($0 \le x \le 0.1, y = 0.2$): $\Phi = \Phi_0$ for the potential; $n = 3 \times 10^{17} cm^{-3}$ for the concentration; $T = 300°K$ for the temperature; $u = 0\mu m/ps$ for the horizontal velocity; and Neumann boundary condition for the vertical velocity v (i.e. $\frac{\partial v}{\partial \vec{n}} = 0$ where $\vec{n}$ is the normal direction of the boundary).

- At the drain ($0.5 \le x \le 0.6, y = 0.2$): $\Phi = \Phi_0 + vbias = \Phi_0 + 2$ for the potential; $n = 3 \times 10^{17} cm^{-3}$ for the concentration; $T = 300°K$ for the temperature; $u = 0\mu m/ps$ for the horizontal velocity; and Neumann boundary condition for the vertical velocity v.

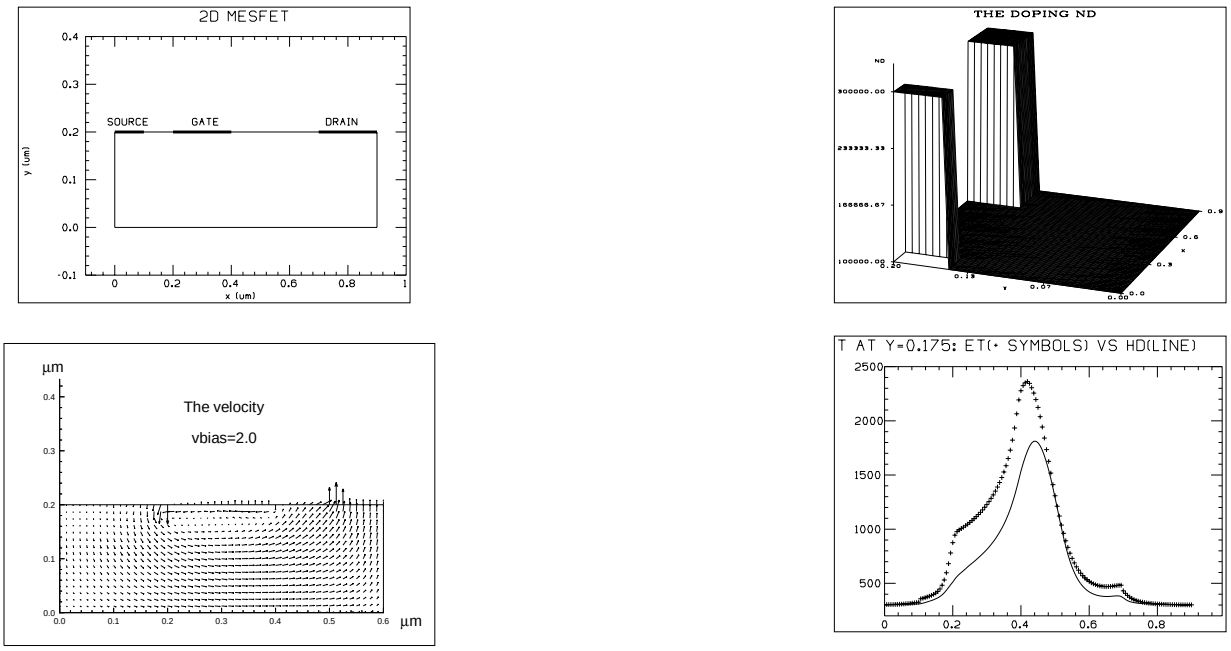


Figure 5.14: Two dimensional MESFET. Top left: the geometry; Top right: the doping n_d; Bottom left: the velocity (u, v) obtained by the HD model; Bottom right: comparison of the hydrodynamic (HD) model (solid line) and the energy transport (ET) model (plus symbols), cut at the middle of the high doping blobs $y = 0.175$, the temperature T.

- At the gate $(0.2 \leq x \leq 0.4, y = 0.2)$: $\Phi = \Phi_0 + vgate = \Phi_0 - 0.8$ for the potential; $n = 3.9 \times 10^5 cm^{-3}$ for the concentration; $T = 300° K$ for the temperature; $u = 0 \mu m/ps$ for the horizontal velocity; and Neumann boundary condition for the vertical velocity v.

