UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Multi-Dimensional ENO Schemes for General Geometries

Ami Harten
Sukumar R. Chakravarthy

August 1991

CAM Report 91-16

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

MULTI-DIMENSIONAL ENO SCHEMES FOR GENERAL GEOMETRIES

Ami Harten*
School of Mathematical Sciences, Tel-Aviv University
and
Department of Mathematics, UCLA

and

Sukumar R. Chakravarthy**
Rockwell Science Center, Thousand Oaks, California

Dedicated to Saul Abarbanel on
his 60^{th} Birthday

* Research supported by ONR grant N00014-86-K-0691, DARPA grant in the ACMP program and NSF grant DMS 88-11863.

** Research supported by Rockwell International Science Center's IR&D funds

1. Introduction

In this paper we generalize the ENO schemes of [8], [9] and [3]. to several space dimensions with structured and unstructured grids; it should be read in conjunction with these earlier articles.

The ENO schemes are of the form

$$v^{n+1} = A \cdot E(\tau) \cdot R(\cdot; v^n) \equiv E_h \cdot v^n \quad , \tag{1.1}$$

where v^n are cell-averages of $u(x, t_n)$, the solution at time t_n. $R(x; v^n)$ is a reconstruction procedure which produces a high-order accurate global approximation to $u(x, t_n)$ from its given cell-averages v^n; in this paper we consider R to be a polynomial function in each cell. $E(t)$ is the evolution operator of the PDE which includes the influence of boundary conditions. A is the cell-averaging operator. When we consider $E(t)$ which corresponds to a PDE with divergence free form, the scheme E_h (1.1) is automatically in conservation form no matter what is the particular shape of the cells.

The scheme (1.1) is linked to a grid in a very loose way. Rewriting (1.1) as

$$\widetilde{E}_h = R \cdot A \cdot E = P_h \cdot E \tag{1.2}$$

we can view the scheme as a composition of exact evolution E with a projection $P_h = R \cdot A$, the role of which is to project the solution into a finite-dimensional space of functions. The averaging A and the reconstruction R may even use a different set of cells. This observation is particularly useful for purposes of various grid manipulations like component grids, multigrid calculations, and time-dependent adaptive grids.

The question of stability and convergence of numerical schemes is related to the boundedness or possible growth of spurious oscillations in the computed solution. The largest source of spurious oscillations in the numerical solution is a Gibbs-like phenomenon associated with interpolation through a discontinuity; these are 0(1) oscillations with respect to refinement.

The particular form of the scheme (1.1) leaves the question of control over spurious oscillations to the design of the reconstruction R, which is a problem in the approximation of functions. The ENO schemes attempt to avoid growth of spurious oscillations by an adaptive-stencil approach, in which each cell is assigned its own stencil of cells for purposes of reconstruction. For each cell we select an interpolating stencil in which the solution is smoothest in some sense. Thus cells near a discontinuity are assigned stencils from the smooth part of the solution and a Gibbs-like phenomenon is so avoided. The term essentially non-oscillatory is used because spurious oscillations on the scale of the interpolation error in the smooth part of the solution are not ruled out. This adaptive-stencil strategy seems to ensure the stability and convergence of the scheme (1.1).

The question of accumulation of error, *i.e.,* the relation between the local truncation error and the actual error at the end of the calculation is related to the nature of the stability of the scheme. In [7] we have examined a problem in which the selection of smoothest stencil for the reconstruction leads to a scheme (1.1) which is linearly unstable in the whole interval. We have found that once the high-order derivatives have begun to oscillate and grow, the selection of stencil became erratic and this has stabilized the computation, so the calculation was convergent but with a reduced order of accuracy. Later Meiburg [13] has shown that similar loss of accuracy can occur near a point at which all the derivatives up to a certain order vanish. Shu [14] has demonstrated that this unnecessary loss of accuracy can be avoided by biasing the selection of the stencil toward a central one in the smooth part of the solution — we adopt this strategy here as well.

In Section 2 we introduce notation which enables us to describe the scheme (1.1) in the most general case. In Section 3 we describe the process of reconstruction in a given cell. After these preliminaries we begin to tackle the two really important issues of designing a procedure for selecting a stencil in general geometries and the practical aspect of an efficient implementation.

In Section 4 we go through several levels of approximation for the numerical flux, to finally arrive at a simple expression which is easy to use and is yet adequate.

In Section 5 we describe two efficient implementations of the resulting scheme.

In Section 6 we examine the schemes (1.1) with a fixed central stencil which is to be used with some form of added numerical dissipation. In this context we outline a new technique to obtain an ENO reconstruction within the fixed central stencil by hybridizing the high-order reconstruction with the first-order one. In Appendix 1 we present more details and analysis. We bring this preview of a future paper here in order to give a general picture.

In Section 7 we present the adaptive algorithm for the selection of an appropriate stencil of cells in the most general case and describe its efficient implementation as an ordered Gauss elimination with adaptive row-pivoting. At the end of this section we show how to use this procedure to automatically select fixed stencils which are either central or directionally biased. We also show how to bias the ENO stencil toward a central one.

In Section 8 we discuss special reconstruction techniques for solutions of hyperbolic systems of conservation laws. These special techniques may be needed in order to handle particularly strong interaction of discontinuities in the computed solution.

In Section 9 we describe the application of ENO schemes for rectangular grids and show the great simplification that occurs in this case. For rectangular and smoothly varying grids we use reconstruction via deconvolution in order to obtain efficient schemes that use a tensor-product of one-dimensional stencils. The third-order accurate scheme turns out to be particularly simple, and we feel that it is of immediate practical importance as the next generation to the second-order accurate TVD schemes. In Appendix 2 we describe the implementation of the third-order scheme to the solution of the Euler equations of gasdynamics.

In Section 10 we consider the question of time-integration and compare the relative merits of a method of lines approach to a local Cauchy-Kowalewski procedure of a single step. The method of lines is easier to program but more expensive to use. Thus we advocate the use of the one-step procedure for production codes whenever feasible.

In Appendix 3 we outline a new scheme which uses alternating dual grids. The two sets of grids correspond to centroids and vertices. The new scheme alternatingly computes point-values in one set of cells and cell-averages in the dual one, at no extra computational cost. The use of these two sets of values enables us to obtain a more compact reconstruction. We present this "2 for the price of 1" scheme because we feel that this is the next step in the development of high-order ENO schemes.

In writing this paper we have attempted to give as broad a picture as possible on the development of ENO schemes as we see it. The number of schemes which can be formed by different numerical fluxes, various reconstruction techniques, and methods of time-integration is huge. Consequently our numerical experimentation amounts to a mere sampling. The use of analysis for the design of these highly nonlinear schemes is still very limited and one has to rely heavily on numerical experimentation. We hope that this article will encourage others to experiment with these schemes.

We would like to point out some related work that we know of. The scheme (1.1) was originated by Godunov to design his first-order scheme [6], and was subsequently extended to second-order accuracy by van Leer [17], and improved upon by Colella and Woodward [4].

The case $r = 2$ of the ENO schemes also corresponds to second-order accurate TVD schemes; thus all finite-volume TVD schemes are related to it. In this context we would like to refer the reader to a recent work by Durlofsky, Osher, and Engquist [5] on second-order TVD schemes in a triangular grid.

We would also like to point out the work of Casper [2] on fourth-order accurate ENO implementation for rectangular and smoothly varying grids and the work of Shu and Osher [15]. The latter scheme is an efficient ENO scheme for point-values on rectangular and smoothly varying grids, and thus is not of the form (1.1). The conservation form of this scheme is obtained by a clever trick in which the numerical flux is treated as a primitive of some other function.

We would also like to refer the reader to a recent paper by Barth and Fredrickson [1] who have implemented the scheme (1.1) for unstructured triangular grids using large fixed central stencils with least squares reconstruction. Their numerical results demonstrate that a high order of accuracy can be achieved for smooth solutions even on highly irregular grids.

2. The Numerical Scheme

In this paper we consider the Initial Boundary Value Problem (IBVP) for a hyperbolic system of conservation laws in S–space dimensions:

$$u_t + \text{div}\ f(u) = 0 \quad , \quad x \in \mathcal{D} \subset \mathcal{R}^S \quad , \quad t > 0 \tag{2.1a}$$

$$u(x,0) = u_0(x) \quad , \quad x \in \mathcal{D} \tag{2.1b}$$

with given boundary conditions on $\partial\mathcal{D}$, the boundary of $\mathcal{D}$. The computational domain $\mathcal{D}$ is divided into cells C_j

$$\mathcal{D} = \bigcup C_j \quad , \quad C_j \bigcap C_k = 0 \quad . \tag{2.2a}$$

We assume that ∂C_j, the boundary of the cell C_j, is piecewise-smooth, *i.e.*,

$$\partial C_j = \bigcup_k \partial C_j^k \tag{2.2b}$$

where ∂C_j^k is smooth. Usually ∂C_j^k is linear, yet our formulations allow for nonzero curvature. We also assume that there is a refinement parameter h such that the largest sphere contained in the cells is of radius $O(h)$, and that the ratio between the largest sphere to the smallest one in the computational domain remains bounded under refinement. We denote

$$|C_j| = \int_{C_j} dV \quad , \tag{2.2c}$$

and by c_j, the centroid of the cell C_j

$$c_j = \frac{1}{|C_j|} \int_{C_j} x dV \quad . \tag{2.2d}$$

Let $\overline{u}_j$ denote the cell-average of the function $u(x)$ over C_j

$$\overline{u}_j = \frac{1}{|C_j|} \int_{C_j} u(x) dx = A(C_j)\, u \tag{2.3}$$

and denote by $A(C_j)$ the cell-averaging operator. Given cell-averages $\overline{u} = \{\overline{u}_j\}$ of $u(x)$ in $\mathcal{D}$, we denote by $R(x;\overline{u}\,)$ a reconstruction of u from $\overline{u}$ which satisfies

$$R(x;\overline{u}\,) = u(x) + O\left(h^r\right) \text{ wherever } u \text{ is smooth} \tag{2.4a}$$

$$A(C_j)\, R(\cdot;\overline{u}\,) = \overline{u}_j \text{ (conservation)} \quad . \tag{2.4b}$$

Typically R is a piecewise-polynomial function of degree $r-1$ which is discontinuous across the cell-boundary ∂C_j. Let $R_j(x;v^n)$ denote the polynomial which defines R in C_j, i.e.,

$$R(x;v^n) = R_j(x;v^n) \quad \text{for} \quad x \in C_j \quad . \tag{2.4c}$$

Let $E(t)$ denote the evolution operator of the IBVP (2.1)

$$u(\cdot,t) = E(t)u_0 \quad , \quad E(t+\tau) = E(\tau)E(t) \quad . \tag{2.5}$$

Note that $E(t)$ includes the boundary conditions on $\partial\mathcal{D}$.

We turn now to describe the numerical scheme which is an explicit method for the approximation of the cell-averages of $u(x,t)$

$$v_j^n \approx A(C_j)\, u(\cdot,t_n) \quad . \tag{2.6}$$

We initialize the computation by setting

$$v_j^0 = A(C_j)\, u_0 \quad . \tag{2.7a}$$

where u_0 is the initial value (2.1b). Given $v^n = \{v_j^n\}$ we compute v^{n+1} by

$$v_j^{n+1} = A(C_j)\, E(\tau) R(\cdot;v^n) \quad . \tag{2.7b}$$

Thus we first get a piecewise-polynomial approximation $R(x;v^n)$ to the solution $u(x,t_n)$. Then we apply $E(\tau)$ to $R(x;u^n)$, i.e., we get a solution in the small (small τ) to the IBVP

$$w_t + \text{div}\, f(w) = 0 \quad , \quad x \in \mathcal{D} \quad , \quad 0 \le t \le \tau \tag{2.8a}$$

$$w(x,0) = R(x;v^n) \quad , \quad x \in \mathcal{D} \tag{2.8b}$$

with the given boundary conditions on $\partial\mathcal{D}$.

Integrating the PDE (2.8a) over $C_j \times [0,\tau]$ and using the divergence theorem we get

$$|C_j|\left(v_j^{n+1} - v_j^n\right) + \int_0^\tau \oint_{\partial C_j} [f(w(x,t)) \cdot N]\, dS = 0 \quad . \tag{2.8c}$$

Here N is the outward normal to ∂C_j and the integration is over the boundary of the cell. Hence the application of $A(C_j)$ to the solution of (2.8a), (2.8b) at time τ is given by

$$v_j^{n+1} = v_j^n - \frac{1}{|C_j|} \int_0^\tau \oint_{\partial C_j} [f(E(t) \cdot R(\cdot; v^n)) \cdot N] \, dS \quad . \tag{2.9}$$

The above expression is an explicit scheme in conservation form. We point out that the term v_j^n in (2.9) is obtained from

$$A(C_j) w(x, 0) = A(C_j) R(x; v^n) = v_j^n \quad . \tag{2.10}$$

Thus property (2.4b) of the reconstruction is essential in order to get conservation form from (2.7b).

Since $A(C_j)$ is a positive operator and $E(\tau)$ is the exact evolution operator, the control over possible growth of oscillations in the numerical solution is applied through the reconstruction $R(x; v^n)$. In Section 7, we shall describe an Essentially Non-Oscillatory (ENO) reconstruction technique which is designed in order to achieve this goal.

In this paper we concentrate on the semi-discrete formulation of (2.9), which is obtained by dividing (2.8c) by τ and letting $\tau \to 0$. In all the cells which do not have a common side with the boundary ∂D we get

$$\frac{\partial}{\partial t} v_j = -\frac{1}{|C_j|} \sum_k \int_{\partial C_j^k} f_N^R(R_j, R_*) \, dS \quad . \tag{2.11a}$$

In (2.11) we have introduced some new notations and conventions:

$$f_N = f \cdot N \quad ; \tag{2.11b}$$

$$f_N^R(u_1, u_2) = f_N(W(0; u_1, u_2)) \quad , \tag{2.11c}$$

where $W(\xi/t; u_1, u_2)$ is the self-similar solution of the one-dimensional Riemann problem

$$w_t + \frac{\partial}{\partial \xi} f_N(w) = 0 \quad ; \tag{2.12a}$$

$$w(\xi, 0) = \begin{cases} u_1 & \xi < 0 \\ u_2 & \xi > 0 \end{cases} \quad . \tag{2.12b}$$

Observe that ξ is scalar. R_j in the term $f_N^R(R_j, R_*)$ is the polynomial (2.4c) which is evaluated at the boundary ∂C_j; R_* is the reconstruction in the cell which is in the exterior of C_j and has ∂C_j^k in common. Thus $*$ is a symbolic notation for the index of such a cell. Another innovation is the breakup of the integral over ∂C_j into its smooth pieces ∂C_j^k in (2.2b). (The superscript R in f_N^R stands for "Riemann".)

When ∂C_j^k is an element of the boundary $\partial \mathcal{D}$ we make the following substitution in (2.11)

$$\int_{\partial C_j^k} f_N^R(R_j, R_*)\, ds \rightarrow \int_{\partial C_j^k} f_N(E(0_+)R_j)\, ds \tag{2.11$'$}$$

which is computed at $\partial \mathcal{D}$ and includes the boundary conditions.

