Acta Numerica (2007), pp. 155–238
doi: 10.1017/S0962492906300013

Finite volume methods for hyperbolic conservation laws

K. W. Morton
Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford OX3 0DW, UK
E-mail: morton@comlab.ox.ac.uk

T. Sonar
Computational Mathematics,
TU Braunschweig, Pockelsstraße 14, D-38106 Braunschweig, Germany
E-mail: t.sonar@tu-bs.de

Finite volume methods apply directly to the conservation law form of a differential equation system; and they commonly yield cell average approximations to the unknowns rather than point values. The discrete equations that they generate on a regular mesh look rather like finite difference equations; but they are really much closer to finite element methods, sharing with them a natural formulation on unstructured meshes. The typical projection onto a piecewise constant trial space leads naturally into the theory of optimal recovery to achieve higher than first-order accuracy. They have dominated aerodynamics computation for over forty years, but they have never before been the subject of an *Acta Numerica* article. We shall therefore survey their early formulations before describing powerful developments in both their theory and practice that have taken place in the last few years.

CONTENTS

1. Introduction

The comprehensive book by Quarteroni and Valli (1994) on the numerical approximation of partial differential equations, which covers finite difference, finite element and spectral methods, devotes only its last eight pages to finite volume methods. However, they do point out that these methods are 'very popular in computational fluid dynamics' and use a terminology for their various formulations which is consistent with that which we will use, so that brief account provides a useful introduction to this survey article.

The term *finite volume method* seems to have appeared in the literature only in the early 1970s (see, *e.g.*, McDonald (1971) and Rizzi and Inouye (1973)) when it was applied to methods used to approximate the hyperbolic conservation law system corresponding to the Euler equations of gas dynamics; but the main ideas are much older. In Varga (1962) an integration method is used to derive finite difference approximations of self-adjoint elliptic equations on a non-uniform rectangular mesh, which reflected standard practice in the nuclear industry at that time and could now be regarded as a standard finite volume method. At about the same time, Preissmann (1961) was advocating a *box scheme* for approximating the St. Venant equations of hydraulic flow which we now regard as one of the basic finite volume schemes.

Consider the scalar conservation law for $u(x,t)$,

$$u_t + f(u)_x = s(x,u), \tag{1.1}$$

which has the form of the momentum equation of the St. Venant system. In deriving a one-dimensional model of a river it is important to divide it up into sections of varying length, each with fairly uniform properties. It is also important to use an implicit time-stepping procedure because the important flood waves typically travel much more slowly than the characteristic waves that would define the CFL stability condition. So we integrate the conservation law over a rectangular box in the (x,t)-plane, use Gauss's theorem to convert the volume integral on the left to an integral along the boundary of the box shown in Figure 1.1(a), and use the trapezoidal rule to approximate the resulting integrals. Using U_j^n to denote our approximation to $u(x_j,t^n)$, we obtain the following scheme, which we consider to be the simplest form of a *cell-vertex scheme*:

$$\begin{aligned} &\tfrac{1}{2}(x_{j+1}-x_j)\Big[U_{j+1}^{n+1}+U_j^{n+1}-U_{j+1}^n-U_j^n\Big] \\ &\quad+\tfrac{1}{2}(t^{n+1}-t^n)\Big[F_{j+1}^{n+1}+F_{j+1}^n-F_j^{n+1}-F_j^n\Big] \\ &\quad=\tfrac{1}{4}(x_{j+1}-x_j)(t^{n+1}-t^n)\Big[S_{j+1}^{n+1}+S_j^{n+1}+S_{j+1}^n+S_j^n\Big], \end{aligned} \tag{1.2}$$

where we have written F_j^n for $f(U_j^n)$ with a similar notation for s.

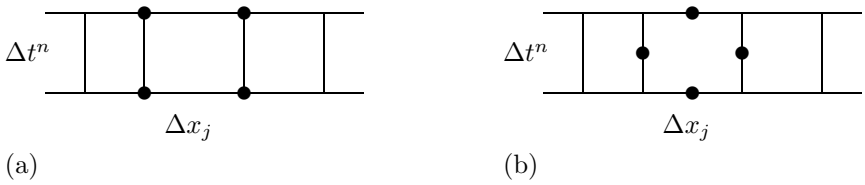


Figure 1.1. (a) The cell-vertex or Preissmann box scheme.
(b) The Godunov or cell-centre scheme.