- At all other parts of the boundary $(0.1 \leq x \leq 0.2, y = 0.2;\ 0.4 \leq x \leq 0.5, y = 0.2;$ $x = 0, 0 \leq y \leq 0.2;\ x = 0.6, 0 \leq y \leq 0.2;$ and $0 \leq x \leq 0.6, y = 0)$, all variables are equipped with Neumann boundary conditions.

The boundary conditions chosen are based upon physical and numerical considerations. They may not be adequate mathematically, as is evident from some serious boundary layers observable in the concentration (see pictures in [41]). ENO methods, owing to their upwind nature, are robust to different boundary conditions (including over-specified boundary conditions) and do not exhibit numerical difficulties in the presence of such boundary layers, even with the extremely low concentration prescribed at the gate (around 10^{-12} relative to the high doping). We point out, however, that boundary conditions affect the global solution significantly. We have also simulated the same problem with different boundary conditions, for example with Dirichlet boundary conditions everywhere for the temperature, or with Neumann boundary conditions for all variables except for the potential at the contacts. The numerical results (not shown here) are noticeably different. This indicates the importance of studying adequate boundary conditions, from both a physical and a mathematical point of view.

The velocity vectors resulting from the hydrodynamic model simulation are presented in Fig. 5.14, bottom left. In Fig. 5.14, bottom right, we compare the temperature at $y = 0.175$ from the simulations of the hydrodynamic model and of the ET model. Clearly there is a significant difference between these two models for this 2D case.

There are also new models in semiconductor device simulation (e.g. [14], [15]), which are worthy of investigations. ENO and WENO schemes provide robust and reliable tools for carrying out such investigations.

References

[1] R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation*, Journal of Computational Physics, v114 (1994), pp.45–58.

[2] N. Adams and K. Shariff, *A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems*, Journal of Computational Physics, v127 (1996), pp.27–51.

[3] H. Atkins and C.-W. Shu, *GKS and eigenvalue stability analysis of high order upwind scheme*, in preparation.

[4] G. R. Baker and M. J. Shelley, *On the connection between thin vortex layers and vortex sheets*, Journal of Fluid Mechanics, v215 (1990), pp.161–194.

[5] M. Bardi and S. Osher, *The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations*, SIAM Journal on Mathematical Analysis, v22 (1991), pp.344–351.

[6] J. Bell, P. Colella and H. Glaz, *A Second Order Projection Method for the Incompressible Navier-Stokes Equations*, Journal of Computational Physics, v85, 1989, pp.257–283.

[7] B. Bihari and A. Harten, *Application of generalized wavelets: an adaptive multiresolution scheme*, Journal of Computational and Applied Mathematics, v61 (1995), pp.275–321.

[8] W. Cai and C.-W. Shu, *Uniform high-order spectral methods for one- and two-dimensional Euler equations*, Journal of Computational Physics, v104 (1993), pp.427–443.

[9] C. Canuto, M.Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.

[10] M. Carpenter and C. Kennedy, *Fourth-order 2N-storage Runge-Kutta schemes*, NASA TM 109112, NASA Langley Research Center, June 1994.

[11] J. Casper, *Finite-volume implementation of high-order essentially nonoscillatory schemes in two dimensions*, AIAA Journal, v30 (1992), pp.2829–2835.

[12] J. Casper and H. Atkins, *A finite-volume high-order ENO scheme for two dimensional hyperbolic systems*, Journal of Computational Physics, v106 (1993), pp.62–76.

[13] J. Casper, C.-W. Shu and H. Atkins, *Comparison of two formulations for high-order accurate essentially nonoscillatory schemes*, AIAA Journal, v32 (1994), pp.1970–1977.

[14] C. Cercignani, I. Gamba, J. Jerome and C.-W. Shu, *Applicability of the high field model: an analytical study via asymptotic parameters defining domain decomposition*, VLSI Design, to appear.

[15] C. Cercignani, I. Gamba, J. Jerome and C.-W. Shu, *Applicability of the high field model: a preliminary numerical study*, VLSI Design, to appear.

[16] S. Christofi, *The study of building blocks for ENO schemes*, Ph.D. thesis, Division of Applied Mathematics, Brown University, September 1995.

[17] B. Cockburn, *Discontinuous Galerkin method*, this volume.

[18] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework*, Mathematics of Computation, v52 (1989), pp.411–435.

[19] B. Cockburn, S.-Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems*, Journal of Computational Physics, v84 (1989), pp.90–113.

[20] B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*, Mathematics of Computation, v54 (1990), pp.545–581.

[21] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, preprint. Submitted to Journal of Computational Physics.