3. Reconstruction

Given cell-averages $\overline{u} = \{\overline{u}_j\}$ of $u(x)$, we describe in this section a reconstruction technique $R(x; \overline{u})$ which satisfies properties (2.4)

$$R(x; \overline{u}) = u(x) + O(h^r) \text{ wherever } u(x) \text{ is smooth (accuracy)} \quad (3.1a)$$

$$A(C_j)\, R(\cdot; \overline{u}) = \overline{u}_j \text{ (conservation)} \quad . \quad (3.1b)$$

In this paper we consider a piecewise-polynomial reconstruction, which is defined by a polynomial of degree $(r-1)$ in each of the cells. As in (2.4c) we denote the polynomial in the cell C_i by $R_i(x; \overline{u})$ and express it as a Taylor expansion around the centroid c_i:

$$R(x; \overline{u}) = R_i(x; \overline{u}) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} (x - c_i)^\ell D_\ell \quad , \quad x \in C_i \quad . \quad (3.2)$$

Here we use a multi-index notation and convention

$$\ell = (\ell_1, \cdots, \ell_s) \quad , \quad |\ell| = \ell_1 + \ell_2 + \cdots + \ell_s \quad (3.3a)$$

$$y^\ell = (y_1)^{\ell_1} \cdot (y_2)^{\ell_2} \cdots (y_s)^{\ell_s} \quad . \quad (3.3b)$$

The summation convention in $\sum_{|\ell|=k}$ stands for

$$\sum_{|\ell|=k} = \underset{\ell_1 + \ell_2 + \cdots \ell_s = k}{\sum_{\ell_1=0}^{k} \sum_{\ell_2=0}^{k} \cdots \sum_{\ell_s=0}^{k}} \quad (3.3c)$$

so that terms D_ℓ corresponding to mixed derivatives appear in the summation exactly the number of times they should. We also use the multi-index convention for differentiation

$$\frac{\partial^\ell}{\partial x^\ell} = \frac{\partial^{|\ell|}}{\partial x_1^{\ell_1} \partial x_2^{\ell_2} \cdots \partial x_s^{\ell_s}} \quad . \quad (3.3d)$$

Comparing (3.2) with the Taylor expansion of $u(x)$ around the centroid C_i

$$u(x) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} (x - c_i)^\ell \left. \frac{\partial^\ell u}{\partial x^\ell} \right|_{C_i} + O(h^r) \quad (3.4)$$

we see that the accuracy requirement (3.1a) amounts tc

$$D_\ell = \frac{\partial^\ell u}{\partial x^\ell}(c_i) + O\left(h^{r-|\ell|}\right) \quad , \quad 0 \le |\ell| \le r-1 \quad . \tag{3.5a}$$

We note that (3.5a) for $|\ell| = 0$ reads

$$D_0 = R_i(c_i;\overline{u}) = u(c_i) + O(h^r) \quad . \tag{3.5b}$$

Let $J(i)$ be a set of indices of cells which includes i. We shall refer to $J(i)$ as a stencil of cells associated with the cell i and denote the number of cells in it by $|J|$. We consider now the set of linear equations for D_ℓ, $0 \le |\ell| \le r-1$, which is obtained by taking a cell-average of the polynomial $R_i(x;\overline{u})$ in (3.2) over all the cells j in $J(i)$

$$A(C_j)R_i = \overline{u}_j \quad , \quad j \in J(i) \tag{3.6a}$$

or

$$\sum_{k=0}^{r-1}\sum_{|\ell|=k} a_{j,\ell}D_\ell = \overline{u}_j \quad , \quad j \in J(i) \tag{3.6b}$$

where

$$a_{j,\ell} = \frac{1}{k!}A(C_j)(x-c_i)^\ell = \frac{1}{k!\,|C_j|}\int_{C_j}(x-c_i)^\ell\,dV \quad . \tag{3.6c}$$

To get proper scaling let us also consider an alternative form of (3.6)

$$\sum_{k=0}^{r-1}\sum_{|\ell|=k} a'_{j\ell}D'_\ell = \overline{u}_j \tag{3.6b$'$}$$

where

$$a'_{j,\ell} = a_{j\ell}/h^{|\ell|}$$
$$D'_\ell = h^{|\ell|}D_\ell \quad .$$

Next we rewrite this system of linear equations in matrix form

$$\overline{Q}d = \overline{u} \quad . \tag{3.7a}$$

Here d is the vector of unknowns

$$d = (d_1, \cdots, d\kappa)^T \quad , \quad \kappa = \kappa(r) \tag{3.7b}$$

in which D'_ℓ are arranged in groups of equal $|\ell|$ with increasing order of $|\ell|$. $\overline{Q}$ is a matrix with $|J(i)|$ rows and $\kappa(r)$ columns; let us agree that the first row always corresponds to $j = i$. The entries of $\overline{Q}$ are the proper rearrangement of the coefficients $a'_{j\ell}$ in (3.6b$'$). $\overline{u}$ is the vector of the given cell-averages

$$\overline{u} = \left(\overline{u}_1, \cdots, \overline{u}_{|J|}\right) \quad , \quad \overline{u}_1 = \overline{u}_i \quad . \tag{3.7c}$$

When we apply the cell-averaging operator to $u(x)$ in (3.4) we get

$$\begin{aligned} \overline{u}_j = A(C_j)\,u(x) &= \sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} \frac{\partial^\ell u}{\partial x^\ell}(c_i) + O\left(h^r\right) \\ &= \sum_{k=0}^{r-1} \sum_{|\ell|=k} a'_{j,\ell} \left[h^{|\ell|} \frac{\partial^\ell u}{\partial x^\ell}(c_i)\right] + O\left(h^r\right) \quad . \end{aligned} \tag{3.8a}$$

Let us denote by d^E the vector (3.7b) in which we substitute D'_ℓ by $h^{|\ell|}\frac{\partial^\ell u}{\partial x^\ell}(c_i)$; clearly

$$\overline{Q} d^E = \overline{u} + O\left(h^r\right) \quad . \tag{3.8b}$$

Subtracting (3.8b) from (3.7a) we get

$$\overline{Q}\left(d - d^E\right) = O\left(h^r\right) \quad . \tag{3.8c}$$

Let us denote symbolically by $\overline{Q}^{-1}$ the solution procedure we are going to use in order to obtain d from $\overline{u}$, i.e.,

$$d = \overline{Q}^{-1}\overline{u} \quad . \tag{3.9a}$$

If

$$\| \overline{Q}^{-1} \| \leq \text{ constant as } h \to 0 \tag{3.9b}$$

then it follows from (3.8c) that

$$\| d - d^E \| = O\left(h^r\right) \quad . \tag{3.9c}$$

This, due to the scaling (3.6b$'$), implies the accuracy requirements (3.5a) and (3.1a).

We turn now to consider the requirement (3.9b) from the point of view of the stencil $J(i)$. Clearly we need

$$\text{rank } \overline{Q} = \kappa(r) \quad , \tag{3.10a}$$

i.e., the number of linearly independent equations in (3.6) should be the same as the number of terms in the Taylor expansion (3.2). This, of course, implies that $|J(i)|$, the number of cells in the stencil $J(i)$, should be at least $\kappa(r)$

$$|J(i)| \geq \kappa(r) \quad . \tag{3.10b}$$

Moreover, to assure that there are enough linearly independent equations in the stencil, the stencil should be large enough in all spatial directions, so that all derivatives can be properly approximated to the required order of accuracy (3.5a).

When the computational cells are defined by some structured grid, it is possible to predetermine proper stencils with $|J(i)| = \kappa(r)$. However, for completely unstructured grids it seems wise to start with very large stencils,

$$|J(i)| >> \kappa(r) \tag{3.11a}$$

and to either use a least-squares approach, *i.e.*, to solve the $\kappa \times \kappa$ system

$$\overline{Q}^T \overline{Q} d = \overline{Q}^T \overline{u} \tag{3.11b}$$

thus minimizing

$$\| \overline{Q} d - \overline{u} \|_{L_2} \quad , \tag{3.11c}$$

or to use techniques that select $\kappa(r)$ linearly independent equations out of the many available in the stencil. In this latter category we can use an ordered Gaussian elimination with row pivoting (of the type that is described in Section 7 for ENO reconstruction) or to get the exact number of linearly independent equations needed by grouping several of the cells in $J(i)$ into a single "super-cell" and replace the several equations for the individual cells by a single one for the average over the super-cell. In all such techniques the process of selection should be ordered so as to give preference to closest neighbors, *i.e.*, the stencil should be as centered around C_i as permissible by the various constraints.

We turn now to examine the conservation property (3.1b), and observe that this is exactly the equation for $j = i$ in the system of linear equations (3.6) (which by agreement is the first equation in (3.7)). It follows that if we use the above selection procedures, this property is satisfied automatically. However, if we use the least-squares approach (3.11), we have to replace the first element d_1 in the computed solution by

$$d_1 - \overline{u}_i - \sum_{k=1}^{\kappa(r)} \overline{q}_{1,k} d_k \quad , \tag{3.11d}$$

in other words, we make sure that the first equation is satisfied exactly. In this case, the first equation may first be eliminated from every other equation forming matrix $\tilde{Q}$ of rank $\kappa(r) - 1$ and a reduced set of equations may be solved before applying (3.11d).

$$\tilde{Q}^T \tilde{Q} \tilde{d} = \tilde{Q}^T \tilde{u}. \tag{3.11b$'$}$$

More generally, if we compute approximations D_ℓ to the derivatives $\frac{\partial^\ell u}{\partial x^\ell}$ for $|\ell| \geq 2$ which satisfy the accuracy requirement (3.5a), we define

$$D_0 = \overline{u}_i - \sum_{k=2}^{r-1} \sum_{|\ell|=k} a_{0,\ell} D_\ell \quad . \tag{3.12}$$

This ensures that the resulting reconstruction (3.2) satisfies the conservation requirement as well as the accuracy requirement in (3.1).

We point out that the summation in (3.12) starts with $k = 2$. The reason for that is that

$$a_{i,\ell} = A\left(C_i\right) \cdot \left(x - c_i\right)^\ell = 0 \text{ for } |\ell| = 1 \tag{3.13a}$$

due to the fact that the centroid c_i (2.2d) is defined by

$$c_i = A\left(C_i\right) x \quad .$$

It follows therefore that the cell-average is an $O\left(h^2\right)$ approximation to the point-value at the centroid

$$\overline{u}_i - u\left(c_i\right) = O\left(h^2\right) \quad . \tag{3.13b}$$

We remark that the coefficients in the $O\left(h^2\right)$ term above are $a_{i,\ell}$ for $|\ell| = 2$, which depend on the shape of the cell. Consequently, numerical differentiation of $\overline{u}_i$ (instead of the point values $u\left(c_i\right)$) can give an $O\left(h^2\right)$ approximation to the derivatives of u, only in the case of a smoothly varying structured grid.

We end this section with the analogous problem for point-values: Given point-values of $u(x)$ in the centroids, $u_j = u\left(c_j\right)$ for $j \in J(i)$, find an r^{th} order polynomial approximation $I(x;0)$ to $u(x)$ in C_i. Rewriting this polynomial as a Taylor expansion around c_i

$$I(x;u) = \sum_{j=0}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} \left(x - c_i\right)^\ell D_\ell \quad , \tag{3.14}$$

we consider the system of linear equations for D_ℓ

$$I\left(c_j;u\right) = u_j \quad , \quad j \in J(i) \tag{3.15a}$$

or

$$\sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} D_\ell = u_j \quad , \quad j \in J(i) \tag{3.15b}$$

where

$$a_{j,\ell} = \frac{1}{k!} \left(c_j - c_i\right)^\ell \quad . \tag{3.15c}$$

From this point on we proceed as in the reconstruction procedure to rewrite (3.15) in the matrix form

$$Qd = u \quad , \tag{3.16}$$

which can be solved either by a least-squares approach

$$Q^T Q d = Q^T u \tag{3.17}$$

or by the same selection techniques that were mentioned before.

We observe that $a_{j,\ell}$ for the reconstruction (3.6c) is the cell average of $(x - c_i)^\ell$, while $a_{j,\ell}$ in (3.15c) is the point-value of it. Consequently,

$$\overline{Q} \to Q \text{ as } h \to 0 \quad . \tag{3.18}$$

Hence for h sufficiently small we can use the set of cells which yield linearly independent equations in Q for $\overline{Q}$ and vice versa.

4. Numerical Flux

The semi-discrete scheme (2.11) can be rewritten in the form

$$\frac{\partial v_j}{\partial t} = -\sum_k \frac{|\partial C_j^k|}{|C_j^k|} \overline{f}_{j,*}^k \tag{4.1a}$$

where the summation in k corresponds to the partition (2.2b) of the cell boundary ∂C_j, and

$$\overline{f}_{j,*}^k = \frac{1}{|\partial C_j^k|} \int_{\partial C_j^k} f_N^R(R_j, R_*)\, dS \quad . \tag{4.1b}$$

Here we use the notation

$$|\partial C_j^k| = \int_{\partial C_j^k} dS \quad . \tag{4.2}$$

We refer to the term $\overline{f}_{j,*}^k$ in (4.1b) as the numerical flux at ∂C_j^k; it expresses the average normal flux across ∂C_j^k in the solution of the IBVP (2.8). In the following we propose various approximations to (4.1b) which allow either a simplified or a computationally more efficient algorithm, or both. To simplify our notation, we shall retain the notation $\overline{f}_{j,*}^k$ to the approximate as well as the exact numerical flux.

The most straightforward simplification is to replace the integral in (4.1b) by an appropriate numerical quadrature,

$$\overline{f}_{j,*}^k = \sum_m \alpha_m f_N^R(R_j(x_m; v^n), R_*(x_m; v^n)) \tag{4.3}$$

where x_m are the quadrature points on ∂C_j^k, and α_m are the corresponding quadrature coefficients. The quadrature formula should be exact for a polynomial of degree $(r-1)$ or more. Gaussian quadrature seems to be particularly attractive in this context.

The next level of simplification is to replace the flux of the exact solution to the Riemann problem (2.11c) by a simpler approximate one. We observe that $W(x/\ell; u_1, u_2)$, the solution to the Riemann problem (2.12), is a Lipschitz continuous function of u_1 and u_2. Consequently,

$$f_N^R(u_1, u_2) = \frac{1}{2}[f_N(u_1) + f_N(u_2)] + O(|u_2 - u_1|) \quad . \tag{4.4a}$$

Since

$$|R_j(x_m;v^n) - R_*(x_m;v^n)| = O\left(h^r\right) \tag{4.4b}$$

in regions of smoothness, we can replace (2.11c) by

$$f_N^R\left(u_1,u_2\right) = \frac{1}{2}\left[f_N(u_1) + f_N(u_2) - \nu(u_1,u_2)(u_2-u_1)\right] \tag{4.5a}$$

where ν is bounded, without adversely affecting the formal order of accuracy of the scheme. Again to simplify notation we retain f_N^R for the approximate fluxes in (4.5a). The following expressions for $\nu\left(u_1,u_2\right)$ are suitable:

$$\nu\left(u_1,u_2\right) = |A_N(u_1,u_2)| \tag{4.5b}$$

where

$$A_N(u,u) = \frac{\partial f_N}{\partial u} \quad .$$

$A_N\left(u_1,u_2\right)$ can be taken as $A_N\left(\hat{u}\right)$, where $\hat{u}$ is some average of u_1 and u_2.

$$\nu\left(u_1,u_2\right) = |a_N(u_1,u_2)| \tag{4.5b$'$}$$

where a_N is the maximal eigenvalue of $A_N\left(u_1,u_2\right)$.

$$\nu\left(u_1,u_2\right) = \max_{\mathcal{D}}\left|a_N(v_j^n)\right| \quad . \tag{4.5b$''$}$$

The last quantity is constant in space during a time-step, and is computed anyway in order to calculate the permissible time-step under a CFL restriction.

We remark that in the context of first order schemes, (4.5b) corresponds to Roe's scheme, (4.5b$'$) corresponds to Rusanov's scheme, and (4.5b$''$) is the Lax-Friedrichs scheme.

The following simplification allows us to compute a single "Riemann solver" per side, rather than the number required by the quadrature formula in (4.3):

$$\overline{f}_{j,*}^k = \frac{1}{2}\left[\hat{f}_j^k + \hat{f}_*^k - \nu\left(\hat{u}_j^k,\hat{u}_*^k\right)\left(\hat{u}_*^k - \hat{u}_j^k\right)\right] \quad . \tag{4.6a}$$

Here

$$\hat{u}_j^k \approx \frac{1}{\left|\partial C_j^k\right|} \int_{\partial C_j^k} R_j dS \quad , \quad \overline{u}_*^k \approx \frac{1}{\left|\partial C_j^k\right|} \int_{\partial C_j^k} R_* dS \tag{4.6b}$$

$$\hat{f}_j^k \approx \frac{1}{\left|\partial C_j^k\right|} \int_{\partial C_j^k} f_N\left(R_j\right) dS \quad , \quad \hat{f}_*^k \approx \frac{1}{\left|\partial C_j^k\right|} \int_{\partial C_j^k} f_N\left(R_*\right) dS \quad . \tag{4.6c}$$

Both $\hat{u}_j^k$ and $\hat{f}_j^k$ can be computed by the same quadrature formula as in (4.3), i.e.,

$$\hat{u}_j^k = \sum_m \alpha_m R_j\left(x_m; v^n\right) \tag{4.7a}$$

$$\hat{f}_j^k = \sum_m \alpha_m f_N\left(R_j(x_m; v^n)\right) \quad . \tag{4.7b}$$

We observe that when R_j is given in its Taylor series form (3.2), $\hat{u}_j^k$ can be computed analytically by

$$\hat{u}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|=k} b_\ell^k D_\ell \quad , \tag{4.8a}$$

$$b_\ell^k = \frac{1}{\left|\partial C_j^k\right| |\ell|!} \int_{\partial C_j^k} \left(x - c_j\right)^\ell ds \tag{4.8b}$$

which is usually more economical than the computation in (4.7a). The expression for $\nu\left(u_1, u_2\right)$ in (4.6a) is the same as in (4.5).

Further simplification is obtained when a Taylor expansion of $f\left(R_j(x_j u^n)\right)$ around c_j is available to us in the form

$$f\left(R_j(x; u^n)\right) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} \left(x - c_i\right)^\ell F_\ell + O\left(h^r\right) \tag{4.9a}$$

with

$$F_\ell = \frac{\partial^\ell}{\partial x^\ell} f(c_i) + O\left(h^{r-|\ell|}\right) \quad . \tag{4.9b}$$

Then, as in (4.8),

$$\hat{f}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|=k} b_\ell^k \left(F_\ell \cdot N_j^k\right) \tag{4.10}$$

where N_j^k is the outward normal to ∂C_j^k. Note that we have taken only a single value of the normal, thus for boundaries with nonzero curvature the RHS of (4.10) has to be modified accordingly.

We remark that in the constant coefficient case, all the schemes with the same $\nu(u_1, u_2)$ are identical to each other, and the scheme with $\nu(u_1, u_2) = |A_N(u_1, u_2)|$ is identical to the original scheme (2.11).

5. Algorithmic Considerations

In this section we consider the simplest scheme of the previous section

$$\frac{\partial v_j}{\partial t} = -\sum_k \frac{|\partial C_j^k|}{|\partial C_j^k|} \overline{f}_{j,*}^k \quad , \tag{5.1a}$$

$$\overline{f}_{j,*}^k = \frac{1}{2}\left[\hat{f}_j^k + \hat{f}_*^k - \nu(\hat{u}_j^k, \hat{u}_*^k)(\hat{u}_*^k - \hat{u}_j^k)\right] \tag{5.1b}$$

where

$$\hat{u}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|=k} b_\ell^k D_\ell \tag{5.1c}$$

$$\hat{f}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|=k} b_\ell^k \left(F_\ell \cdot N_j^k\right) \tag{5.1d}$$

and discuss two versions which differ in the way in which F_ℓ are obtained.