An even earlier scheme, and one which has been the inspiration for many finite volume methods, is that due to Godunov (1959) (see also Richtmyer and Morton (1967, Section 12.15)) who developed it for application to the Euler equations of gas dynamics. If we apply it to (1.1) with the mesh as shown in Figure 1.1(b), the quantity U_j^n now represents the *cell average* of $u(x,t)$ at time level t^n in cell j, and $F_{j+1/2}^{n+1/2}$ an average between times t^n and t^{n+1} of the flux through the cell boundary at $x_{j+1/2}$. This is normally implemented as an explicit method so that, with cell length $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ and time step $\Delta t^n = t^{n+1} - t^n$, we obtain

$$U_j^{n+1} = U_j^n - \Delta t^n \Big[\Big(F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2} \Big) / \Delta x_j - S_j^n \Big]. \tag{1.3}$$

To obtain the fluxes, the approximation at time level t^n can be interpreted as piecewise constant so that a Riemann problem is set up by the discontinuity at each cell boundary: this is solved exactly or approximately to give the flux. Such a scheme, when developed for two space dimensions, will be called a *cell-centre scheme*.

A scheme of the form (1.3) may seem so obvious, simple and natural that one may wonder why there have been so many alternatives in the literature – even in the class of first-order, explicit schemes. A brief explanation is in order here because it highlights the advantages of the finite volume formulation. In the absence of a source term, we can sum (1.3) over any set of contiguous cells, say $l \le j \le r$, to obtain the overall flux balance

$$\sum_{j=l}^{r} \Delta x_j (U_j^{n+1} - U_j^n) + \Delta t^n \Big[F_{r+1/2}^{n+1/2} - F_{l-1/2}^{n+1/2} \Big] = 0. \tag{1.4}$$

Such a property is crucial to the correct modelling of shocks, whose structure is determined by the conservation law rather than by any differential equation derived from it. And it comes about as a result of two key choices, for the individual fluxes and the mesh length. The calculation of the flux by solving a Riemann problem at a cell boundary can be complicated, and for systems of equations a closed form solution may not exist. So it is natural

to try to make use of fluxes defined as $F_j^n = f(U_j^n)$, as in a finite difference formulation in which U_j^n would be interpreted as a pointwise approximation to u and the mesh length as a difference between the corresponding mesh points. Then a simple explicit first-order scheme would make use of flux differences and take one of the following forms,

$$U_j^{n+1} = U_j^n - \Delta t^n \left[\frac{F_{j+1}^n - F_j^n}{x_{j+1} - x_j} \quad \text{or} \quad \frac{F_j^n - F_{j-1}^n}{x_j - x_{j-1}} \right]. \tag{1.5}$$

Stability, by the CFL condition, would require the choice to be determined by the sign of the characteristic speed $a(u) = f'(u)$ to give a so-called upwind scheme. But any switch between the two at a change in sign of a would preclude the cancellation and collapse of the flux sum that occurs in (1.4). Moreover, the summation on the left would not sensibly represent an integral of U except in the case of a uniform mesh.

So we are driven to the finite volume formulation as above, with some choice of the interface fluxes. The simplest such choice was given by Murman and Cole (1971) and is the scalar form of a Roe-scheme (Roe 1981): it is

$$F_{j+1/2}^{n+1/2} = \begin{cases} f(U_j^n) & \text{if } a_{j+1/2}^n \geq 0, \\ f(U_{j+1}^n) & \text{if } a_{j+1/2}^n < 0, \end{cases} \tag{1.6}$$

where $a_{j+1/2}^n = [f(U_{j+1}^n) - f(U_j^n)]/[U_{j+1}^n - U_j^n]$. This scheme deals with shocks very well; but unfortunately it treats smooth transitions, *i.e.*, expansion waves where $a(\cdot)$ is increasing from left to right, in the same way and hence gives a non-physical kink in the solution. This can be rectified, but the theoretically preferred first-order scheme is due to Engquist and Osher (1981) and takes the following form, where we write $A_j^n = a(U_j^n)$ and u_s is the *sonic point* at which $a(u_s) \equiv f'(u_s) = 0$:

$$\begin{aligned} F_{j+1/2}^{n+1/2} = \tfrac{1}{2}\big[&(1 + \mathrm{sgn}A_j^n)F_j^n + (\mathrm{sgn}A_{j+1}^n - \mathrm{sgn}A_j^n)f(u_s) \\ &+ (1 - \mathrm{sgn}A_{j+1}^n)F_{j+1}^n\big]. \end{aligned} \tag{1.7}$$

These two schemes, (1.6) and (1.7), have very important theoretical properties which we will refer to in later sections, and which make them very important starting points for the development of higher-order schemes for systems of equations in higher dimensions.