[22] M. Crandall and P. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society, v277 (1983), pp.1–42.

[23] M. Crandall and P. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Mathematics of Computation, v43 (1984), pp.1–19.

[24] A. Dolezal and S. Wong, *Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, v120 (1995), pp.266–277.

[25] R. Donat and A. Marquina, *Capturing shock reflections: an improved flux formula*, Journal of Computational Physics, v125 (1996), pp.42–58.

[26] W. E and C.-W. Shu, *A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow*, Journal of Computational Physics, v110 (1994), pp.39–46.

[27] G. Erlebacher, Y. Hussaini and C.-W. Shu, *Interaction of a shock with a longitudinal vortex*, Journal of Fluid Mechanics, v337 (1997), pp.129–153.

[28] E. Fatemi, J. Jerome and S. Osher, *Solution of the hydrodynamic device model using high order non-oscillatory shock capturing algorithms*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, v10 (1991), pp.232–244.

[29] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Mathematics of Computation, to appear.

[30] B. Gustafsson, H.-O. Kreiss and A. Sundstrom, *Stability theory of difference approximations for mixed initial boundary value problems, II*, Mathematics of Computation, v26 (1972), pp.649–686.

[31] E. Harabetian, S. Osher and C.-W. Shu, *An Eulerian approach for vortex motion using a level set regularization procedure*, Journal of Computational Physics, v127 (1996), pp.15–26.

[32] A. Harten, *The artificial compression method for computation of shocks and contact discontinuities III: self-adjusting hybrid schemes*, Mathematics of Computation, v32 (1978), pp.363–389.

[33] A. Harten, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, v49 (1983), pp.357–393.

[34] A. Harten, *Preliminary results on the extension of ENO schemes to two dimensional problems*, in Proceedings of the International Conference on Hyperbolic Problems, Saint-Etienne, 1986.

[35] A. Harten, *ENO schemes with subcell resolution*, Journal of Computational Physics, v83 (1989), pp.148–184.

[36] A. Harten, J. Hyman and P. Lax, *On finite difference approximations and entropy conditions for shocks*, Communications in Pure and Applied Mathematics, v29 (1976), pp.297–322.

[37] A. Harten and S. Osher, *Uniformly high-order accurate non-oscillatory schemes, I*, SIAM Journal on Numerical Analysis, v24 (1987), pp.279–309.

[38] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, *Uniformly high order essentially non-oscillatory schemes, III*, Journal of Computational Physics, v71 (1987), pp.231–303.

[39] A. Harten, S. Osher, B. Engquist and S. Chakravarthy, *Some results on uniformly high order accurate essentially non-oscillatory schemes*, Applied Numerical Mathematics, v2 (1986), pp.347–377.

[40] A. Iske and T. Soner, *On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions*, Numerische Mathematik, v74 (1996), pp.177–201.

[41] J. Jerome and C.-W. Shu, *Energy models for one-carrier transport in semiconductor devices*, in IMA Volumes in Mathematics and Its Applications, v59, W. Coughran, J. Cole, P. Lloyd and J. White, editors, Springer-Verlag, 1994, pp.185–207.

[42] J. Jerome and C.-W. Shu, *Transport effects and characteristic modes in the modeling and simulation of submicron devices*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, v14 (1995), pp.917–923.

[43] G. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, v126 (1996), pp.202–228.

[44] G. Jiang and S.-H. Yu, *Discrete shocks for finite difference approximations to scalar conservation laws*, SIAM Journal on Numerical Analysis, to appear.

[45] G. Jiang and D. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, preprint.

[46] D. A. Kopriva, *A Practical Assessment of Spectral Accuracy for Hyperbolic Problems with Discontinuities*, Journal of Scientific Computing, v2, 1987, pp.249–262.

[47] R. Krasny, *A study of singularity formation in a vortex sheet by the point-vortex approximation*, Journal of Fluid Mechanics, v167 (1986), pp.65–93.

[48] R. Krasny, *Desingularization of periodic vortex sheet roll-up*, Journal of Computational Physics, v65 (1986), pp.292–313.

[49] F. Ladeinde, E. O'Brien, X. Cai and W. Liu, *Advection by polytropic compressible turbulence*, Physics of Fluids, v7 (1995), pp.2848–2857.

[50] F. Lafon and S. Osher, *High-order 2-dimensional nonoscillatory methods for solving Hamilton-Jacobi scalar equations*, Journal of Computational Physics, v123 (1996), pp.235–253.