In the first version, we start by selecting a stencil $J(i)$ to the cell C_i, and compute the reconstruction $R_i(x; v^n)$ in its Taylor series form (3.2). Once we have calculated $\{D_\ell\}$ we evaluate $\{F_\ell\}$ by using analytic expressions for

$$F_\ell = \frac{\partial^\ell}{\partial x^\ell} f\left(R_i(x; v^n)\right)\Big|_{x=c_i} \quad . \tag{5.2}$$

Still within the cell C_i, we proceed to calculate $\hat{u}_i^k$ and $\hat{f}_i^k$ by (5.1c),(5.1d) respectively for all sides k.

After doing so for all cells C_i in $\mathcal{D}$, we sweep over all the boundaries ∂C_i^k to compute the numerical flux $\overline{f}_{i,*}^k$ by (5.1b). Finally, we go over all the cells C_i to evaluate $\frac{\partial v_i}{\partial t}$ by (5.1a).

This version makes computational sense when the expressions in (5.2) are relatively simple, which is the case for the Euler equations of compressible perfect gas (see Appendix 2). The main advantage of this algorithm is that most of the computational work is done within the cell, with minimal communication with other cells. Hence it seems attractive for unstructured grids, especially for parallel computers.

We turn now to describe the second version which offers a low operational count at the cost of a larger storage requirement and more communication between cells. The main difference from the previous version is that F_ℓ are computed from interpolation formula.

We start this algorithm by selecting a stencil $J(i)$ for the cell C_i and compute the reconstruction $R_i\left(x; v^n\right)$. From this reconstruction we compute the point-values at the centroid of the cell

$$u_i^n = R_i\left(c_i; v^n\right) = v_i^n - \sum_{k=2}^{r-1} \sum_{|\ell|=k} a_{i,\ell} D_\ell \tag{5.3a}$$

and

$$f_i^n = f\left(u_i^n\right) \quad . \tag{5.3b}$$

Next we sweep again over all cells to compute

$$F_\ell = \frac{\partial^\ell}{\partial x_\ell} I\left(x; f^n\right)\big|_{x=c_i} \tag{5.4a}$$

$$\widetilde{D}_\ell = \frac{\partial^\ell}{\partial x^\ell} I\left(x; u^n\right)\big|_{x=c_i} \quad . \tag{5.4b}$$

Still within the cell, we calculate $\hat{u}_i^k$ and $\hat{f}_i^k$ by (5.1c) and (5.1d). From this point on we proceed exactly as in the previous algorithm to compute the numerical fluxes $\overline{f}_{i,*}^k$ by (5.1b) and in another sweep to compute the RHS of (5.1a).

The notation $I\left(x; f^n\right)$ stands for the point-value approximation (3.14), and $\frac{\partial^\ell}{\partial x^\ell} I\left(x; f^n\right)$ is the appropriate coefficient D_ℓ, which is computed from the linear system (3.16). We point out that the same set of cells is used for both the reconstruction step and the interpolation step (see the discussion at the end of Section 3); this is of particular importance for ENO schemes with adaptive selection of stencils. Another observation is that $\hat{u}_j^k$ can be computed directly from the reconstruction using D_ℓ rather than $\widetilde{D}_\ell$.

This second scheme has a low operational count, especially for structured grids where the computation of D_ℓ and F_ℓ is accomplished by predetermined finite difference expressions. The operational count for this scheme is 1 selection of stencil and 1 $f(u)$ evaluation per cell, and 1 "Riemann solver" per side. We shall come back to this scheme in Section 9 where we discuss schemes for a rectangular grid.

We note that even in the constant coefficient case, the two versions are not identical — the second version uses many more cells than the first one.

6. Fixed-Stencil Schemes

In this section we consider the schemes described in Sections 4 and 5, with a fixed central stencil $J(i)$. The stencil is assigned to the cell on the basis of geometrical considerations alone so that it will be as centered around the cell as possible. Thus in the constant coefficient case

$$f(u) = au \tag{6.1a}$$

the scheme is a linear operator (unlike the ENO schemes of the next section where the stencil assigned to the cell depends on the solution).

We note that although the stencil for the reconstruction step is centered, the resulting scheme (2.7) is upwind biased: In the constant coefficient case

$$E(\tau)R(x;v^n) = R(x - a\tau; v^n) \tag{6.1b}$$

and therefore

$$v_j^{n+1} = A(C_j)R(x - a\tau; v^n) \quad . \tag{6.1c}$$

Based on analysis of some simple cases and some numerical experiments, we feel it is safe to conjecture that the scheme (2.7) with a centered stencil $J(i)$ is L_2–stable in the constant coefficient case. We also note that the choice of centered stencil results in the most accurate reconstruction. Therefore such schemes are excellent numerical solvers for problems with smooth solutions or with weak shocks; in the latter case one can add some form of high-order numerical dissipation to the scheme, e.g., one can add dissipation in the form of a filter [12].

Clearly when the stencil $J(i)$ contains a discontinuity of the solution, the reconstruction R_i has spurious oscillations due to a Gibbs-like phenomenon. In this case we can either select to use a different stencil which does not include a discontinuity — this technique will be described in the next section, or to modify the reconstruction within the same stencil. To do so we go back to the rather old ideas of hybrid schemes ([10],

[11]) except that here we hybridize the reconstructions rather than numerical fluxes. Symbolically this can be written as

$$R_i(x; v^n) = \theta_i v_i^n + (1 - \theta_i) R_i^r(x; v^n) \tag{6.2a}$$

where R_i^r, $r \geq 2$ is the high-order reconstruction (3.2), and v_i^n corresponds to the piecewise constant reconstruction

$$R_i(x; v^n) = v_i^n \quad , \quad x \in C_i \tag{6.2b}$$

which is monotone.

The automatic switch θ_i has the properties

$$0 \leq \theta_i \leq 1 \quad , \tag{6.3a}$$

$$\theta_i \approx 1 \quad \text{when } J(i) \text{ contains a discontinuity} \quad , \tag{6.3b}$$

$$\theta_i = O\left(h^{r-1}\right) \quad \text{when the solution is smooth in } J(i) \quad . \tag{6.3c}$$

Rewriting (6.2a) as

$$R_i(x; v^n) = R_i^r(x; v^n) + \theta_i \left[v_i^n - R_i^r(x; v^n)\right] \tag{6.4a}$$

we see that since

$$v_i^n - R_i^r(x; v^n) = O(h) \tag{6.4b}$$

where the solution is smooth, we get from (6.3c) that the formal order of accuracy is preserved.

Preliminary analysis for rectangular grids shows that one can construct an automatic switch which satisfies properties (6.3), and that the resulting scheme is essentially non-oscillatory. We refer the reader to Appendix 1, where we present analysis for the one-dimensional case.

In this paper we concentrate on deriving ENO schemes by an adaptive stencil technique which is described in the next section.

7. Adaptive-Stencil Schemes

In this section we consider an adaptive reconstruction technique of the form (3.2)

$$R(x;\overline{u}) = R_i(x;\overline{u}) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} (x-c_i)^{\ell} D_{\ell} \quad , \quad x \in C_i \tag{7.1a}$$

$$D_{\ell} = \frac{\partial^{\ell} u}{\partial x^{\ell}}(c_i) + O\left(h^{r-|\ell|}\right) \tag{7.1b}$$

$$D_0 = R_i(c_i;\overline{u}) = \overline{u}_i - \sum_{k=2}^{r-1} \sum_{|\ell|=k} a_{i,\ell} D_{\ell} \tag{7.1c}$$

where $a_{i,\ell} = \frac{1}{|\ell|!} A(C_i)(x-c_i)^{\ell}$.

The main objective in this adaptive reconstruction is to make sure that if C_i lies in the smooth part of the function, then all the approximations to derivatives D_{ℓ} are computed also from the smooth part of the function. This guarantees that

$$|R(x;\overline{u}) - u(x)| = O(h^r) \text{ for } x \in C_i \tag{7.2}$$

for all cells C_i that do not contain a discontinuity and thus a Gibbs-like phenomenon of spurious oscillations of size $O(1)$ in the neighborhood of a discontinuity is avoided. We remark that analysis of the one-dimensional case in [8] shows that the reconstruction is generically monotone in the cell that contains a discontinuity.

In the following we present two techniques for the selection of appropriate stencils, which generalize the two algorithms that have been used in the one-dimensional case in the earlier development of the ENO schemes. The first approach is to consider several candidate stencils, and to select the one in which the function is smoothest. The second approach is hierarchical: We begin with the i^{th} cell, and at each step of the algorithm we add another cell to the existing stencil for the computation of an additional derivative.

In the first approach we have to specify a set **J** of candidate stencils. Since the choice of a central stencil is best from the point of view of accuracy and stability, let us agree that $\mathbf{J}$ should always include a central stencil $J_c(i)$, which should be selected if the function to be reconstructed is smooth in it. For reasons of computational efficiency we would like the number of stencils in $\mathbf{J}$, which we denote by $|\mathbf{J}|$, to be as small as possible. On the other hand $\mathbf{J}$ should include enough directionally biased stencils so that we can select a stencil in which the function is smooth, no matter where we place a discontinuity in the vicinity of C_i. It seems to us that for each side ∂C_i^k of the cell we need to have a directionally biased stencil which lies in a conical section, the apex of which is the cell C_i, and its axis of symmetry is the *inward* normal to ∂C_i^k. This stencil widens as we go away from the cell C_i, and the cone is truncated once it contains enough cells to determine all the required derivatives. Let us denote by $J_k(i)$ the directionally biased stencil corresponding to the side ∂C_i^k, and let K denote the number of sides in ∂C_i, so the minimal set of candidate stencils seems to be

$$\mathbf{J} = \{J_c(i), J_1(i), \cdots, J_K(i)\} \quad . \tag{7.3}$$

Thus for triangles we use four stencils which are shown schematically in Figure 1.

Let us denote by D_ℓ^k , $|\ell| \geq 1$ the approximation to derivatives (7.1b) which is obtained from the directionally biased stencils, and by D_ℓ^c the values that correspond to the centered stencil $J_c(i)$. Let σ_r

$$\sigma_r = \sum_{|\ell|=r-1} |D_\ell| \quad , \tag{7.4a}$$

serve as a measure of smoothness of the function in the stencil. Clearly σ is large when the stencil contains a discontinuity; if the function is smooth in the stencil then

$$\sigma_r \approx \sum_{|\ell|=r-1} \left| \frac{\partial^\ell u}{\partial x^\ell}(c_i) \right| \quad . \tag{7.4b}$$

Therefore we select the stencil in which σ_r is minimal, giving preference to the central stencil. Following Shu [14] this can be done by using $\alpha\sigma^C$, $\frac{1}{2} \leq \alpha \leq 1$ for the central stencil in the comparison.

Next we would like to describe a variant of the above technique, which selects $\{D_\ell\}$ in (7.1b) term by term, without selecting a single stencil for the reconstruction. This

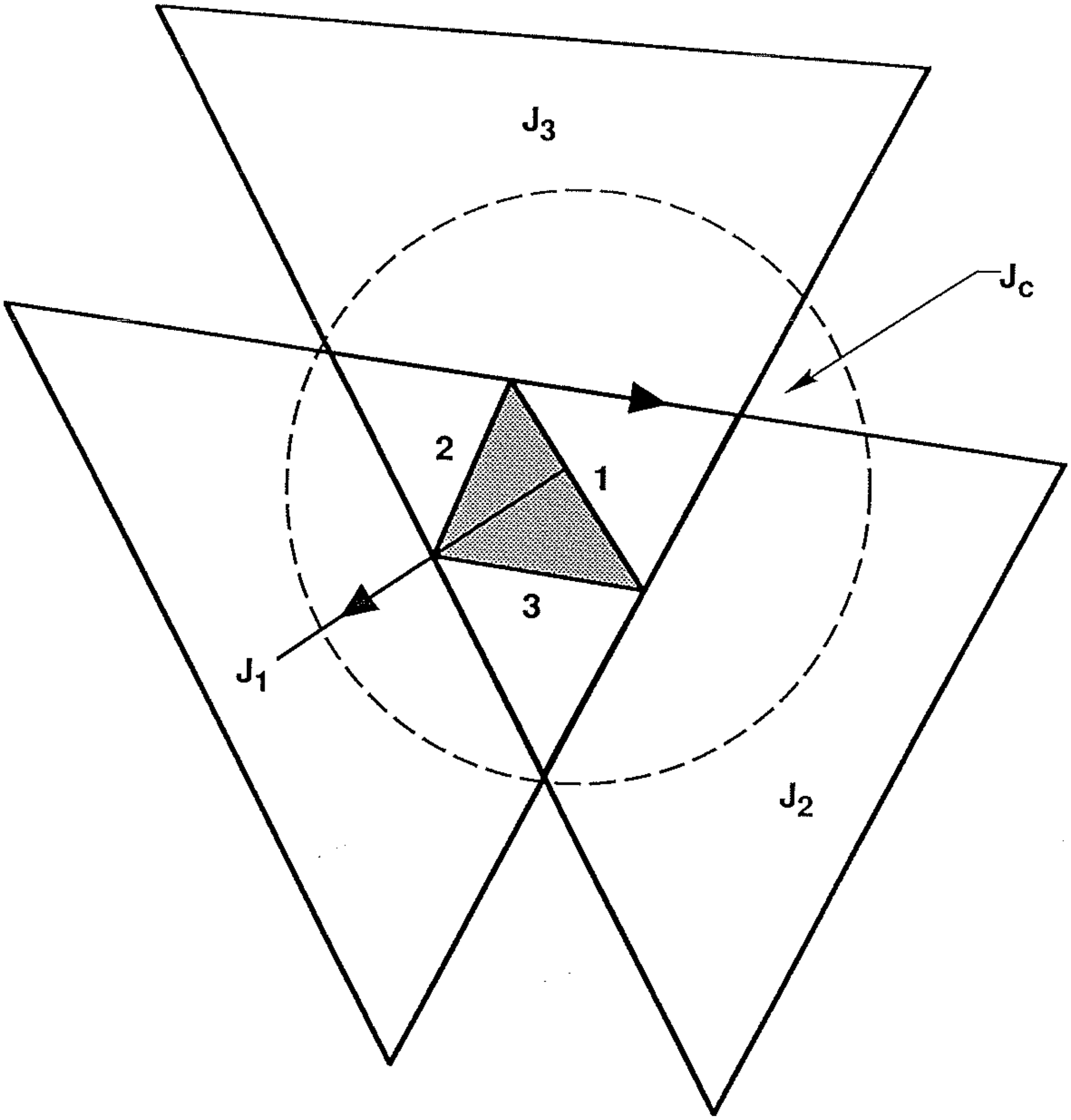

Fig. 1. Candidate stencils for ENO schemes on triangles

is accomplished by

$$\widetilde{D}_\ell = m_\alpha\left(D^1_\ell, \cdots, D^k_\ell; D^c_\ell\right) \quad , \quad 1 \le |\ell| \le r-1 \; , \; \frac{1}{2} \le \alpha \le 1 \tag{7.5a}$$

$$\widetilde{D}_0 = \overline{u}_i - \sum_{k=2}^{r-1} \sum_{|\ell|=k} a_{i,\ell} \widetilde{D}_\ell \tag{7.5b}$$

$$R_i(x; \overline{u}\,) = \widetilde{D}_0 + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{|\ell|=k} (x - c_i)^\ell \, \widetilde{D}_\ell \quad . \tag{7.5c}$$

Here $m_\alpha\left(x_1, \cdots, x_k; y\right)$ is a modified minmod function which is defined by

$$m_\alpha\left(x_1, \cdots, x_k; y\right) = \begin{cases} x_i & \text{if } m = |x_i| \\ y & \text{if } m = \alpha|y| \end{cases} \tag{7.5a$'$}$$

where

$$m = \min\left\{\left|x_i\right|, \cdots, \left|x_k\right|, \alpha|y|\right\}$$

We turn now to describe the hierarchical algorithm which generalizes the one-dimensional algorithm in [8] to multidimensions and general geometries.

We denote by $J_m(i)$ the stencil of m cells which is assigned to C_i at the m^{th} step of this algorithm

$$J_m(i) = \{i_1, \cdots, i_m\} \tag{7.6a}$$

and by J^*_m the indices of all the side-neighbors of $S_m = \bigcup\limits_{j \in J_m} C_j$, i.e., the cells in the exterior of S_m which share a common side with ∂S_m.