The partial differential equation problems that we shall consider in this article will be of the general form

$$\begin{aligned} &\mathbf{u}_t + \mathrm{div}\,\mathcal{F}(\mathbf{u}, \nabla\mathbf{u}) = \mathbf{s}(\mathbf{x}, t, \mathbf{u}), \quad \mathbf{u} : (\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}^+ \to \mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^m, \\ &\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}^0(\mathbf{x}). \end{aligned} \tag{1.8}$$

We shall concentrate on two space dimensions, and many engineering problems are steady, in which case the time t will not be involved. In the purely

hyperbolic cases, such as for the Euler equations of gas dynamics, the fluxes $\mathcal{F}$ will be independent of the gradients $\nabla \mathbf{u}$ so that we have a first-order system of equations. Compressible gas dynamics is a key application area for the methods, however, so that it is important that they are readily applicable to the compressible Navier–Stokes equations through the inclusion of viscous flux terms. Another key feature of this field is that there is normally no source term $\mathbf{s}(\mathbf{x}, t, \mathbf{u})$, with the shape of the boundary being the key determinant of the flow.

The methods that we will describe will be applicable to quite general conservation laws of the form (1.8), but our discussion of them will frequently use terms and ideas that derive from fluid dynamics, as has already been the case. This is partly because this is the field with which we have most experience, but also because of the enormous influence this field has had on the development both of the mathematical models and their numerical approximation – see the beautiful historical essay on this topic by Birkhoff (1983).

Each of the finite volume schemes outlined above will meet new difficulties when applied to problems in two space dimensions; and, as we shall describe below, the extra dimension will lead to alternative variants. The system of equations generated by the Preissmann box scheme applied to the St. Venant equations describing one-dimensional river flow are generally solved by Newton iteration, exploiting the block tridiagonal form of the Jacobian system. But this does not extend to two dimensions. Hence, although the cell-vertex schemes have advantages in accuracy, the resulting algebraic systems are more difficult to solve than those generated by alternative schemes. On the other hand, the cell-centre schemes clearly provide, through their cell averages, only first-order approximations to the flow variables. Thus a very important aspect of these methods is the way in which higher-order approximations are generated from such data as cell averages. This comes within the general compass of *optimal recovery* (see Micchelli and Rivlin (1977)) and a large part of this account will be devoted to this topic. The general framework is as follows. Suppose that an unknown function is assumed to lie in a given function space and one is given the values of a set of linear functionals evaluated for the function. What then is the best estimate that one can make for the value of another linear functional? For example, how does one recover the point values of a function from its cell averages? The choice of mesh will also be crucial, especially in the neighbourhood of complicated flow features such as shocks. We shall therefore devote considerable attention to the topic of mesh adaptivity.

There is one final point that we wish to make in this Introduction. In the vast literature on finite volume methods they have sometimes been generated as finite difference schemes, and sometimes as some sort of finite element method. Given the flexibility and power of the latter methods in

generating approximations on unstructured meshes, and the powerful theoretical framework in which they are formulated, we will below always regard the schemes we describe as finite element methods: in particular, we shall regard them as *Petrov–Galerkin methods*, in which the trial space may take any form but the test space is composed of piecewise constant functions. We will concentrate on algorithmic and theoretical aspects of the methods but will give sufficient numerical examples to demonstrate their power and generality.

2. Systems of conservation laws

Although the finite volume methods that we will develop should be applicable to the general conservation law form (1.8), most of their development and study has been in the context of hyperbolic equations, that is, where the fluxes $\mathcal{F}$ depend only on $\mathbf{u}$. We will therefore concentrate on these throughout this article, and begin by outlining the theoretical background; for a more detailed exposition see Godlewski and Raviart (1991) or Smoller (1983).