[51] P. D. Lax and B. Wendroff, *Systems of conservation laws*, Communications in Pure and Applied Mathematics, v13 (1960), pp.217–237.

[52] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser Verlag, Basel, 1990.

[53] X.-D. Liu, S. Osher and T. Chan, *Weighted essentially nonoscillatory schemes*, Journal of Computational Physics, v115 (1994), pp.200–212.

[54] X.-D. Liu and S. Osher, *Convex ENO high order multi-dimensional schemes without field by field decomposition or staggered grids*, preprint.

[55] A. Majda, J. McDonough and S. Osher, *The Fourier Method for Nonsmooth Initial Data*, Mathematics of Computation, v32, 1978, pp.1041–1081.

[56] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, Journal of Computational Physics, v87 (1990), pp.408–463.

[57] S. Osher, *Riemann solvers, the entropy condition, and difference approximations*, SIAM Journal on Numerical Analysis, v21 (1984), pp.217–235.

[58] S. Osher and S. Chakravarthy, *Upwind schemes and boundary conditions with applications to Euler equations in general geometries*, Journal of Computational Physics, v50 (1983), pp.447–481.

[59] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation*, Journal of Computational Physics, v79 (1988), pp.12–49.

[60] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis, v28 (1991), pp.907–922.

[61] C.S. Peskin, *Numerical analysis of blood flow in the heart*, Journal of Computational Physics, v25 (1977), pp.220–252.

[62] P. L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, Journal of Computational Physics, v43 (1981), pp.357–372.

[63] A. Rogerson and E. Meiberg, *A numerical study of the convergence properties of ENO schemes*. Journal of Scientific Computing, v5 (1990), pp.151–167.

[64] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Dynamics, Computer Vision, and Material Science*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, New York, 1996.

[65] C.-W. Shu, *TVB uniformly high order schemes for conservation laws*, Mathematics of Computation, v49 (1987), pp.105–121.

[66] C.-W. Shu, *Total-Variation-Diminishing time discretizations*, SIAM Journal on Scientific and Statistical Computing, v9 (1988), pp.1073–1084.

[67] C.-W. Shu, *Numerical experiments on the accuracy of ENO and modified ENO schemes*, Journal of Scientific Computing, v5 (1990), pp.127–149.

[68] C.-W. Shu, *Preface to the republication of "Uniform high order essentially non-oscillatory schemes, III," by Harten, Engquist, Osher, and Chakravarthy*, Journal of Computational Physics, v131 (1997), pp.1–2.

[69] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, v77 (1988), pp.439–471.

[70] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes II*, Journal of Computational Physics, v83 (1989), pp.32–78.

[71] C.-W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker, and S. Osher, *High order ENO schemes applied to two- and three- dimensional compressible flow*, Applied Numerical Mathematics, v9 (1992), pp.45–71.

[72] C.-W. Shu and Y. Zeng, *High order essentially non-oscillatory scheme for viscoelasticity with fading memory*, Quarterly of Applied Mathematics, to appear.

[73] K. Siddiqi, B. Kimia and C.-W. Shu, *Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution*, Computer Vision Graphics and Image Processing: Graphical Models and Image Processing (CVGIP:GMIP), to appear.

[74] J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, Springer-Verlag, New York, 1983.

[75] G. A. Sod, *Numerical Methods in Fluid Dynamics*, Cambridge University Press, Cambridge, 1985.

[76] J. Strikwerda, *Initial boundary value problems for the method of lines*, Journal of Computational Physics, v34 (1980), pp.94–107.

[77] M. Sussman, P. Smereka, S. Osher, *A level set approach for computing solutions to incompressible two phase flow*, Journal of Computational Physics, v114 (1994), pp.146–159.

[78] P. K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM Journal on Numerical Analysis, v21 (1984), pp.995–1011.

[79] B. van Leer, *Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method*, Journal of Computational Physics, v32 (1979), pp.101–136.

[80] F. Walsteijn, *Robust numerical methods for 2D turbulence*, Journal of Computational Physics, v114 (1994), pp.129–145.

[81] J.H. Williamson, *Low-storage Runge-Kutta schemes*, Journal of Computational Physics, v35 (1980), pp.48–56.

[82] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics, v54, 1984, pp.115-173.

[83] H. Yang, *An artificial compression method for ENO schemes, the slope modification method*, Journal of Computational Physics, v89 (1990), pp.125–160.