With $J_m(i)$ we associate an invertible system of m linear equations

$$\overline{Q}^{(m)} d^{(m)} = \overline{u}^{(m)} \quad , \tag{7.6b}$$

$$\text{rank}\left(\overline{Q}^{(m)}\right) = m \tag{7.6c}$$

which is obtained from writing the m relations

$$A\left(C_j\right) R_i\left(x; \overline{u}\,\right) = \overline{u}_j \quad , \quad j \in J_m(i) \tag{7.7}$$

in the form (3.7)

$$\overline{Q}d = \overline{u} \tag{7.8a}$$

where Q is an $m \times \kappa(r)$ matrix and $d = \left(d_1, \cdots, d_{\kappa(r)}\right)^T$. We take $\overline{Q}^{(m)}$ to be the first m columns of $\overline{Q}$, $d^{(m)}$ is the first m components of d

$$d^{(m)} = (d_1, \cdots, d_m)^T \tag{7.8b}$$

and

$$\overline{u}^{(m)} = (\overline{u}_{i_1}, \cdots, \overline{u}_{i_m})^T \quad . \tag{7.8c}$$

We begin the algorithm by setting

$$i_1 = i \quad , \quad J_1 = \{i\} \quad ; \tag{7.9a}$$

then for $m = 1$, $\kappa(r) - 1$ we define

$$J_{m+1}(i) = J_m(i) \bigcup \{i_{m+1}\} \quad , \quad i_{m+1} \in J_m^* \quad . \tag{7.9b}$$

In order to select i_{m+1} we consider the candidate stencils

$$J_{m+1}^{(j)} = \{i_1, \cdots, i_m, j\} \quad , \quad j \in J_m^* \tag{7.10a}$$

and the asociated systems of $(m+1)$ linear equations of the form (7.6b) corresponding to them

$$\overline{Q}_j^{(m+1)} d_j^{(m+1)} = \overline{u}_j^{(m+1)} \quad , \quad j \in J_m^* \quad . \tag{7.10b}$$

Next we compute d_j^{m+1} whenever the corresponding system is invertible. We take c_{m+1} to be the j for which $\left|d_j^{m+1}\right|$ is minimal, *i.e.*,

$$\left|d_{i_{m+1}}^{(m+1)}\right| = \min_{j \in J_m^*} \left|d_j^{(m+1)}\right| \quad . \tag{7.10c}$$

Here $| \quad |$ denotes some weighted norm or seminorm.

After completing this do loop, we get the desired reconstruction (7.1) by setting

$$d = d^{(\kappa(r))} \quad . \tag{7.10d}$$

We observe that J_m^* includes all the side neighbors of $J_m(i)$ and thus span all possible directions; therefore it is possible to find j in J_m^* so that

$$\text{rank}\left(\overline{Q}_j^{(m+1)}\right) = m+1 \quad .$$

Furthermore, if $J_m(i)$ is in the smooth part of the function $u(x)$ and J_m^* includes a cell in the smooth part of $u(x)$ which is directionally suitable, then $J_{m+1}(i)$ will also be in the smooth part.

In the following we propose an efficient implementation of this algorithm, which can be viewed as an <u>ordered</u> Gauss elimination with <u>adaptive</u> row-pivoting.

In this implementation we work with the original form of the equations (7.8a), which are ordered as follows: In the m^{th} step of the algorithm, the first m equations correspond to $i_1, \cdots, i_m$ in $J_m(i)$ (7.6a). These are followed by all the equations corresponding to J_{m-1}^*. We assume that at the beginning of the step the first m equations are in upper triangular form, and that $d_1, \cdots, d_{m-1}$ have been eliminated from the rest of the equations. As is customary in Gauss elimination we add the RHS of the equations as an extra column in an extended matrix. Thus the extended matrix $\widehat{Q}^{(m)}$ at the beginning of the m^{th} step is as follows

$$\widehat{Q}^{(m)} = \left[\begin{array}{cccc} \hat{q}_{11}^{(m)} & & & \vdots \\ & \ddots & & \vdots \\ 0 & & \ddots & \vdots \\ & & \hat{q}_{mm}^{(m)} & \vdots \\ 0 & & \vdots & \vdots \\ & & \vdots & \vdots \end{array}\right] \begin{array}{c} i_1 \\ \vdots \\ \vdots \\ i_m \\ J_{m-1}^* \\ \end{array} \tag{7.11}$$

We start this algorithm by setting $\widehat{Q}^{(1)}$ to be the $1 \times (\kappa+1)$ matrix which corresponds to the equation for the cell C_i.

Given $\widehat{Q}^{(m)}$ we show now how to evaluate i_{m+1} and $\widehat{Q}^{(m+1)}$. We begin by adding to $\widehat{Q}^{(m)}$ the equations for all the side-neighbors of C_{i_m}, thus completing the set of extra equations to that of J^*_m. (Note that duplicity of equations is possible, but this does not interfere with the execution of the algorithm.) We use the diagonal elements in the first $(m-1)$ equations to eliminate $d_1, \cdots, d_{m-1}$ from the additional equations. The next stage of the algorithm is to use $\hat{q}^{(m)}_{m,m}$ to eliminate d_m from the $(m+1)^{\text{th}}$ equation and on. Doing so, we are now in a position to form $\widehat{Q}^{(m)}_j$, which is an upper triangular form of the system (7.10b), by moving any of the equations for $j \in J^*_m$ to the $(m+1)^{\text{th}}$ row. $d^{(m+1)}_j$ in (7.10b) can now be computed by back substitution with the appropriate RHS which is stored in the $(\kappa+1)^{\text{th}}$ column of the matrix.

After selecting i_{m+1} by (7.10c), we form the matrix $\widehat{Q}^{(m+1)}$ by assigning the equation of the cell i_{m+1} to the $(m+1)^{\text{th}}$ row. This completes the m^{th} step of the algorithm.

Once we have computed $\widehat{Q}^{(\kappa)}$, the required solution d is obtained by back substitution.

We remark that if we take in (7.10c)

$$\left| d^{(m+1)} \right| = |d_{m+1}| \tag{7.12}$$

i.e., the semi-norm which assigns to a vector the absolute value of its last component, then there is no need to back substitute in every step of the algorithm. Our numerical experiments seem to indicate that this is a viable practice.

We have termed this procedure as ordered Gauss elimination with adaptive row-pivoting for the following reasons: "ordered" – because we feed in equations, *i.e.*, cells, in a particular order; "adaptive" – because the criterion for selecting the clustered cells is the minimization (7.10c) of the derivatives of the reconstruction at the centroid of the i^{th} cell. Within the context of adaptive-stencil schemes this is done anew at each time in each cell and the result depends on the local smoothness of the reconstructed function.

In the following we show how to use this procedure in order to construct fixed stencils which minimize the reconstruction (or interpolation) error of a given function.

This is done by setting the problem so that the vector d in (7.6) and (7.8) is the reconstruction error of the function and its derivatives at the centroid.

Let us denote by $f(x)$ a given function and denote its derivatives at the centroid of the i^{th} cell by f^{ℓ},

$$f^{\ell} = \frac{\partial^{\ell}}{\partial x^{\ell}} f(c_i) \quad , \quad 0 \leq |\ell| \leq r-1 \tag{7.13a}$$

and its cell-averages by $\overline{f}_j$,

$$\overline{f}_j = A(C_j) f \quad . \tag{7.13b}$$

Let us define the vector $\overline{e}$, $\overline{e} = \overline{e}(f)$, by

$$\overline{e}_j = \overline{f}_j - \sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} f_{\ell} \tag{7.13c}$$

and consider the application of our adaptive algorithm to the solution of the set of equations

$$\sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} E_{\ell} = \overline{e}_j \tag{7.14a}$$

for the unknowns $E_{\ell} = E_{\ell}(f)$.

Let us denote the stencil which the algorithm assigns to C_i by $J(i;f)$, and denote by $\{F_{\ell}\}$ the solution to the system

$$\sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} F_{\ell} = \overline{f}_j \quad , \quad j \in J(i;f) \quad ; \tag{7.14b}$$

clearly $\{F_{\ell}\}$ is an approximation to $\{f_{\ell}\}$ and

$$E_{\ell} = F_{\ell} - f_{\ell} \quad . \tag{7.14c}$$

Let us denote the value of the norm in (7.10c) by $|E(f)|$, and examine the way in which the algorithm arrives at the stencil $J(i;f)$: We start with the cell C_i, and at each step of the algorithm we look at the side neighbors of the existing stencil and add to it the one which minimizes $E(f)$. We observe that several functions $f^1(x), \cdots, f^s(x)$ can be considered simultaneously by adding the appropriate terms (7.14a)

$$e\left(f^1\right), \cdots, e\left(f^s\right)$$

as an extended matrix for the Gauss elimination and use in (7.10c) $\left|E(f^1,\cdots,f^s)\right|$ which is some weighted combination of $\left|E(f^1)\right|,\cdots,\left|E(f^s)\right|$.

This observation provides a very useful tool for the construction of stencils with special properties. Attempting to minimize the reconstruction error at the centroid c_i of some smooth function will certainly favor a centered stencil; if we use a discontinuous piecewise-smooth function $f(x)$ the algorithm will select a stencil which is as centered as possible subject to the constraint that it should not include the discontinuity.

The simplest way to construct a centered stencil of $\kappa(r)$ cells is to apply this procedure to monomials of degree r

$$f(x) = x^{\ell'} \quad , \quad |\ell'| = r \quad . \tag{7.15a}$$

We note that $e(f)$ (7.13c) takes a particularly simple form:

$$\overline{e}_j = a_{j,\ell'} - \sum_{k=0}^{r-1} \sum_{|\ell|=k} a_{j,\ell} \, \frac{\partial^\ell}{\partial x^\ell}\left(x^{\ell'}\right)\Big|_{c_j} \quad . \tag{7.15b}$$

In order to construct the directionally biased stencil $J_k(i)$ for (7.4) in an automatic way we can use this procedure with a piecewise-polynomial function $f(x)$ of degree r which is discontinuous at the face ∂C_i^k, or alternatively to use a combination of monomials of degree r and a step-discontinuity function which is aligned with ∂C_i^k.

Another way to construct $J_k(i)$ is by using this procedure with a monomial of degree r, but restricting the side-neighbors that we feed into the algorithm to those which are contained in the prescribed conical sections.

We remark that at the end of all these procedures we have an LU decomposition which is needed for the reconstruction; as is customary in Gauss elimination we store both L and U in the same matrix.

We observe that this technique can also be used within the adaptive stencil schemes in order to bias the selection of stencil towards a central one. This can be accomplished by adding to the extended matrix (7.11) a column of $e(f)$ (7.13c) for some smooth function f, and properly weigh $|d|$ with $|E(f)|$ in (7.10c).

We remark that the side-neighbors which we feed into the algorithm are restricted to available ones. Thus near the boundary $\partial\mathcal{D}$, the growth of the stencil is along and away from $\partial\mathcal{D}$.

We would like to point out that the ENO techniques for reconstruction apply as well to interpolation. To get ENO interpolation all we need is to replace the matrix $\overline{Q}$ for the cell-averages (3.7a) by the matrix Q (3.16) for the point values.

Finally we remark that up to now we have stressed the desirability of biasing the reconstruction toward a central stencil for reasons of accuracy and stability. Taking into account the cost of selecting an adaptive stencil, it makes sense to use a fixed central stencil altogether, unless it contains a discontinuity. In order to decide whether this is the case, we can use the quantity θ_i (6.3) which serves as an automatic switch in the context of hybrid reconstruction (6.4). θ_i is $O(1)$ when the stencil contains a discontinuity and $O\left(h^r\right)$ when the function is smooth there. Therefore it seems possible to determine some threshold c, so that an adaptive stencil is used only if $\theta_i > c$. See Appendix 1 for more details.

8. Hyperbolic Systems of Conservation Laws

In previous sections we have considered the reconstruction of a piecewise-smooth function $u(x)$ from its given cell-averages purely as a problem in the approximation of functions, without actually specifying whether $u(x)$ is scalar or a vector function. When we are dealing with a vector function $u(x)$ on the level of approximation, it is up to us whether to consider its components as independent scalar functions or to treat the whole vector as a single entity. Accordingly we can either apply the adaptive reconstruction to a vector function in a component-wise fashion, which means an assignment of independent stencils to each of the components, or treat the vector as a single entity and assign a single stencil for all components. In the first case we use $| \quad |$ in the expressions of Section 7 as an absolute value of a scalar quantity; in the second case we interpret $| \quad |$ as a vector norm.

In this section we discuss the additional aspects that come from the fact that the reconstructed function u is a solution of a hyperbolic system of conservation laws.

Since $u = u(x,t)$ is not only discontinuous but also time-dependent, it is possible for discontinuities to come close to each other and interact. Around the time of interaction, no smooth stencil is available in the section between the nearby discontinuities and some spurious oscillations in the numerical solution can be generated. Our numerical experiments seem to indicate that the component-wise reconstruction is more robust in this case than the vector reconstruction.

Another related problem is the fact that some derived quantities, *i.e.*, functions of the conserved variable u, are subject to constraints imposed by the physical phenomenon that is being modelled, *e.g.*, density and pressure are nonnegative quantities. Numerical experiments with the Euler equations of compressible gas show that slight oscillations in the conserved variables due to interaction can cause much larger oscillations in derived quantities, probably due to the fact that the conserved variables are out of alignment. Fortunately we find that this problem is less severe in multidimensional calculations than it is in the one-dimensional case, where all interactions are necessarily frontal.

In the one-dimensional case we can overcome this difficulty by using locally defined characteristic variables, since this set of variables is smoother than the conserved variables during interaction of discontinuities. This technique applies in a straightforward manner to several space dimensions: For the purpose of reconstructing u in the cell C_i we consider the locally defined linear transformation

$$u = T\left(\overline{u}_i\right) w \quad , \quad w = T^{-1}\left(\overline{u}_i\right) u \tag{8.1a}$$

and observe that due to the linearity of the transformation

$$\overline{w}_j = T^{-1}\left(\overline{u}_i\right) \overline{u}_j \text{ for all } j \quad . \tag{8.1b}$$

We apply component-wise reconstruction to the values in (8.1b) to compute $\frac{\partial^\ell w}{\partial x^\ell}$ at the centroid c_i. From these values we get a reconstruction for u by computing derivatives of u at the centroid by

$$\left.\frac{\partial^\ell u}{\partial x^\ell}\right|_{c_i} = T\left(\overline{u}_i\right) \left.\frac{\partial^\ell w}{\partial x^\ell}\right|_{c_i} \quad . \tag{8.1c}$$

In the one-dimensional case we take $T\left(\overline{u}_i\right)$ to be the matrix of eigenvectors of the Jacobian $\frac{\partial f}{\partial u}$. To get a smoother set of locally defined characteristic variables in several space dimensions, we have to identify the normal direction to the discontinuity in u there, and use the eigenvectors of the Jacobian matrix of the normal flux. We remark that since (8.1) is completely local, the formal accuracy of the reconstruction does not depend on the choice of normal direction.

We turn now to describe another technique that works well for solutions of Euler equations of compressible gas in 1D. In this technique, we use the variables (ρ, q, p), *i.e.*, density, velocity, and pressure in order to reconstruct the conserved variables, which are density, momentum, and total energy. There are two reasons for the success of this approach:

(1) ρ and p are variables that are constrained by the physics of the problem to be nonnegative. By selecting a stencil in which these variables are smoothest enables us to better control the oscillations in these quantities.

(2) q and p are continuous across a contact discontinuity, which is the central wave in the Riemann problem corresponding to the interaction. Consequently these variables are smoother than the conserved variables in regions of interaction.

Let us now return to the general case, and assume that there is a preferred set of variables $w(u)$ which is a nonlinear transformation with a well defined inverse $u(w)$. Let us also assume that we have analytic expressions for $\frac{\partial^\ell}{\partial x^\ell} w\big(u(x)\big)$ and $\frac{\partial^\ell}{\partial x^\ell} u\big(w(x)\big)$. The quantities immediately available to us are $w\left(\overline{u}_j\right)$, which by (3.13b) satisfy

$$w\left(\overline{u}_j\right) = w\big(u(c_j)\big) + O\left(h^2\right) \quad . \tag{8.2a}$$

The coefficients in the $O\left(h^2\right)$ term above involve the quantities

$$A\left(C_j\right)\left(x - c_j\right)^\ell \quad , \quad |\ell| = 2 \quad , \tag{8.2b}$$

which, in smoothly varying grids, are point-values of some differentiable function. Hence for grids which vary smoothly enough we can use a component-wise interpolation $I\big(x; w(\overline{u}\,)\big)$ in (3.14) to get $O\left(h^2\right)$ approximation to derivatives of $w\big(u(x)\big)$ at the centroid c_i. From these values we can compute

$$\left.\frac{\partial^\ell u\big(w(x)\big)}{\partial x^\ell}\right|_{x=c_i} \quad , \quad |\ell| = 1, 2 \tag{8.2c}$$

to $O\left(h^2\right)$ and thus obtain an $O\left(h^3\right)$ reconstruction $R_i(x; \overline{u}\,)$ which results in a third-order accurate scheme.

We observe that unlike (8.1a) the quantities $\left\{w\left(\overline{u}_j\right)\right\}$ are defined globally. Hence this procedure for third-order schemes is easier to program and less expensive than that of the locally defined characteristic variables. When we use this technique for the Euler equations of gasdynamics in 2D and 3D it is advisable to define a local normal direction in each cell and represent the velocity vector in normal and tangential components. This is a linear transformation which does not interfere with the formal order of accuracy of the scheme.

We remark that this technique can be extended to the general case in two ways:

(1) Use the values of $\{w_k(\overline{u}_j)\}$ for the purpose of selecting a stencil in which $w_k(u)$ is smoothest. Use this stencil for the reconstruction of the whole vector u, and compute derivatives of u at the centroid c_i. From these derivatives of u compute analytically

$$\left.\frac{\partial^\ell w_k(u)}{\partial x^\ell}\right|_{c_i}$$

to the desired accuracy. After doing so for all components k, get the derivatives of u at the centroid from the analytic expression for $\frac{\partial u\big(w(x)\big)}{\partial x}$.