2.1. Hyperbolic systems

Consider the system of first-order conservation laws for $\mathbf{u}(\mathbf{x},t) \in \mathbb{R}^m$, $(\mathbf{x},t) \in \mathbb{R}^d \times \mathbb{R}^+$, with initial data $\mathbf{u}(\mathbf{x},0) = \mathbf{u}^0(\mathbf{x})$, in which $\mathcal{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m)$,

$$\partial_t \mathbf{u} + \sum_{\ell=1}^{d} \partial_{x_\ell} \mathbf{f}_\ell(\mathbf{u}) = \mathbf{0}, \tag{2.1}$$

where each flux vector $\mathbf{f}_\ell$ is a C^1-function. Then we can introduce the corresponding Jacobians of the fluxes, which we will denote by A_ℓ, so that when the solution is smooth it satisfies the quasilinear system of equations

$$\partial_t \mathbf{u} + \sum_{\ell=1}^{d} A_\ell(\mathbf{u}) \partial_{x_\ell} \mathbf{u} = \mathbf{0}. \tag{2.2}$$

This system is *hyperbolic* in a region $G \subset \mathbb{R}^m$ of the state space if every linear combination of the Jacobians,

$$A(\boldsymbol{\nu}) := \sum_{\ell=1}^{d} \nu_\ell A_\ell(\mathbf{u}), \tag{2.3}$$

corresponding to a unit vector $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_d)^T \in \mathbb{R}^d$, has m real eigenvalues and associated linearly independent eigenvectors for every $\mathbf{u} \in G$. In a later section we will describe methods which make use of this property to approximate the time evolution of the solution: but for the moment we

note only that it indicates how smooth data can evolve into a non-smooth solution after a finite time. Consider for example the *inviscid Burgers equation*, namely $u_t + uu_x = 0$. Its characteristics are given by $\mathrm{d}x/\mathrm{d}t = u$ along which u is constant, so they are straight lines; so where the initial data is a decreasing function of x it will form a front which will steepen until it breaks to form a shock.

It is therefore necessary to broaden the concept of what constitutes a solution of the PDE problem. The *weak form* of the equation is derived by multiplying (2.1) with a vector of test functions $\boldsymbol{\varphi} \in C_0^1(\mathbb{R}^d \times [0,\infty))^m$ and integrating over a $(d+1)$-dimensional sphere large enough to contain the support of the test functions. Integration by parts then results in

$$\int_{\mathbb{R}^d \times \mathbb{R}^+} \left[\mathbf{u} \cdot \partial_t \boldsymbol{\varphi} + \sum_{\ell=1}^{d} \mathbf{f}_\ell(\mathbf{u}) \cdot \partial_{x_\ell} \boldsymbol{\varphi} \right] \mathrm{d}\mathbf{x}\, \mathrm{d}t + \int_{\mathbb{R}^d} \mathbf{u}^0(\mathbf{x}) \cdot \boldsymbol{\varphi}(\mathbf{x}, 0)\, \mathrm{d}\mathbf{x} = 0. \quad (2.4)$$

So a *weak solution* of (2.1) is one for which (2.4) is satisfied for all such test functions; and $L^1_{\mathrm{loc}}(\mathbb{R}^d \times \mathbb{R}^+)^m$ seems an appropriate space for such solutions.

However, this is too large a space. For example, for Burgers' equation it would include all those containing a jump from a constant u_L on the left to a constant u_R on the right; but only those with $u_L > u_R$ are true *shocks* which would evolve from smooth data or be the limits of solutions to the viscous Burgers equation $u_t + uu_x = \mu u_{xx}$ as the viscosity $\mu \to 0$. Motivated by the equations of fluid dynamics, we therefore introduce the concept of *entropy*. An entropy, for the equation (2.1), is a convex function $\eta : \mathbb{R}^m \to \mathbb{R}$ for which there exists d scalar *entropy fluxes* q_ℓ such that the following relations hold:

$$(\nabla_{\mathbf{u}} \eta)^T A_\ell = (\nabla_{\mathbf{u}} q_\ell)^T \quad 1 \le \ell \le d, \quad (2.5)$$

for each $\mathbf{u}$. Then it is clear that a smooth solution of (2.1) will also satisfy

$$\partial_t \eta(\mathbf{u}) + \sum_{\ell=1}^{d} \partial_{x_\ell} q_\ell(\mathbf{u}) = 0; \quad (2.6)$$

that is, a further conservation law is satisfied. However, when dissipative terms are added to the equations the convexity of η ensures that the left-hand side of (2.6) is non-positive. Thus, when we take the limit as the dissipation tends to zero, we obtain the following *entropy condition* for the weak solution $\mathbf{u}$: $\forall \varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+)$,

$$\int_{\mathbb{R}^d \times \mathbb{R}^+} \left[\eta(\mathbf{u}) \partial_t \varphi + \sum_{\ell=1}^{d} q_\ell(\mathbf{u}) \partial_{x_\ell} \varphi \right] \mathrm{d}\mathbf{x}\, \mathrm{d}t \ge 0, \quad (2.7)$$

the condition corresponding to the given entropy η and its associated flux

functions. A weak solution $\mathbf{u}$ is called an *entropy solution* of the PDE system if such an entropy condition is satisfied for all entropies possessed by the system of equations.