(2) Use local linearization: Define

$$T(\overline{u}_i) = \frac{\partial u}{\partial w}(\overline{u}_i) \tag{8.3}$$

and proceed as in (8.1).

9. Rectangular Grids

In this section we describe the schemes of previous sections for the case of rectangular grids, and point out some of the simplifications that result from considering this case. At the end of this section we shall describe in detail the third-order case which is particularly simple and seems to be of immediate practical importance. To simplify notations we consider only the two-dimensional case; extension to 3D is immediate. As is customary we denote the space variables by x and y, and the flux components by f and g.

We consider the IBVP

$$u_t + fx + g_y = 0 \quad , \quad (x, y) \in \mathcal{D} \quad , \quad t \geq 0 \tag{9.1a}$$

$$u(x, y, 0) = u_0(x, y) \quad , \quad (x, y) \in \mathcal{D} \tag{9.1b}$$

with given boundary conditions on $\partial\mathcal{D}$. Typically the exterior part of $\partial\mathcal{D}$ is rectangular. In many applications $\mathcal{D}$ contains holes which correspond to rigid objects, in which case the interior part of $\partial\mathcal{D}$ is in general not aligned with the grid, and may even be curved. In the latter case the cells which are side-neighbors of the interior part of $\partial\mathcal{D}$ are treated by the general formulation of the previous sections. In here we shall consider only rectangular cells, and to simplify things further let us assume that the grid is uniform

$$x_i = ih_x \quad , \quad i = 1, \cdots, I_x \quad , \tag{9.2a}$$

$$y_j = jh_y \quad , \quad j = 1, \cdots, J_y \quad . \tag{9.2b}$$

The cells are identified by the pair (i, j)

$$C_{i,j} = [x_i, x_{i+1}] \, x \, [y_j, y_{j+1}] \quad ; \tag{9.3a}$$

$c_{i,j}$, the centroid of $C_{i,j}$, is

$$c_{i,j} = \left(x_{i+1/2}, y_{j+1/2}\right) \quad . \tag{9.3b}$$

Discarding the multi-index notation, we express the Taylor expansion of $u(x)$ around $c_{i,j}$ by

$$\begin{aligned} u(x,y) = &\sum_{k=0}^{r-1} \frac{1}{k!} \sum_{\ell=0}^{k} \binom{k}{\ell} \left(x - x_{i+1/2}\right)^{\ell} \left(y - y_{j+1/2}\right)^{k-\ell} \\ &\times \frac{\partial^k u}{\partial x^{\ell} \partial y^{k-\ell}} \left(x_{i+1/2}, y_{j+1/2}\right) + O\left(h^r\right) \ . \end{aligned} \tag{9.4}$$

Here we assume that u is sufficiently smooth, $x - x_{i+1/2} = O\left(h_x\right)$, $y - y_{j+1/2} = O\left(h_y\right)$. Applying $A\left(C_{ij}\right)$ to $u(x,y)$ in (9.4) we get that all the terms with odd ℓ on odd $k - \ell$ in the summation above vanish due to anti-symmetry; thus we get

$$\overline{u}_{i,j} = \sum_{k=0}^{\left[\frac{r-1}{2}\right]} \alpha_{2k} \left(h_x^2 \partial_x^2 + h_y^2 \partial_y^2\right)^k u\Big|_{c_{i,j}} + O\left(h^{2\left[\frac{r+1}{2}\right]}\right) \quad , \tag{9.5a}$$

where [ ] denotes the integer part and

$$\alpha_k = 2^{-k}/(k+1)! \quad . \tag{9.5b}$$

For $r = 6$ we get in (9.5a)

$$\begin{aligned} \overline{u}_{ij} = &\Big[u + \frac{1}{24}\left(h_x^2 u_{xx} + h_y^2 u_{yy}\right) \\ &+ \frac{1}{1920}\left(h_x^4 u_{xxxx} + 2h_x^2 h_y^2 u_{xxyy} + h_y^4 u_{yyyyy}\right)\Big]_{c_{ij}} + O\left(h^6\right) \quad . \end{aligned} \tag{9.5c}$$

We turn now to describe the reconstruction $R\left(x,y;\overline{u}\,\right)$, where $\overline{u} = \{\overline{u}_{ij}\}$. As before we rewrite $R_{ij}\left(x,y;\overline{u}\,\right)$, the polynomial of degree $(r-1)$ for the cell C_{ij}, in the form of a Taylor expansion around the centroid. To simplify our notation we translate the origin of the coordinate system to the centroid and scale it by h_x and h_y respectively, i.e.,

$$\frac{x - x_{i+1/2}}{h_x} \leftrightarrow x \quad , \quad \frac{y - y_{j+1/2}}{h_y} \leftrightarrow y \quad ; \tag{9.6}$$

for simplicity we retain the notation (x,y) for the scaled system.

$$R_{ij}\left(x,y;\overline{u}\,\right) = D^{0,0} + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{\ell=0}^{k} \binom{k}{\ell} x^{\ell} y^{k-\ell} D^{\ell,k-\ell} \ , \text{ for } |x| \le \frac{1}{2} \ , \ |y| \le \frac{1}{2} \ . \tag{9.7a}$$

The accuracy requirement (3.5a) can be expressed by

$$D^{\ell,k-\ell} = (h_x)^{\ell} (h_y)^{k-\ell} \left. \frac{\partial^k u}{\partial x^{\ell} \partial y^{k-\ell}} \right|_{(0,0)} + O(h^r) \ ; \ 0 \le k \le r-1 \ , \ 0 \le \ell \le k \ ; \quad (9.7b)$$

the terms $D^{\ell,k-\ell}$ correspond to undivided differences.

In an analogous way to (9.5), we get

$$D^{0,0} = \overline{u}_{i,j} - \sum_{k=1}^{\left[\frac{r-1}{\ell}\right]} \alpha_{2k} \sum_{\ell=0}^{k} \binom{k}{\ell} D^{2\ell,2(k-\ell)} \quad , \quad (9.7c)$$

which is equivalent to the conservation property (3.1b). For $r = 6$ we get from (9.7c)

$$D^{0,0} = \overline{u}_{ij} - \frac{1}{24} \left(D^{2,0} + D^{0,2} \right) - \frac{1}{1920} \left(D^{4,0} + 2D^{2,2} + D^{0,4} \right) \quad . \quad (9.7c')$$

We turn now to describe the simple form that the numerical scheme (5.1) takes for rectangular grids. To do so we introduce the Taylor expansion of the flux (4.9a) in the form

$$f\left(R_{ij}(x,y;\overline{u})\right) = F^{0,0} + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{\ell=0}^{k} \binom{k}{\ell} x^{\ell} y^{k-\ell} F^{\ell,k-\ell} + O(h^r) \quad , \quad (9.8a)$$

$$g\left(R_{ij}(x,y;\overline{u})\right) = G^{0,0} + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{\ell=0}^{k} \binom{k}{\ell} x^{\ell} y^{k-\ell} G^{\ell,k-\ell} + O(h^r) \quad (9.8b)$$

where for $0 \le k \le r-1 \quad , \quad 0 \le \ell \le k$

$$F^{\ell,k-\ell} = (h_x)^{\ell} (h_y)^{k-\ell} \left. \frac{\partial^k}{\partial x^{\ell} \partial y^{k-\ell}} f(u) \right|_{(0,0)} + O(h^r) \quad (9.8c)$$

$$G^{\ell,k-\ell} = (h_x)^{\ell} (h_y)^{k-\ell} \left. \frac{\partial^k}{\partial x^{\ell} \partial y^{k-\ell}} g(u) \right|_{(0,0)} + O(h^r) \quad . \quad (9.8d)$$

The scheme (5.1) takes the form

$$\frac{d}{dt} u_{ij} = -\frac{1}{h_x} \left(\overline{f}_{i+1,j} - \overline{f}_{i,j} \right) - \frac{1}{h_y} \left(\overline{g}_{i,j+1} - \overline{g}_{i,j} \right) \quad (9.9a)$$

with the numerical fluxes

$$\overline{f}_{i,j} = \frac{1}{2} \left[\hat{f}^1_{i-1,j} + \hat{f}^0_{i,j} - \nu\left(\hat{u}^1_{i-1,j}, \hat{u}^0_{i,j} \right) \left(\hat{u}^0_{i,j} - \hat{u}^1_{i-1,j} \right) \right] \quad , \quad (9.9b)$$

$$\overline{g}_{i,j} = \frac{1}{2}\left[\hat{g}^1_{i,j-1} + \hat{g}^0_{i,j} - \nu\left(\hat{v}^1_{i,j-1}, \hat{v}^0_{i,j}\right)\left(\hat{v}^0_{i,j} - \hat{v}^1_{i,j-1}\right)\right] \quad , \tag{9.9c}$$

where for $m = 0, 1$

$$\hat{u}^m_{i,j} = D^{0,0} + \sum_{k=1}^{r-1}(-1)^{k(m+1)}\alpha_k \sum_{\ell=0}^{[k/2]} \binom{k+1}{2\ell+1} D^{k-2\ell,2\ell} \quad , \tag{9.9d}$$

$$\hat{v}^m_{i,j} = D^{0,0} + \sum_{k=1}^{r-1}(-1)^{k(m+1)}\alpha_k \sum_{\ell=0}^{[k/2]} \binom{k+1}{2\ell+1} D^{2\ell,k-2\ell} \quad , \tag{9.9e}$$

$$\hat{f}^m_{i,j} = F^{0,0} + \sum_{k=1}^{r-1}(-1)^{k(m+1)}\alpha_k \sum_{\ell=0}^{[k/2]} \binom{k+1}{2\ell+1} F^{k-2\ell,2\ell} \quad , \tag{9.9f}$$

$$\hat{g}^m_{i,j} = G^{0,0} + \sum_{k=1}^{r-1}(-1)^{k(m+1)}\alpha_k \sum_{\ell=0}^{[k/2]} \binom{k+1}{2\ell+1} G^{2\ell,k-2\ell} \quad ; \tag{9.9g}$$

α_k is given by (9.5b).

We return now to describe an algorithm for reconstruction, *i.e.*, how to compute $D^{\ell,k-\ell}$ in (9.7) to $O(h^r)$ from the given cell averages $\{\overline{u}_{i,j}\}$. The most convenient way to do so in rectangular grids is via a process of deconvolution [9], which is based on the observation that the sliding-average of u

$$\begin{aligned}\overline{u}(x,y) &= A\left([x - h_x/2, x + h_x/2] \times [y - h_y/2, y + h_y/2]\right) \cdot u \\ &= \frac{1}{h_x h_y}\int_{-h_y/2}^{h_y/2}\int_{-h_x/2}^{h_x/2} u(x+\xi, y+\eta)d\xi d\eta\end{aligned} \tag{9.10a}$$

is a smooth function of (x, y) (in fact it is one order smoother than $u(x, y)$) and that the given cell averages are its point values at the centroid, *i.e.*,

$$\overline{u}_{i,j} = \overline{u}\left(x_{i+1/2}, y_{j+1/2}\right) \quad . \tag{9.10b}$$

Expanding $u(x + \xi, y + \eta)$ in the integral as a Taylor series, we get that the relation (9.5) holds for every (x, y). Therefore we can differentiate this relation to express derivatives of the sliding-average $\overline{u}$ in terms of derivatives of u. As in [9], once we properly truncate the RHS and write these relations as a system of linear equations, we get an upper triangular matrix which is easily inverted by back substitution.

Before deriving this system of equations, we introduce the notation

$$\overline{D}^{\ell,k-\ell} = (h_x)^{\ell} (h_y)^{k-\ell} \left. \frac{\partial^k \overline{u}}{\partial x^{\ell} \partial y^{k-\ell}} \right|_{(0,0)} + O(h^r) \ , \ 0 \le k \le r-1 \ , \ 0 \le \ell \le k \quad (9.11a)$$

$$\overline{D}^{0,0} = \overline{u}_{i,j} \quad . \quad (9.11b)$$

We derive this system of equations by symbolic differentiation of (9.7c), where differentiation is equivalent to increasing the appropriate index. Doing so we get

$$\overline{D}^{p,q-p} = D^{p,q-p} + \sum_{k=1}^{\left[\frac{r-1-q}{2}\right]} \alpha_{2k} \sum_{\ell=0}^{k} \binom{k}{\ell} D^{2\ell+p,2(k-\ell)+q-p} \ ,$$

$$\text{for } 0 \le q \le r-3 \ , \ 0 \le p \le r-3-q \ , \quad (9.12)$$

We remark that this deconvolution procedure applies also to smoothly varying grids, provided that the corresponding sliding-average function (9.10) is sufficiently smooth.

Reconstruction is accomplished by substituting an appropriate individual difference for $\left\{\overline{D}^{m,n}\right\}$ in (9.12) and then inverting this upper triangular system of linear equations to get $\{D^{m,n}\}$ by back-substitution. Thus to get a fixed-stencil scheme with a centered stencil we use centered undivided differences for $\overline{D}^{p,q}$; these schemes are naturally of odd-order. For adaptive-stencil schemes we use differences of $\overline{u}$ which are computed within the assigned stencil. We observe that we do not have to actually select a two-dimensional stencil and that the same computational task can be performed with tensor product of one-dimensional stencils as follows:

Step 1

(i) Using values of $\{\overline{u}_{i+m,j}\}$, $m = 0, \pm 1, \pm 2, \cdots$, apply the one-dimensional algorithm (see [8]) to select a one-dimensional stencil of r cells starting with $\hat{\imath}(i,j)$, and use this stencil to compute

$$\left\{\overline{D}_{i,j}^{p,0}\right\} \quad , \quad 0 \le p \le r-1 \quad . \quad (9.13a)$$

(ii) Similarly in the y-direction, select a y-stencil of r cells starting with $\hat{\jmath}(i,j)$ and compute

$$\left\{\overline{D}^{0,q}_{i,j}\right\} \quad , \quad 0 \le q \le r-1 \quad . \tag{9.13b}$$

Step 2

Compute mixed derivatives $\overline{D}^{p,q}$ as follows:

(i) Apply D_x^p, a p^{th} order finite-difference operator in the x-direction to $\overline{D}^{0,p}$

$$\overline{D}^{p,q} = D_x^p \overline{D}^{0,q} \text{ for } \left[\frac{r-1}{2}\right] \le q \le r-1 \quad , \quad 1 \le p \le \min[q, r-1-q] \tag{9.14a}$$

using values of $\overline{D}^{0,q}$ in the x-stencil for (i,j), i.e.,

$$\left\{\overline{D}^{0,q}_{i,j}\right\} \quad , \quad 0 \le i^1 - \hat{\imath}(i,j) \le r-1 \quad . \tag{9.14b}$$

(ii) Similarly in the y-direction compute

$$\overline{D}^{p,q} = D_y^q \overline{D}^{p,0} \text{ for } \left[\frac{r-1}{2}\right] \le p \le r-1 \quad , \quad 1 \le q \le \min(p, r-1-p) \quad , \tag{9.14c}$$

using values of

$$\left\{\overline{D}^{p,0}_{i,j}\right\} \quad , \quad 0 \le j^1 - \hat{\jmath}(i,j) \le r-1 \quad . \tag{9.14d}$$

Mixed derivatives $\overline{D}^{s,s}$ that are evaluated both by (9.14a) and (9.14b) can be either averaged or minmod'ed. The values used in (9.14b) and (9.14c) should be closest to (i,j) as possible.

We observe that due to averaging over cells and faces, many of the mixed derivatives cancel out in the expressions for the point-value and the numerical flux. As we have mentioned in Section 5, the quantities $\{F^{k,\ell}\}$ and $\{G^{k,\ell}\}$ in (9.9) can be computed either by an analytic Taylor expansion (5.2) using values of $\{D^{k,\ell}\}$, or alternatively by finite difference operators applied to $\left\{f(D^{0,0}_{i,j})\right\}$ and $\left\{g(D^{0,0}_{i,j})\right\}$. We can compute these derivatives in exactly the same way as in (9.13)–(9.14) using the same stencil.

We turn now to consider the case $r = 3$, which corresponds to a particularly simple scheme: In (9.12) we get

$$D^{0,0} = \overline{u}_{i,j} - \frac{1}{24}\left(\overline{D}^{2,0} + \overline{D}^{0,2}\right) \tag{9.15a}$$

$$D^{1,0} = \overline{D}^{1,0} \quad , \quad D^{2,0} = D^{2,0} \quad , \tag{9.15b}$$

$$D^{0,1} = \overline{D}^{0,1} \quad , \quad D^{0,2} = \overline{D}^{0,2} \quad ; \tag{9.15c}$$

in (9.9) we get for $m = 0, 1$

$$\hat{u}^{m}_{i,j} = D^{0,0} + \frac{1}{2}(-1)^{m+1} D^{1,0} + \frac{1}{24}\left(3D^{2,0} + D^{0,2}\right) \tag{9.16a}$$

$$\hat{f}^{m}_{i,j} = F^{0,0} + \frac{1}{2}(-1)^{m+1} F^{1,0} + \frac{1}{24}\left(3F^{2,0} + F^{0,2}\right) \tag{9.16b}$$

and symmetric expressions for the y-direction. Note that we do not have to compute the mixed derivative term $D^{1,1}$ in (9.15) because it is not used in (9.16) due to cancellation. Therefore the reconstruction (9.12) is terminated at the end of Step 1 (9.13) for the one-dimensional derivatives. Once we have $D^{2,0}$ and $D^{0,2}$ we compute $D^{0,0}$ by (9.15a), and

$$F^{0,0} = f\left(D^{0,0}\right) \quad , \quad G^{0,0} = g\left(D^{0,0}\right) \quad . \tag{9.17a}$$

Now $F^{1,0}, F^{2,0}, F^{0,2}$ and $G^{0,1}, G^{0,2}, G^{2,0}$ can be computed either by an analytic expansion (5.2) or by differencing the values of $F^{0,0}$ and $G^{0,0}$ on the one-dimensional stencils that were assigned to the interpolation of $\overline{u}$ in Step 1, *i.e.*,

$$\begin{aligned}
F^{2,0}_{i,j} &= F^{0,0}_{\hat{\imath}+2,j} - 2F^{0,0}_{\hat{\imath}+1,j} + F^{0,0}_{\hat{\imath},j} \quad , \\
F^{1,0}_{i,j} &= F^{0,0}_{\hat{\imath}+1,j} - F^{0,0}_{\hat{\imath},j} + \left(i - \hat{\imath} - 1/2\right) F^{2,0}_{i,j} \quad , \\
F^{0,2}_{i,j} &= F^{0,0}_{i,\hat{\jmath}+2} - 2F^{0,0}_{i,\hat{\jmath}+1} + F^{0,0}_{i,\hat{\jmath}}
\end{aligned} \tag{9.17b}$$

with symmetric expressions for G; here $\hat{\imath} = \hat{\imath}(i,j)$ and $\hat{\jmath} = \hat{\jmath}(i,j)$.

In Appendix 2 we describe the implementation of this third-order accurate scheme for the solution of the Euler equations of compressible gas.

We end this section with a brief review of the fourth-order scheme ($r = 4$). From (9.12) we get

$$D^{0,0} = \overline{u}_{i,j} - \frac{1}{24}\left(\overline{D}^{2,0} + \overline{D}^{0,2}\right) \quad , \tag{9.18a}$$

$$D^{1,0} = \overline{D}^{1,0} - \frac{1}{24}\left(\overline{D}^{3,0} + \overline{D}^{1,2}\right) \quad , \quad D^{2,0} = \overline{D}^{2,0} \quad , \quad D^{3,0} = \overline{D}^{3,0} \quad , \tag{9.18b}$$

$$D^{0,1} = \overline{D}^{0,1} - \frac{1}{24}\left(\overline{D}^{0,3} + \overline{D}^{2,1}\right) \quad , \quad D^{0,2} = \overline{D}^{0,2} \quad , \quad D^{0,3} = \overline{D}^{0,3} \quad , \tag{9.18c}$$

$$D^{2,1} = \overline{D}^{2,1} \quad , \quad D^{1,2} = \overline{D}^{1,2} \quad ; \tag{9.18d}$$

in (9.9) we get

$$\hat{u}^{m}_{i,j} = D^{0,0} + \frac{1}{24}\left(3D^{2,0} + D^{0,2}\right) + \frac{1}{2}(-1)^{m+1}\left[D^{1,0} + \frac{1}{24}(D^{3,0} + D^{1,2})\right] \quad , \tag{9.19a}$$

$$\hat{f}^{m}_{i,j} = F^{0,0} + \frac{1}{24}\left(3F^{2,0} + F^{0,2}\right) + \frac{1}{2}(-1)^{m+1}\left[F^{1,0} + \frac{1}{24}(F^{3,0} + F^{1,2})\right] \quad , \tag{9.19b}$$

and symmetric expressions for the y-direction. After completing the computation of the one-dimensional derivatives of D in (9.13), we get $D^{2,1}$ by (9.14c), *i.e.*,

$$D^{2,1}_{i,j} = \pm\left(D^{2,0}_{i,j\pm 1} - D^{2,0}_{i,j}\right) \tag{9.20a}$$

depending on which of the points $(i, j \pm 1)$ is included in the y-stencils; if both are we can define $D^{2,1}_{i,j}$ by either

$$D^{2,1}_{i,j} = \text{min mod}\left(D^{2,0}_{i,j} - D^{2,0}_{i,j-1}, D^{2,0}_{i,j+1} - D^{2,0}_{i,j}\right) \tag{9.20b}$$

or

$$D^{2,1}_{i,j} = \frac{1}{2}\left(D^{2,0}_{i,j+1} - D^{2,0}_{i,j-1}\right) \quad ; \tag{9.20b$'$}$$

$D^{1,2}_{ij}$ is computed in a symmetric way.

When we compute derivatives of f and g by differencing, we follow the exact same procedure that was used for computing derivatives of u.

10. Time Integration

The semi-discrete formulation (4.1)'can be expressed in operator form as

$$\frac{dv}{dt} = S\big(v(t)\big) \quad , \quad v(t) = \big\{v_j(t)\big\} \tag{10.1a}$$

$$S\big(v(t)\big)_j = -\sum_k \frac{\partial C_j^k}{\big|C_j^k\big|} \overline{f}_{j,x}^k \quad . \tag{10.1b}$$

In order to construct an ENO scheme from the semidiscrete formulation, Shu [16] has designed a class of multistep time integration of the Runge-Kutta (RK) type which has the property that when applied to a total-variation-bounded (TVB) spatial operator S, it is also TVB. For a third-order scheme this can be written as

$$v_0 = v^n \quad , \quad K_0 = S\left(v_0\right) \tag{10.2a}$$

$$v_1 = v_0 + K_0 \quad , \quad K_1 = S\left(v_1\right) \tag{10.2b}$$

$$v_2 = v_0 + \frac{1}{4}\left(K_0 + K_1\right) \quad , \quad K_2 = S\left(v_2\right) \tag{10.2c}$$

$$v^{n+1} = v_3 = v^n + \frac{1}{6}\left(K_0 + K_1 + 4K_2\right) \quad . \tag{10.2d}$$

Since v_0, v_1, and v_2 in (10.2) correspond to different levels of time, the location of discontinuities in these functions is different. A discontinuity in the solution which was located near the boundary of the cell C_i for v_0 may very well be located at a neighboring cell for v_1. This observation suggests that the stencil-selection procedure, which is an expensive part of the algorithm, has to be repeated at each step.

In the following we present a single-step time integration method

$$v_j^{n+1} = v_j^n - \tau \sum_k \frac{\big|\partial C_j^k\big|}{\big|C_j^k\big|} \overline{f}_{j,*}^k \quad , \tag{10.3a}$$

which is a modification of the algorithm in [8]. The numerical flux $\overline{f}_{j,*}^k$ is of the same form as (4.6a)

$$\overline{f}_{j,*}^k = \frac{1}{2}\left[\hat{f}_j^k + \hat{f}_*^k - \nu\big(\hat{u}_j^k, \hat{u}_*^k\big)\big(\hat{u}_*^k - \hat{u}_j^k\big)\right] \quad , \tag{10.3b}$$

except that here

$$\hat{u}_j^k \approx \frac{1}{\tau\left|\partial C_j^k\right|}\int_0^\tau \int_{\partial C_j^k} \left[E(t)R_j\right] d_S dt \tag{10.3c}$$

and

$$\hat{f}_j^k \approx \frac{1}{\tau\left|\partial C_j^k\right|}\int_0^\tau \int_{\partial C_j^k} f\big(E(t)\cdot R_j\big) d_S dt \quad . \tag{10.3d}$$

We note that $u_j(x,t) = E(t)R_j(x;v^n)$ is the small-time solution of (2.1) with the smooth polynomial initial value $u_0(x) = R_j(x;v^n)$. The approximations that we use in (10.3) are obtained by replacing the integrands above by their Taylor expansion in space and time around the centroid of the cell c_j and $t=0$:

$$u_j(x,t) = E(t)R_j = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|+m=k} (x-c_i)^\ell t^m D_{(\ell,m)} + O(h^r) \quad , \tag{10.4a}$$

$$D_{(\ell,m)} = \frac{\partial^{|\ell|+m}}{\partial x_1^{\ell_1}\cdots\partial x_s^{\ell_s}\partial t^m} u(x,t)\bigg|_{x=c_i,t=0} + O\big(h^{r-|\ell|-m}\big) \quad ; \tag{10.4b}$$

$$f\big(E(t)R_j\big) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\ell|+m=k} (x-c_i)^\ell t^m F_{(\ell,m)} + O(h^r) \quad , \tag{10.5a}$$

$$F_{(\ell,m)} = \frac{\partial^{|\ell|+m}}{\partial x_1^{\ell_1}\cdots\partial_s^{\ell_s}\partial t^m} f\big(u(x,t)\big)\bigg|_{x=c_i,t=0} + O\big(h^{r-|\ell|-m}\big) \quad . \tag{10.5b}$$

Again the expressions for $\hat{u}_j^k$ and $\hat{f}_j^k$ take the same form as (4.8a) and (4.10)

$$\hat{u}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|+m=k} b_{(\ell,m)}^k D_{(\ell,m)} \quad , \tag{10.6a}$$

$$\hat{f}_j^k = \sum_{k=0}^{r-1} \sum_{|\ell|+m=k} b_{(\ell,m)}^k \big(F_{(\ell,m)}\cdot N^k\big) \quad , \tag{10.6b}$$

except that here

$$b_{(\ell,m)}^k = \frac{\tau^m}{(m+1)|\ell|!\left|\partial C_j^k\right|}\int_{\partial C_j^k} (x-c_j)^\ell dS = \frac{\tau^m}{m+1} b_\ell^k \tag{10.6c}$$

and b_ℓ^k is (4.8b); $\ell = (\ell_1,\cdots,\ell_s)$ is the vector of s indices for the space variables and $(\ell,m) = (\ell_1,\cdots,\ell_s,m)$ is the extended vector of $m+1$ indices where the last one, m, stands for time.

After computing $\{D_{(\ell,0)}, F_{(\ell,0)}\}$ for $0 \le |\ell| \le r-1$ as in the semidiscrete scheme we proceed to compute $D_{(\ell,m)}$ and $F_{(\ell,m)}$ for $1 \le m \le r-1$, $0 \le |\ell| \le r-1-m$ by a Cauchy-Kowalewski procedure (see [8]) as follows:

DO for $m = 1, \cdots, r-1$

$$\frac{\partial^{|\ell|+m}}{\partial x^\ell \partial t^m} u = -\mathrm{div}\left(\frac{\partial^{|\ell|+m-1}}{\partial x^\ell \partial t^{m-1}} f\right) \quad , \quad 0 \le |\ell| \le r-1-m \tag{10.7a}$$

$$\frac{\partial^{|\ell|+m}}{\partial x^\ell \partial t^m} f = \frac{\partial^\ell}{\partial x^\ell} H^m\left(u_1 \frac{\partial u}{\partial t}, \cdots, \frac{\partial^m u}{\partial t^m}\right) \quad , \quad 0 \le |\ell| \le r-1-m \quad , \tag{10.7b}$$

END DO.

Here we have used differentiation of the PDE (2.1a) to get (10.7a) and H^m denotes the functional dependence

$$\frac{\partial^m}{\partial t^m} f(u) = H^m\left(u, \frac{\partial u}{\partial t}, \cdots, \frac{\partial^m}{\partial t^m}\right) \quad . \tag{10.7c}$$

In the following we present two algorithms for the implementation of (10.7) which are a direct extension of the ones described in Section 5 for the semidiscrete case.

Algorithm 1

Compute $\{D_{(\ell,0)}, F_{(\ell,0)}\}$, $0 \le |\ell| \le r-1$ as in (5.2).

DO for $m = 1, \cdots, r-1$

$$D_{(\ell,m)} = -\ \mathrm{DIV}\left\{F_{(\ell,m-1)}\right\} \quad , \quad 0 \le |\ell| \le r-1-m \tag{10.7a$'$}$$

$$F_{(\ell,m)} = H^{\ell,m}\left(D_{(*,1)}, \cdots, D_{(*,m)}\right) \quad , \quad 0 \le |\ell| \le r-1-m \quad , \tag{10.7b$'$}$$

END DO.

We have used the notation "DIV" for the analog of "div" in terms of indices; $H^{\ell,m}$ denotes the functional dependence in (10.7b) and $*$ stands for spatial derivatives of order less or equal $|\ell|$.

We observe that once $\{D_{(\ell,0)}\}$ are computed, all other calculations in (10.7′) are performed within the cell.

Algorithm 2

Compute $\{D_{(\ell,0)}, F_{(\ell,0)}\}$, $0 \leq |\ell| \leq r-1$ as in (5.4)

DO for $m = 1, \cdots, r-1$.

(i) For all j in the computational domain compute

$$D_{(0,m)} = -\operatorname{DIV}\{F_{(0,m-1)}\} \tag{10.8a}$$

$$F_{(0,m)} = H^m\big(D_{(0,0)}, \cdots, D_{(0,m)}\big) \tag{10.8b}$$

(ii) For all j and $0 \leq |\ell| \leq r-1-m$ compute

$$D_{(\ell,m)} = -\operatorname{DIV}\{F_{(\ell,m-1)}\} \tag{10.8c}$$

$$F_{(\ell,m)} = \frac{\partial^\ell}{\partial x^\ell} I_j\big(x; F_{(0,m)}\big)\Big|_{x=c_j} \quad , \tag{10.8d}$$

END DO.

Here $I_j(x;\cdot)$ denotes interpolation on the stencil that is assigned to the cell C_j.

As in Section 5 we observe that Algorithm 1 is particularly suitable for unstructured grids, provided that the analytic expressions for $H^{\ell,m}$ are available and are simple enough; this is the case for the Euler equation of compressible perfect gas — see Appendix 2. The second algorithm is most suitable for structured grids where the differentiation of the interpolation in (10.8d) can be expressed by some differencing operator. In this case the operational count for Algorithm 2 is rather low.

We would like to point out that the fully discrete schemes described above differ from the abstract scheme (2.9) in one important feature: They use $E(t) \cdot R_j$ (10.3d) rather than $E(t) \cdot R$ in (2.9). The use of the solution to the Riemann problem across the boundary element ∂C_j^k for the whole time-step ignores the interactions at the vertices due to the meeting of different states there. Consequently the CFL limitation of these schemes is $\frac{1}{2}$, rather than 1 for the abstract scheme (2.9).

Comparing the fully discrete versions with the semidiscrete formulation it seems clear that the semidiscrete formulation is easier to program but it is more expensive to use.

References

[1] T. J. Barth and P. O. Frederickson, "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, January 1990.

[2] J. Casper, "Finite-Volume Application of High Order ENO Schemes to Two-Dimensional Boundary-Value Problems," AIAA Paper 91-0631, January 1991.

[3] S. R. Chakravarthy, A. Harten, and S. Osher, "Essentially Non-Oscillatory Shock-Capturing Schemes of Arbitrarily-High Accuracy," AIAA Paper 86-0339, January 1986.

[4] P. Colella and P. R. Woodward, "The Piecewise-Parabolic Method (PPM) for Gas-Dynamical Simulations, " *Journal of Computational Physics*, Vol. 54, 1984, pp. 174-201.

[5] L. J. Durlofsky, S. Osher, B. Engquist, "Triangle Based TVD Schemes for Hyperbolic Conservation Laws", ICASE Report No. 90-10.

[6] S. K. Godunov, "A Finite-Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Matematicheskii Sbornik*, Vol. 47, 1959, pp. 271-290.

[7] A. Harten, "On the Nonlinearity of Modern Shock-Capturing Schemes," in "Wave Motion: Theory, Modelling, and Computation,", Proceedings of a Conference in Honor of the 60th Birthday of Peter Lax; edited by A. J. Chorin and A. J. Majda; Mathematical Sciences Research Institute Publications, Springer-Verlag, 1987, pp. 147-201; also ICASE Report No. 86-69.

[8] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes III," *Journal of Computational*

Physics, Vol. 71, pp. 231-303, 1987; also ICASE Report No. 86-22, April 1986.

[9] A. Harten and S. Osher, "Uniformly High-Order Accurate Non-Oscillatory Schemes I," *SIAM Journal on Numerical Analalysis*, Vol. 24, pp. 279-309, 1987; also MRC Technical Summary Report No. 2823, May 1985.

[10] A. Harten and H. Tal-Ezer, "On a Fourth Order Accurate Implicit Finite Difference Scheme for Hyperbolic Conservation Laws: II. Five-Point Schemes," *Journal of Computational Physics*, Vol. 41, No. 2, pp. 329-356, 1981; also ICASE Report No. 79-10, June 1979.

[11] A. Harten and G. Zwas, "Self Adjusting Hybrid Schemes for Shock Computations," *Journal of Computational Physics*, Vol. 9, No. 3, pp. 568-583, 1972.

[12] F. Lafon and S. Osher, "High Order Filtering Methods for Approximating Hyperbolic Systems of Conservation Laws," ICASE Report No. 90-25, March 1990.

[13] E. Meiburg, A. Rogerson, C-W Shu, "A Numerical Study Concerning the Convergence of ENO Schemes," Preprint, 1990.

[14] C.-W. Shu, "Numerical Experiments on the Accuracy of ENO and Modified ENO Schemes," *Journal of Scientific Computations*," to appear.

[15] C.-W. Shu and S. Osher, "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II," *Journal of Computational Physics*, Vol. 83, 1989, pp. 32-78.

[16] C.-W. Shu and S. Osher, "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 77, 1988, pp. 439-471.