We need to introduce just one further key characterization of solutions to hyperbolic PDEs before we can state an important existence and uniqueness theorem. If g is a real function defined on the open set $\Omega \subset \mathbb{R}^d$, and $g \in L^1_{\rm loc}(\Omega)$, then its *total variation* is defined as

$$TV_\Omega(g) := \sup\left\{ \int_\Omega g \,\mathrm{div}\boldsymbol{\phi}\, \mathrm{d}\mathbf{x},\ \boldsymbol{\phi} \in C^1_0(\Omega)^d, \|\boldsymbol{\phi}\|_{L^\infty(\Omega)} \le 1 \right\}. \tag{2.8}$$

Thus we introduce the notation for functions of *bounded variation*,

$$BV(\Omega) := \left\{ g \in L^1_{\rm loc}(\Omega);\, TV(_\Omega(g) < \infty \right\}.$$

The fact that the existence of smooth solutions to the Navier–Stokes equations in $\mathbb{R}^3$ is one of the Millennium Grand Challenge Problems of the Clay Mathematics Institute – see Jaffe (2006) – shows that we are far from having a comprehensive theory for such PDE problems. However, there is one special case in which all the above concepts show their worth.

In the case of a scalar problem, $m = 1$ in (2.1), results obtained by Oleinik (1957) and Kr̆užkov (1970) enable us to state the following theorem.

Theorem 2.1. If $m = 1$ and the initial data $u^0 \in L^1(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d) \cap BV(\mathbb{R}^d)$ then (2.1) has a unique entropy solution $u(\cdot, t)\ \forall\ t > 0$, for which

$$\|u(\cdot,t)\|_{L^\infty(\mathbb{R}^d)} \le \|u^0\|_{L^\infty(\mathbb{R}^d)}, \tag{2.9}$$

$$TV_{\mathbb{R}^d}(u(\cdot,t)) \le TV_{\mathbb{R}^d}(u^0). \tag{2.10}$$

This result is not only a valuable guide to the selection of numerical methods but it was also proved by taking the limit of approximations obtained by their use. The concept of a *total variation* (TV)*-stable* numerical scheme was introduced by Harten (1984), where he showed that for a scalar problem in one dimension any scheme that is consistent with the conservation law and its entropy inequality gives a convergent approximation if it is TV-stable. Second-order schemes with these properties were presented in that paper and in Harten (1983), where the widely used concept of *TVD* (*total variation diminishing*) schemes was introduced. We note that the two explicit finite volume methods introduced in Section 1, Roe's scheme (1.6) and the Engquist–Osher scheme (1.7), are TVD and stable under a very natural CFL condition.

2.2. Haar's lemma

Unfortunately, since finite volume schemes are based on using piecewise constant test functions, they use an integral form of the PDE and it is not at all clear *a priori* that this will single out the same solutions as the

weak form. Suppose we introduce a *control volume* $\Omega \subset \mathbb{R}^d$ with outward normal $\mathbf{n}$ and surface measure $\mathrm{d}S$. Then, to give a form which will be useful later, we integrate the equation (2.1) over the $(d+1)$-dimensional cylinder $(t, t+\Delta t) \times \Omega$ and apply Gauss's theorem to obtain

$$\int_\Omega [\mathbf{u}(t+\Delta t) - \mathbf{u}(t)]\,\mathrm{d}\Omega + \int_t^{t+\Delta t} \oint_{\partial\Omega} \mathcal{F} \cdot \mathbf{n}\,\mathrm{d}S\,\mathrm{d}t = \mathbf{0}. \tag{2.11}$$

For the present purposes, however, it is convenient to work with a more general $(d+1)$-dimensional control volume Ω^e obtained by extending the $\mathbf{x}$-variable to $\mathbf{x}^e = (t, x_1, x_2, \ldots, x_d)^T$ and similarly $\mathbf{n}$ to $\mathbf{n}^e$ and $\mathrm{d}S$ to $\mathrm{d}S^e$: then we can take advantage of the $(d+1)$-dimensional divergence form of (2.1) to write the more general integral form as

$$\oint_{\partial\Omega^e} \begin{pmatrix} \mathbf{u} \\ \mathcal{F}(\mathbf{u}) \end{pmatrix} \cdot \mathbf{n}^e\,\mathrm{d}S^e = \mathbf{0}. \tag{2.12}$$

Haar's lemma is the general name given to statements which link the weak form (2.4) with this integral form of the PDE: the name derives from the early result given by Haar (1919).

The work of Morrey (1960) (see Klötzler (1970)) can be used to give the following result.

Theorem 2.2. Suppose that $\mathbf{u}$ and the d fluxes $\mathbf{f}_\ell$ are summable over the bounded region $G \subset \mathbb{R}^d \times \mathbb{R}^+$. Then

$$\oint_{\partial C} \begin{pmatrix} \mathbf{u} \\ \mathcal{F}(\mathbf{u}) \end{pmatrix} \cdot \mathbf{n}^e\,\mathrm{d}S^e = \mathbf{0},$$

for almost all cuboids $C \subset G$, if and only if

$$\int_G \begin{pmatrix} \mathbf{u} \\ \mathcal{F}(\mathbf{u}) \end{pmatrix} \cdot \boldsymbol{\nabla}\boldsymbol{\varphi}\,\mathrm{d}\mathbf{x}^e \equiv \int_G \left[\mathbf{u} \cdot \partial_t \boldsymbol{\varphi} + \sum_{\ell=1}^{d} \mathbf{f}_\ell(\mathbf{u}) \cdot \partial_{x_\ell} \boldsymbol{\varphi} \right] \mathrm{d}\mathbf{x}\,\mathrm{d}t = 0$$

for every $\boldsymbol{\varphi}$ which vanishes on or near ∂G and is uniformly Lipschitz-continuous on $\overline{G}$.

The same result has been proved for balls instead of cuboids; and, by using a generalized divergence operator due to Müller (1957), Bruhn (1985) has extended it to quite general control volumes. Thus it is this that we exploit when developing finite volume methods by integration of the conservation laws over quite general shapes.

2.3. Euler and Navier–Stokes equations

In two space dimensions, the Euler equations for inviscid compressible gas flow have the form (2.1), expressing the conservation of mass, the two components of momentum and the total energy. We write them in terms of

the density ρ, the two velocity components $\mathbf{v} := (v_1, v_2)^T$, the pressure p, the total energy per unit mass E and the enthalpy which is defined by $H := E + p/\rho$. Then we have the following definitions of $\mathbf{u}$ and the flux vectors $\mathbf{f}_\ell$:

$$\mathbf{u} := \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{bmatrix}, \qquad \mathbf{f}_1(\mathbf{u}) := \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho v_1 H \end{bmatrix}, \qquad \mathbf{f}_2(\mathbf{u}) := \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho v_2 H \end{bmatrix}.$$

These need to be supplemented by an equation of state giving the pressure in order to close the system: for an ideal gas this is taken to be

$$p = (\gamma - 1)\rho\big(E - \tfrac{1}{2}|\mathbf{v}|^2\big), \tag{2.13}$$

where γ denotes the ratio of specific heats; in the case of dry air it is taken as $\gamma = 1.4$.

As these equations have played such an important role in the development of finite volume methods we will describe their key properties in some detail. The Jacobians of the flux functions have the following form, in terms of the same variables:

$$A_1(\mathbf{u}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}v_1^2 + \frac{\gamma-1}{2}v_2^2 & (3-\gamma)v_1 & (1-\gamma)v_2 & \gamma - 1 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ (\gamma-1)v_1|\mathbf{v}|^2 - \gamma v_1 E & \gamma E - \frac{\gamma-1}{2}(v_2^2 + 3v_1^2) & (1-\gamma)v_1 v_2 & \gamma v_1 \end{bmatrix}$$

and

$$A_2(\mathbf{u}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ \frac{\gamma-3}{2}v_2^2 + \frac{\gamma-1}{2}v_1^2 & (1-\gamma)v_1 & (3-\gamma)v_2 & \gamma - 1 \\ (\gamma-1)v_2|\mathbf{v}|^2 - \gamma v_2 E & (1-\gamma)v_1 v_2 & \gamma E - \frac{\gamma-1}{2}(v_1^2 + 3v_2^2) & \gamma v_2 \end{bmatrix}.$$

These can be written in alternative forms and have several important features. We note first that if we form a linear combination as in (2.3), corresponding to the direction $\boldsymbol{\nu} = (\nu_1, \nu_2)^T$, and introduce the *sound speed* $c := \sqrt{\gamma p/\rho}$, then $A(\boldsymbol{\nu})$ is diagonalizable with eigenvalues $\mathbf{v} \cdot \boldsymbol{\nu}$ (occurring twice), $\mathbf{v} \cdot \boldsymbol{\nu} - c$ and $\mathbf{v} \cdot \boldsymbol{\nu} + c$. Thus the system is hyperbolic, so long as the density and pressure remain positive, and we will give expressions for the eigenvectors of $A(\boldsymbol{\nu})$ below.