[17] B. van Leer, "Towards the ultimate conservative difference schemes V. A second order sequel to Godunov's method," *Journal of Computational Physics*, Vol. 32, pp. 101-136, 1979.

Appendix 1. Hybrid ENO Reconstruction

In this appendix we describe some preliminary results on the hybrid reconstruction that was symbolically described in Section 6.

First some notations: We consider a stencil of $(r+1)$ points with uniform spacing h

$$(\bar{u}_0, \bar{u}_1, \ldots, \bar{u}_r); \tag{A1.1a}$$

in our application $\bar{u}_j$ are taken to be cell-averages. Let us denote by T the translation operatcr

$$T\bar{u}_j = \bar{u}_{j+1} \tag{A1.1b}$$

and by Δ the undivided forward-differencing operator

$$\text{(A1.1c)} \qquad \Delta = T - I, \quad I = T^0$$

and denote

$$\text{(A1.1d)} \qquad \bar{\Delta}_j^k = (\Delta)^k \bar{u}_j.$$

Next we express $\bar{\Delta}_0^r$ in terms of $\{\bar{\Delta}_j^k\}$, $0 \le j \le r-k$ in the following way

$$\bar{\Delta}_0^r = (\Delta)^{r-k}\bar{\Delta}_0^k = (T-I)^{r-k}\bar{\Delta}_0^k = \sum_{\ell=0}^{r-k} \binom{r-k}{\ell} (-1)^{r-k-\ell} T^\ell \bar{\Delta}_0^k$$

Thus

$$\bar{\Delta}_0^r = \sum_{\ell=0}^{r-k} \alpha_\ell^k \bar{\Delta}_\ell^k, \quad \alpha_\ell^k = (-1)^{r-k-\ell}\binom{r-k}{\ell}. \tag{A1.2}$$

We define θ^k, the automatic switch for the k-th derivative, by

$$\theta^k = \frac{|\bar{\Delta}_0^r|}{\sum_{\ell=0}^{r-k} |\alpha_\ell^k||\bar{\Delta}_\ell^k|} = \frac{|\sum_{\ell=0}^{r-k} \alpha_\ell^k \bar{\Delta}_\ell^k|}{\sum_{\ell=0}^{r-k} |\alpha_\ell^k||\bar{\Delta}_\ell^k|} \tag{A1.3}$$

and observe that

$$0 \le \theta^k \le 1. \tag{A1.4}$$

For computational purposes we add a tolerance ε to the denominator in (A1.3) and take it to be of the size of the round-off error.

We consider now two important cases

$$|\bar{\Delta}_\ell^k| = \delta_{\ell,\ell_0}, \tag{A1.5a)(i}$$

$$\bar{\Delta}_\ell^k = (-1)^\ell |\bar{\Delta}_\ell^k|, \tag{A1.5b)(ii}$$

where $\delta_{\ell,k}$ is the Kronicker-δ. The first case corresponds to a step-discontinuity in $\bar{u}^{(k)}$, the k-th derivative of $\bar{u}$; the second case corresponds to a discontinuity in $\bar{u}^{(m)}$, $m < k$. It is easy to see that in both cases $\theta^k = 1$. Thus

$$\theta^k = 1 \text{ for a discontinuity in } \bar{u}^{(m)}, \quad k \geq m \geq 0. \tag{A1.5c}$$

Next we consider the product $\theta^k \bar{D}^k$ in the case that $\bar{u}^{(r)}$ is continuous and as in (9.11)

$$\bar{D}^k = h^k \bar{u}^{(k)} + O(h^r) \tag{A1.6a}$$

Clearly

$$|\theta^k \bar{D}^k| = O\left(\frac{h^r|\bar{u}^{(r)}|}{h^k|\bar{u}^{(k)}|}\right) \cdot \left[O(h^k|\bar{u}^{(k)}|) + O(h^r)\right] = O(h^r); \tag{A1.6b}$$

note that this remains so even when $u^{(k)}$ and some of its derivatives vanish at a point, i.e.

$$u^{(k)} = \cdots = u^{(k+p-1)} = 0. \tag{A1.7a}$$

In this case the denominator in θ^k is $O(h^{k+p})$, but so is $\bar{D}^k$. Thus

$$\theta^k = O(h^{r-k-p}), \quad \theta^k \bar{D}^k = O(h^r). \tag{A1.7b}$$

Next we consider the case of a discontinuity in $u^{(m)}$, $k < m \leq r$. In this case $\bar{\Delta}_0^r = O(h^m)$ and $\bar{\Delta}_\ell^k = O(h^k)$, hence

$$\theta^k = O(h^{m-k}), \quad \bar{D}^k = h^k \bar{u}^{(k)} + O(h^m) \tag{A1.8}$$

We summarize all these cases by

$$(1-\theta^k)\bar{D}^k = \begin{cases} 0 & k \geq m \geq 0 \\ \bar{D}^k + O(h^m) & k < m \leq r, \\ \bar{D}^k + O(h^r) & m \geq r \end{cases} \qquad 1 \leq k \leq r-1; \tag{A1.9}$$

here m is the order of first discontinuous derivative of $\bar{u}$.

We turn now to consider the r-th order reconstruction (9.7) in the one-dimensional case

$$R_j(x;\bar{u}) = D^0 + \sum_{k=1}^{r-1} \frac{1}{k!} D^k x^k, \quad |x| \leq \frac{1}{2} \tag{A1.10a}$$

$$D^k = h^k u^{(k)}(0) + O(h^r), \quad \bar{D}^k = h^k \bar{u}^{(k)}(0) + O(h^r), \tag{A1.10b}$$

where $\{D^k\}$ are obtained from $\{\bar{D}^k\}$ via deconvolution, i.e. by inverting the system of linear equations

$$\bar{D}^q = D^q + \sum_{k=1}^{[\frac{r-q-1}{2}]} \alpha_{2k} D^{2k+q}, \quad 0 \leq q \leq r-1. \tag{A1.11}$$

Setting $\theta^0 = 0$ we define the hybrid reconstruction as (A.10) in which $\{D^k\}$ are obtained from the system

$$(1-\theta^q)\bar{D}^q = D^q + \sum_{k=1}^{[\frac{r-q-1}{2}]} \alpha_{2k} D^{2k+q}, \quad \text{for } 0 \leq q \leq r-1. \tag{A1.12}$$

The most natural choice of order of accuracy for this hybrid schemes is even $r = 2s$ with $\bar{D}^q$ being the appropriate central differencing for the stencil

$$(\bar{u}_{-s}, \ldots, \bar{u}_0, \ldots, \bar{u}_s), \quad r = 2s. \tag{A1.13}$$

We observe that if $\bar{u}$ has a discontinuous derivative $\bar{u}^{(m)}$ in the stencil (A.13), then from (A1.9) we get that $(1-\theta^k)\bar{D}^k = 0$ for $k \geq m \geq 1$ and from (A1.12) $D^k = 0$ for $r-1 \geq k \geq m \geq 1$. Hence the reconstruction (A1.10a) becomes a polynomial of degree $m-1$.

We turn now to examine the case $r = 2$ of a second-order scheme. In this case

$$D^0 = \bar{D}^0 = \bar{u}_0, \qquad \text{(A1.14a)}$$

$$D^1 = (1-\theta^1)\bar{D}^1$$

with

$$\theta^1 = \frac{|\bar{\Delta}^2_{-1}|}{|\bar{\Delta}^1_0| + |\bar{\Delta}^1_1|} = \frac{|\bar{\Delta}^1_1 - \bar{\Delta}^1_0|}{|\bar{\Delta}^1_1| + |\bar{\Delta}^1_0|} \qquad \text{(A1.14b)}$$

Taking

$$\bar{D}^1 = \frac{1}{2}(\bar{u}_1 - \bar{u}_{-1}) = \frac{1}{2}(\bar{\Delta}^1_1 + \bar{\Delta}^1_0) \qquad \text{(A1.14c)}$$

we get

$$(1-\theta^1)\bar{D}^1 = \frac{1}{2}\left[|\bar{\Delta}^1_1| + |\bar{\Delta}^1_0| - |\bar{\Delta}^1_1 - \bar{\Delta}^1_0|\right] \cdot \frac{\bar{\Delta}^1_1 + \bar{\Delta}^1_0}{|\bar{\Delta}^1_1| + |\bar{\Delta}^0_1|}. \qquad \text{(A1.14d)}$$

Recalling that

$$\min(a,b) = \frac{1}{2}[a + b - |a-b|]$$

we see that

$$(1-\theta^1)\bar{D}^1 = \begin{cases} S \cdot \min(|\bar{\Delta}^1_0|, |\bar{\Delta}^1_1|) & \text{if } \operatorname{sgn}(\bar{\Delta}^1_0) = \operatorname{sgn}(\bar{\Delta}^1_1) = S \\ 0 & \text{otherwise} \end{cases} \qquad \text{(A1.14e)}$$

which is the famous minmod function $m(\bar{\Delta}^1_0,\ \bar{\Delta}^1_1)$. Therefore

$$R(x;\bar{u}) = \bar{u}_0 + x \cdot m(\bar{\Delta}^1_0, \bar{\Delta}^1_1), \quad |x| \le \frac{1}{2} \qquad \text{(A1.15)}$$

which is TVD.

For $r = 4$ we get in (A1.12)

$$\begin{aligned} &\bar{D}^0 = (1-\theta^0)\bar{D}^0 = D^0 + \alpha_2 D^2, \;\; (1-\theta^1)\bar{D}^1 = D^1 + \alpha_2 D^3, \\ &(1-\theta^2)\bar{D}^2 = D^2, \;\; (1-\theta^3)\bar{D}^3 = D^3 \end{aligned} \qquad \text{(A1.16a)}$$

which is inverted to give

$$\begin{aligned} &D^3 = (1-\theta^3)\bar{D}^3, \;\; D^2 = (1-\theta^2)\bar{D}^2, \;\; D^1 = (1-\theta^1)\bar{D}^1 - \alpha_2(1-\theta^3)\bar{D}^3 \\ &D^0 = \bar{D}^0 - \alpha_2(1-\theta^2)\bar{D}^2 \end{aligned} \qquad \text{(A1.16b)}$$

with

$$\begin{aligned}\theta^1 &= |\bar{\Delta}^4_{-2}|/(|\bar{\Delta}^1_{-2}| + 3|\bar{\Delta}^1_{-1}| + 3|\bar{\Delta}^1_0| + |\bar{\Delta}^1_1|),\\ \theta^2 &= |\bar{\Delta}^4_{-2}|/(|\bar{\Delta}^2_{-2}| + 2|\bar{\Delta}^2_{-1}| + |\bar{\Delta}^2_0|),\\ \theta^3 &= |\bar{\Delta}^4_{-2}|/(|\bar{\Delta}^3_{-2}| + |\bar{\Delta}^3_{-1}|),\end{aligned} \tag{A1.16c}$$

and

$$\begin{aligned}\bar{D}^1 &= \frac{2}{3}(\bar{u}_1 - \bar{u}_{-1}) - \frac{1}{12}(\bar{u}_2 - \bar{u}_{-2}),\\ \bar{D}^2 &= \frac{4}{3}(u_1 + u_{-1}) - \frac{1}{12}(u_2 + u_{-2}) - \frac{5}{2}u_0,\\ \bar{D}^3 &= -(\bar{u}_1 - \bar{u}_{-1}) + \frac{1}{2}(\bar{u}_2 - \bar{u}_{-2}).\end{aligned} \tag{A1.16d}$$

The extension to the two-dimensional reconstruction (9.7) is straightforward:

$$R_{ij}(x, y; \bar{u}) = D^{0,0} + \sum_{k=1}^{r-1} \frac{1}{k!} \sum_{\ell=0}^{k} \binom{k}{\ell} x^\ell y^{k-\ell} D^{\ell,k-\ell} \qquad \text{for} \quad |x| \leq \frac{1}{2}, |y| \leq \frac{1}{2} \tag{A1.17a}$$

with $D^{p,q}$ given by inverting the system of linear equations

$$(1-\theta^p_x)(1-\theta^{q-p}_y)\bar{D}^{p,q-p} = D^{p,q-p} + \sum_{k=1}^{[\frac{r-1-q}{2}]} \alpha_{2k} \sum_{\ell=0}^{k} \binom{k}{\ell} D^{2\ell+p,2(k-\ell)+q-p} \qquad \text{for} \quad 0 \leq q \leq r-1, \quad 0 \leq p \leq r-1-q, \tag{A1.17b}$$

where θ^p_x and θ^q_y are the maximal value that the corresponding one-dimensional θ^k (A1.3) takes on the two-dimensional stencil.

Appendix 2. Euler Equations of Gas Dynamics

In this appendix we describe a particular implementation of the third-order accurate scheme (9.15)-(9.16) to the Euler equations of compressible polytropic gas:

$$u_t + f_x + g_y = 0 \tag{A2.1a}$$

$$u = (\rho, \rho q^x, \rho q^y, E)^T, \quad f(u) = q^x u + (0, P, 0, q^x P)^T, \tag{A2.1b}$$

$$g(u) = q^y u + (0, 0, P, q^y P)^T,$$

with the equation of state

$$P = (\gamma - 1)[E - \frac{1}{2}\rho q^2]; \tag{A2.1c}$$

here $q^2 = (qx)^2 + (q^y)^2$, and we denote $m^x = \rho q^x, m^y = \rho q^y$.

In this particular implementation we use the primitive variables

$$w = w(u) = (\rho, q^x, q^y, P)^T \tag{A2.2}$$

for the purpose of reconstruction (see the discussion in section 8). Given cell-averages $\{\bar{u}_{ij}\}$ we compute for all i, j

$$\hat{w}_{i,j} = w(\bar{u}_{ij}) = w(u_{ij}) + O(h^2) \tag{A2.3a}$$

and apply component-wise selection of stencil to $\hat{w}$, i.e. each component $\hat{w}^k$ is assigned two one-dimensional stencils $\hat{\imath}^k(i,j)$, $\hat{\jmath}^k(i,j)$. Using $\hat{\imath} = \hat{\imath}^k(i,j)$ $\hat{\jmath} = \hat{\jmath}^k(i,j)$ we compute for $1 \leq k \leq 4$

$$\hat{D}^k_{xx} = (h_x)^2 \hat{w}^k_{xx}|_{i,j} = \hat{w}^k_{\hat{\imath}+2,j} - 2\hat{w}^k_{\hat{\imath}+1,j} + \hat{w}^k_{\hat{\imath},j} \tag{A2.3b}$$

$$\hat{D}^k_x = h_x \hat{w}^k_x|_{i,j} = \hat{w}^k_{\hat{\imath}+1,j} - \hat{w}^k_{\hat{\imath},j} + (i - \hat{\imath} - \frac{1}{2})\hat{D}_{xx} \tag{A2.3c}$$

and similar expressions for the y-direction. It follows from (A2.3a) that

$$\hat{D}_{xx} = (h_x)^2 w_{xx}(u) + O(h_x^3), \quad \hat{D}_x = h_x w_x(u) + O(h_x^3). \tag{A2.3d}$$

The next set of computation is executed within the cell itself:

(i) Calculate u_x, u_{xx}, u_y, u_{yy} at the center of the cell from the analytic expressions of derivatives of $u(w)$, i.e.

$$\begin{aligned} m_x^x &= q_x^x \rho + q^x \rho_x, \\ m_x^y &= q_x^y \rho + q^y \rho_x, \\ E_x &= P_x/(\gamma - 1) + \frac{1}{2}[q^x m_x^x + q_x^x m^x + q^y m_x^y + q_x^y m^y], \end{aligned} \tag{A2.4a}$$

and

$$\begin{aligned} m_{xx}^x &= \rho q_{xx}^x + 2\rho_x q_x^x + \rho_{xx} q^x \\ m_{xx}^y &= \rho q_{xx}^y + 2\rho_x q_x^y + \rho_{xx} q^y \\ E_{xx} &= P_{xx}/(\gamma - 1) + \frac{1}{2}[m^x q_{xx}^x + 2m_x^x + m_{xx}^x q^x] \\ &\qquad + \frac{1}{2}[m^x q_{xx}^y + 2m_x^y q_x^y + m_{xx}^y q^y] \end{aligned} \tag{A2.4b}$$

and similar expressions in the y-direction. Using the values of (A2.3) in (A2.4) we get approximations D_x, D_{xx}, such that

$$D_x = h_x u_x + O(h_x^3), \quad D_{xx} = h_x^2 u_{xx} + O(h_x^3) \tag{A2.4c}$$

and similarly in the y-direction.

(ii) Calculate

$$u = \bar{u} - \frac{1}{24}(D_{xx} + D_{yy}), \quad f(u), \quad g(u); \tag{A2.5a}$$

note that u is accurate to $O(h^3)$, and therefore also $f(u)$ and $g(u)$.

(iii) Calculate $f_x, f_{xx}, f_{yy}, g_y, g_{yy}, g_{xx}$ from the analytic expressions of derivatives of $f(u)$ and $g(u)$, i.e.

$$\begin{aligned} f_x &= q^x u_x + q_x^x u + (0, Px, 0, q^x P_x + q_x^x P)^T, \\ f_{xx} &= q^x u_{xx} + 2q_x^x u_x + q_{xx}^x u + (0, P_{xx} + 2(q^x)_x P_x + q_{xx}^x P)^T, \\ f_{yy} &= q^x u_{yy} + 2q_y^x u_y + q_{yy}^x u + (0, P_{yy}, 0, q^x P_{yy} + 2q_y^x P_y + q_{yy}^x P)^T. \end{aligned} \tag{A2.5b}$$

Substituting the values of u (A25a), $\hat{D}_x, \hat{D}_{xx}, \hat{D}_y, \hat{D}_{yy}$ (A2.3) and D_x, D_{xx}, D_y, D_{yy} (A2.4) in the RHS of (A2.5b) we get

$$F_x = (h_x)^2 f(u)_x + O(h_x^3), \quad F_{xx} = (h_x)^2 f(u)_{yy} + O(h_x^3), \tag{A2.5c}$$

$$F_{yy} = (h_y^2) f(u)_{yy} + O(h_y^3),$$

and similar expressions in the y-direction.

(iv) Calculate (9.16), i.e. for $m = 0, 1$

$$\begin{aligned} \hat{u}^m_{i,j} &= u_{ij} + \frac{1}{2}(-1)^{m+1} D_x + \frac{1}{24}(3D_{xx} + D_{yy}), \\ \hat{f}^m_{i,j} &= f(u_{ij}) + \frac{1}{2}(-1)^{m+1} F_x + \frac{1}{24}(3F_{xx} + F_{yy}), \end{aligned} \tag{A2.6}$$

and similarly for the y-direction. This ends the calculation within the cell.

Next we compute for all i, j the numerical fluxes by (9.9b), (9.9c). After doing that we can form the RHS of the semi-discrete formulation (9.9a) and update one of the RK steps in (10.2).

As we have mentioned in Section 10, it seems rather wasteful to repeat the calculations in (A2.3)-(A2.6) and (9.9) three times for all the steps of the RK algorithm (10.2), when we can complete the update of the whole time-step by using (10.3) with algorithm 1 in (10.7′) i.e. to compute a modified formula for $\hat{u}^m$ and $\hat{f}^m$ (10.6) in which we add time-derivatives to (A2.6)

$$\begin{aligned} \hat{u}^m_{i,j} &= u_{ij} + \frac{1}{2}(-1)^{m+1}(D_x + \frac{1}{2}D_{xt}) + \frac{1}{2}D_t + \frac{1}{24}(3D_{xx} + D_{yy}) + \frac{1}{3}D_{tt}, \\ \hat{f}^m_{i,j} &= f(u_{ij}) + \frac{1}{2}(-1)^{m+1}(F_x + \frac{1}{2}F_{xt}) + \frac{1}{2}F_t + \frac{1}{24}(3F_{xx} + F_{yy}) + \frac{1}{3}F_{tt}, \end{aligned} \tag{A2.6′}$$

and similar expressions in the y-direction. After that we compute numerical fluxes by the same formula (10.6b) using a single Riemann solver, and continue to compute v^{n+1} by (10.6a). To do so we have to modify the previous algorithm as follows: In (A2.3) we also compute mixed space derivatives by

$$\hat{D}^k_{xy} = m(\Delta^x_- \hat{D}^k_y, \Delta^x_+ \hat{D}^k_y, \Delta^y_- \hat{D}^k_x, \Delta^y_+ \hat{D}^k_x) \tag{A2.3′}$$

where m is the minmod function and $\Delta_\pm$ are respectively the forward and backward undivided difference operators.

In (A2.4) we also compute

$$D_{xy} = h_x h_y u_{xy} + O(h^3) \tag{A2.4′}$$

In (A2.5) we also compute

$$F_{xy} = h_x h^y f(u)_{xy} + O(h^3), \quad G_{xy} = h_x h^y g(u)_{xy} + O(h^3) \tag{A2.5$'$}$$

After (A2.5) we compute time derivatives and mixed time-space derivatives by

$$\begin{aligned} u_t &= -(f_x + g_y) \\ u_{xt} &= -(f_{xx} + g_{xy}) \\ u_{yt} &= -(f_{xy} + g_{yy}) \end{aligned} \tag{A2.7a}$$

$$\begin{aligned} f_t &= q^x_t u + q^x u_t + (0, P_t, 0, q^x_t P + q^x P_t)^T \\ f_{xt} &= q^x_{xt} u + q^x_t u_x + q^x_x u_t + q^x u_{xt} \\ &\qquad + (0, P_{xt}, 0, q^x_{xt} P + q^x_t P_x + q^x_x P_t + q^x P_{xt})^T, \end{aligned} \tag{A2.7b}$$

and similarly for g_t, g_{yt};

$$\begin{aligned} u_{tt} &= -(f_{xt} + g_{yt}), \\ f_{tt} &= q^x_{tt} u + 2q^x_t u_t + q^x u_{tt} + (0, P_{tt}, 0, q^x_{tt} P + 2q^x_t P_t + q^x P_{tt})^T, \end{aligned} \tag{A2.7c}$$

and similarly for g_{tt}.

Note that we do not compute f_{xt} and g_{yt} because they cancel out in the flux integration and thus do not appear in (A2.6$'$). The notation that we use in (A2.6$'$) for time and space-time derivatives is

$$F_t = \tau f(u)_t + O(h^3), \quad F_{xt} = h_x \tau f(u)_{xt} + O(h^3), \quad F_{tt} = \tau^2 f(u)_{tt} + O(h^3) \tag{A2.7d}$$

and similarly for other terms; we assume $\tau = O(h)$.

We remark that it is convenient to derive the quantities $P_t, P_{xt}, P_{yt}, P_{tt}$ by differentiating the equation for the pressure

$$P_t + q^x P_x + q^y P_y + \gamma P(q^x_x + q^y_y) = 0, \tag{A2.8}$$

and derive $q^x_t, q^x_{xt}, q^x_{yt}, q^x_{tt}$ by differentiation of the relation

$$m^x = q^x \rho,$$

and similarly for q^y (see [ ]).

We remark that for purposes of reconstruction in (A2.3) we can use any decomposition of the velocity vector into normal and tangential components rather than q^x and q^y (see Section 8).

Appendix 3. ENO Schemes in Alternating Dual Grids

In this paper we have considered a set of cells $\{C_j\}$ which covers the computational domain with no overlap. We have introduced a reconstruction procedure $R(x, \bar{u})$ in which we assign to each cell a polynomial R_j. Time evolution is done by solving for small time the IBVP (2.8)

$$\begin{aligned} &w_t + \text{div } f = 0 \\ &w(x,0) = R(x; \bar{u}), \end{aligned} \tag{A3.1}$$

followed by averaging of $w(x, \tau)$ on the same set of cells.

We observe that for a time step which is limited by half the CFL number the value at the centroid is defined in terms of the smooth polynomial problem

$$\begin{aligned} &w_t + \text{div } f = 0 \\ &w(x,0) = R_j(x; \bar{u}). \end{aligned} \tag{A3.2}$$

Furthermore, in order to compute the cell-averages we have already prepared the pointwise space and time derivatives at the centroid, so the evolution of the centroid point-value can be done at no extra computational cost. Hence, we can easily compute and store these point-values in order to use them in the beginning of the next time level for purposes of reconstruction. This would enable us to get pointwise derivatives at the centroids from interpolation of these point-values, rather than from direct reconstruction of averages of the conserved variables. This is particularly useful if we want to work with the primitive variables for purposes of selecting smoother data for the solution of Euler equations of gasdynamics as discussed in Section 8.

Once this is done, these point-values are discarded; the point-values are computed anew from the reconstruction at each time step. Hence it is a side calculation for purposes of improved reconstruction and the point-values themselves do not constitute an independent set of variables.

We observe that the point-values at the centroid do not enhance the reconstruction in any other way because they differ from the cell-averages only by an $O(h^2)$ term which contains second and higher order derivatives. To get a better reconstruction we need the two sets, point-values and averages, to be separated in their location. Since the only reasonable place to compute point-values is in the centroid of the cell in which R_j is defined, we look for the possibility of averaging in a different kind of cell. We observe that the cell formed by connecting centroids of cells around a vertex is a suitable one. We refer to this set of cells as vertex-cells and to the original set as centroid cells. When we deal with one of the sets we refer to the other as the dual one. In Figure A3 we show the two sets of cells for a rectangular grid and for a triangulated mesh. We observe that for rectangular grids the dual sets are identical except for a shift; in triangular grids the two sets are different: the centroid set is made of triangles while the vertex set is made of polygons (typically hexagons).

We use these two sets of cells in an alternating fashion in time. At one time-step we assign polynomials R_j to one set of cells and in the next one we assign them to the dual set. At each time-step we compute point-values at the set at which the polynomials R_j are defined, i.e. at the centroid when the polynomials are defined in the centroid set and at the vertex for the time-step in which the polynomials R_j are defined in the vertex cells.

We turn now to consider the computation of the cell-averages. Due to the divergence-free form of the PDE the averages are given by (2.8)

$$|C_j|(v_j^{n+1} - v_j^n) + \int_0^\tau \oint_{\partial C_j} [f(w(x,t)) \cdot N] dS = 0 \tag{A3.3}$$

where $w(x,t)$ is the solution to (A3.1), and N is the outward normal to the boundary ∂C_j. When we consider the restriction of $w(x,0)$ to the cell in question we see that the middle of the cell (centroid or vertex) is an apex for a multiple Riemann problem. However, each side of the cell is intersected by a single discontinuity. Therefore for a time-step which is limited by half the CFL number, the computation of

$$\int_0^\tau \int_{\partial C_j^k} [f(w(x,t)) \cdot N] ds, \tag{A3.4}$$

the flux across the side ∂C_j^k, involves a quasi-one dimensional Riemann problem, which is formed by a linear segment separating two smooth functions that vary in space.

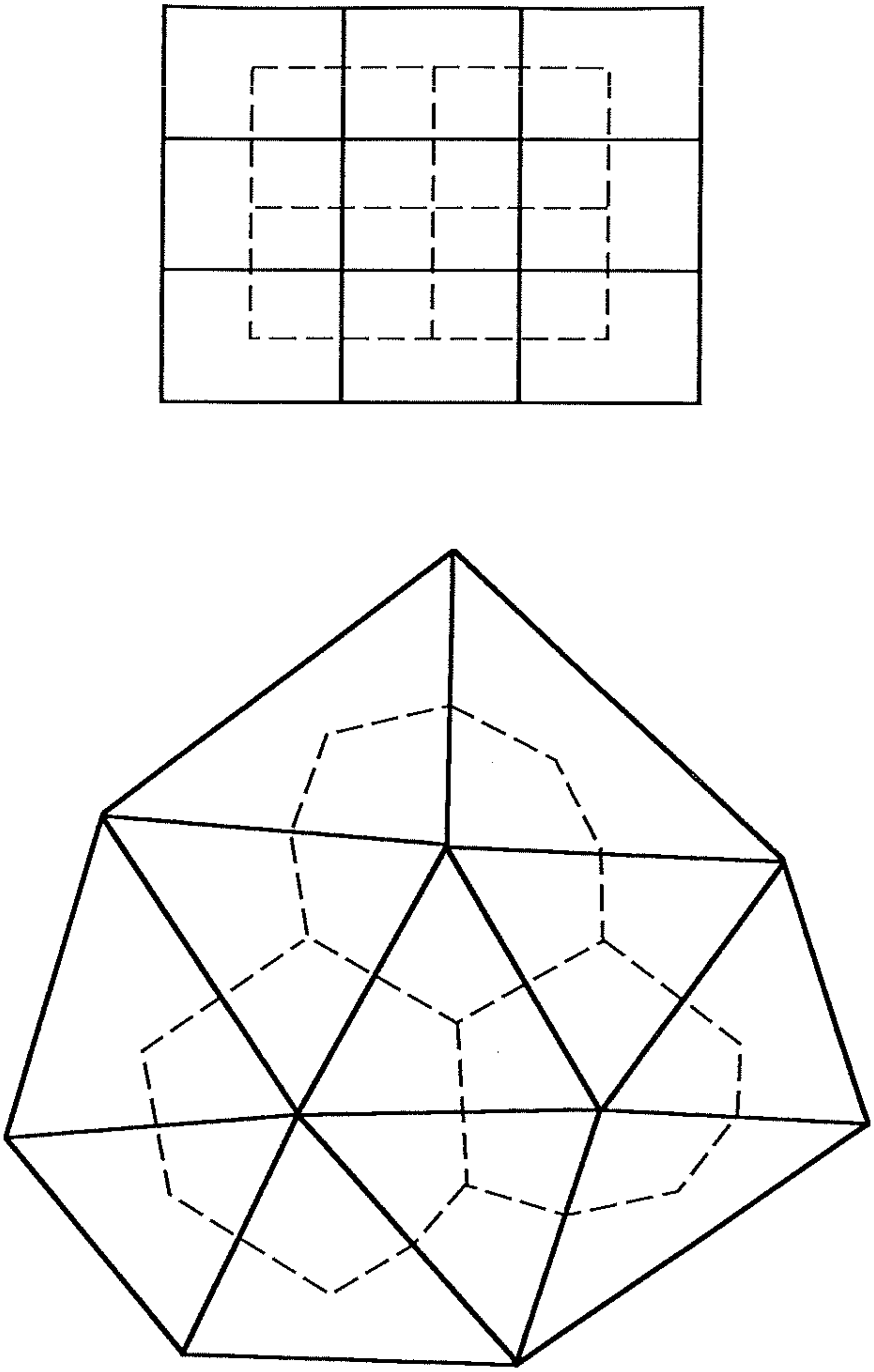

Fig. A3. Dual cells for rectangular and triangular grids

Since these are smoothly varying functions, the solution to such a problem can be approximated by considering a linearized one-dimensional Riemann problem in the direction normal to the linear segment (not to be confused with the normal to ∂C_j^k); the linearization is done with respect to the jump of $w(x,0)$ at the intersection of the linear segment with ∂C_j^k using, say, Roe's technique. We demonstrate this approximation in the two-dimensional rectangular case. First let us align the coordinate system in such a way that the discontinuity is along the x axis and the side of the cell is along the y-axis between $y = -h/2$ to $y = h/2$. We consider the IVP

$$\begin{aligned} &w_t + \text{div } f = 0 \\ &w(x,y,0) = \begin{cases} w_+(x,y) & y \geq 0 \\ w_-(x,y) & y < 0 \end{cases} \end{aligned} \tag{A3.5a}$$

and describe how to approximate the numerical flux $\bar{f}^x$

$$\bar{f}^x = \frac{1}{\tau h} \int_0^\tau \int_{-h/2}^{h/2} f^x(w(0,y,t))dydt. \tag{A3.5b}$$

To do so we use Roe's linearization with respect to $w_-^0 = w_-(0,0)$ and $w_+^0 = w_+(0,0)$, i.e. we consider the corresponding constant coefficient case with a matrix A defined by

$$f^x(w_+^0) - f^x(w_-^0) = A(w_-^0, w_+^0)(w_+^0 - w_-^0) \tag{A3.5c}$$

and use its structure to approximate the integral above. Let $f_\pm(y,t)$ be respectively an approximation to the flux of the smooth solution of

$$\begin{aligned} &w_t + \text{div } f = 0 \\ &w(x,y,0) = w_\pm(x,y). \end{aligned} \tag{A3.6}$$

We approximate $f(w(0,y,t))$ in (A3.5b) by $\tilde{f}^x(y,t)$

$$\tilde{f}^x(y,t) = \begin{cases} f_-(y,t) & -\frac{h}{2} < y < a_1 t \\ f_k(y,t) & a_k t < y < a_{k+1} t \quad k = 1, \ldots, m-1 \\ f_+(y,t) & a_m t < y < \frac{h}{2} \end{cases} \tag{A3.7a}$$

where

$$f_k(y,t) = f_-(y,t) + \sum_{j=1}^{k} [l_j \cdot (f_+ - f_-)] r_j \tag{A3.7b}$$

and $\{a_j, l_j, r_j\}_{j=1}^m$ are the eigenvalues, left-eigenvectors and right eigenvectors of A, respectively. Using $\tilde{f}^x(y,t)$ instead of $f^x(w(0,y,t))$ in the integral in (A3.5b) we get the following numerical flux for this side

$$\bar{f}^x = \frac{1}{2}\left[(\hat{f}^x_- + \hat{f}^x_+) - \sum_{j=1}^{m} \alpha_j r_j\right] \tag{A3.8a}$$

$$\alpha_j = l_j \cdot \left[\hat{f}^x_+ - \hat{f}^x_- - \frac{2}{\tau h}\int_0^\tau \int_{-h/2}^{a_j t} (f_+ - f_-) dy dt\right], \tag{A3.8b}$$

and

$$\hat{f}^x_\pm = \frac{1}{\tau h}\int_0^\tau \int_{-h/2}^{h/2} f_\pm dy dt. \tag{A3.8b}$$

To evaluate these integrals we use the Taylor expansion in space and time that is described in Section 10. We remark that the general case differs from the above only in the alignment of the side on which we compute the numerical flux, i.e. it may be skewed and uncentered.

The main advantage of this setting is that we double the number of values that are available to us for the purpose of reconstruction; this effectively doubles the spatial resolution. The reconstruction procedure uses now a combination of interpolating equations (3.15) and averaging equations (3.6), but the same technique of adaptive selection of stencil can be applied to this combined system.

We remark again that the set of point-values is not an independent one and only plays a role of a side calculation. The method can be viewed as a scheme for cell-averages in which we alternate the set of cells. The most natural way to consider the time evolution aspect of it is by going from reconstruction at t_n to reconstruction at t_{n+1}.

More details will be presented in a future paper.