The first remarkable property of the Euler equations that we note is their *rotational invariance*. Using the rotation matrix $T(\mathbf{n})$, written in terms of

the unit vector $\mathbf{n} = (n_1, n_2)^T$ as

$$T(\mathbf{n}) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & n_1 & n_2 & 0 \\ 0 & -n_2 & n_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

it is easy to check that the Euler equation fluxes satisfy

$$\sum_{\ell=1}^{2} \mathbf{f}_\ell(\mathbf{u}) n_\ell = T^{-1}(\mathbf{n}) \mathbf{f}_1(T(\mathbf{n})\mathbf{u}).$$

We can take direct advantage of this in an integral form comparable with that in (2.11): if we do not carry out the time integration we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \mathbf{u} \, \mathrm{d}\mathbf{x} + \oint_{\partial\Omega} T^{-1}(\mathbf{n}) \mathbf{f}_1(T(\mathbf{n})\mathbf{u}) \, \mathrm{d}s = \mathbf{0}. \tag{2.14}$$

Another property of the equations that is exploited by some numerical schemes is that the fluxes are *homogeneous functions of degree 1* of the conservative variables. This is obvious for some of the terms but, for example, we can write the second term in the vector $\mathbf{f}_1$ as $u_2^2/u_1 + p$ and $p = (\gamma - 1)[u_4 - \frac{1}{2}(u_2^2 + u_3^2)/u_1]$. It follows that the fluxes satisfy *Euler's relation* so that

$$\mathbf{f}_\ell(\mathbf{u}) = A_\ell(\mathbf{u})\mathbf{u}, \quad \ell = 1, 2. \tag{2.15}$$

This means that for smooth solutions it does not matter whether the Jacobian matrices are included in the spatial differentiation in (2.2) or not: $\partial_{x_\ell}(A_\ell \mathbf{u}) = A_\ell \partial_{x_\ell} \mathbf{u}$.

Where solutions are smooth it is often convenient to write the equations in terms of the so-called *primitive variables* ρ, v_1, v_2 and p. If we form these into the vector $\mathbf{w}$ the gradient matrix defining the change of variables $\nabla_{\mathbf{w}} \mathbf{u} =: M$ is given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ v_1 & \rho & 0 & 0 \\ v_2 & 0 & \rho & 0 \\ \frac{1}{2}|\mathbf{v}|^2 & \rho v_1 & \rho v_2 & (\gamma - 1)^{-1} \end{bmatrix}. \tag{2.16}$$

As this is lower triangular, its inverse can be written down immediately and thence the new coefficient matrices, which we denote by $B_\ell := M^{-1} A_\ell M$, can be derived to give the following:

$$B_1(\mathbf{w}) := \begin{bmatrix} v_1 & \rho & 0 & 0 \\ 0 & v_1 & 0 & \rho^{-1} \\ 0 & 0 & v_1 & 0 \\ 0 & \rho c^2 & 0 & v_1 \end{bmatrix}, \qquad B_2(\mathbf{w}) := \begin{bmatrix} v_2 & 0 & \rho & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & v_2 & \rho^{-1} \\ 0 & 0 & \rho c^2 & v_2 \end{bmatrix}. \tag{2.17}$$

Note that we have here used the fact that the sound speed is given by $c^2 = \partial p/\partial\rho$. Clearly these matrices have a very much simpler form than for the conservative form, and make the calculation of the eigenvalue structure of the system much easier to carry out.

It is an important property of these systems, which is shared by the two-dimensional wave equation system, that the two Jacobian matrices do not commute, so that although any linear combination of them can be diagonalized they cannot be simultaneously diagonalized. With the usual notation $B(\boldsymbol{\nu}) = \nu_1 B_1 + \nu_2 B_2$, and denoting the matrix of its right eigenvectors by $R(\boldsymbol{\nu})$, we have $BR = R\Lambda$, where $\Lambda = \mathrm{diag}(\mathbf{v}\cdot\boldsymbol{\nu}, \mathbf{v}\cdot\boldsymbol{\nu}, \mathbf{v}\cdot\boldsymbol{\nu} - c, \mathbf{v}\cdot\boldsymbol{\nu} + c)$ is the diagonal matrix of eigenvalues and

$$R(\boldsymbol{\nu}) := \begin{bmatrix} \nu_1 & 0 & \rho/2c & \rho/2c \\ 0 & \nu_2 & \nu_1/2 & -\nu_1/2 \\ 0 & -\nu_1 & \nu_2/2 & -\nu_2/2 \\ 0 & 0 & \rho c/2 & \rho c/2 \end{bmatrix}. \tag{2.18}$$

It is then straightforward to obtain the eigenvectors of $A(\boldsymbol{\nu})$ through use of the transformation matrix M: see Hirsch (1990) for details. This form of the equations has been used in the development of finite volume evolution Galerkin methods, which will be described in Section 4.3.

Another form of the equations which is very important from both a theoretical and a practical point of view will be described in Section 6. This makes use of so-called *entropy variables* to symmetrize the equations: that is, to write them as a *symmetric hyperbolic* system, in the linearized form (2.2) but with a matrix A_0 multiplying the time derivative term, in which the three coefficient matrices are symmetric.

A key parameter in the Euler equations is the *Mach number* given by $\mathrm{Ma} := |\mathbf{v}|/c$: when and where it is less than unity the flow is subsonic, and where larger than unity it is supersonic. For steady flows in more than one space dimension, the Euler equations are elliptic where the flow is subsonic and hyperbolic where it is supersonic; and this is the source of some of the characteristic challenges posed by both the analysis of the equations and their numerical modelling. Thus, for most commercial aeroplanes in steady flight, the oncoming flow relative to the aeroplane is subsonic; but it accelerates smoothly around the leading edges to form a supersonic patch which terminates in a shock. Such a flow is termed *transonic*, an example of which is shown later in Figure 6.4.

The Euler equations result from neglecting the effects of viscosity and heat conduction in models of compressible fluid flow. Their inclusion changes the structure of the equations from being purely hyperbolic, and leads to some form of the Navier–Stokes equations. We will outline here the form of these changes; for more details the reader should consult texts such as Hirsch

(1988). With the extra flux terms the equations take the following form:

$$\partial_t \mathbf{u} + \sum_{\ell=1}^{2} \partial_{x_\ell} \left[\mathbf{f}_\ell(\mathbf{u}) - \frac{1}{\mathrm{Re}} \mathbf{g}_\ell(\mathbf{u}) \right] = \mathbf{0}, \tag{2.19}$$

where

$$g_\ell := \begin{bmatrix} 0 \\ \tau_{1,\ell} \\ \tau_{2,\ell} \\ v_1\tau_{1,\ell} + v_2\tau_{\ell,2} + \frac{\mu\gamma}{\mathrm{Pr}}\partial_{x_\ell}\epsilon \end{bmatrix}, \quad \ell = 1, 2.$$

Here the extra terms are controlled by the two parameters, the Reynolds number Re and the Prandtl number Pr, which is a thermodynamic property of the gas equal to 0.72 for air. The terms in the viscous stress tensor are given by $\tau_{i,j} := \mu\big(\partial_{x_j} v_i + \partial_{x_i} v_j\big) + \delta_i^j \lambda\big(\partial_{x_1} v_1 + \partial_{x_2} v_2\big)$, where δ_i^j is the Kronecker delta symbol. The coefficient μ is the viscosity and, by Stokes' hypothesis, we set $\lambda = -\frac{2}{3}\mu$. The heat conduction is given above in terms of the specific internal energy, defined by $\epsilon = E - \frac{1}{2}|\mathbf{v}|^2$. The key coefficient is that for viscosity, which is typically assumed to be given in terms of the temperature T by Sutherland's law $\mu = T^{1.5}(1 + S)/(T + S)$, where $T = \gamma(\gamma-1)(|\mathbf{v}|/c)^2\epsilon$ and $S := 110^\circ K/\overline{T}_\infty$ and $\overline{T}_\infty$ denotes the temperature at infinity.

The details of these formulae are unimportant for our present purposes. The points to note are the structure of the extra terms and the heavy dependence on the computed variables and their gradients. The implication is that in a finite volume method it is most important to have accurate and reliable recovery procedures which, typically from cell averages, can produce both point values and gradients of the dependent variables. In addition, many schemes will combine the inviscid and viscous fluxes, as shown in (2.19), so that they can build on the finite volume techniques developed for convection-diffusion problems: see, *e.g.*, Morton (1996).

Well-publicized test problems have played an important role in the development of numerical models for compressible flows: examples from the early days include the one-dimensional shock tube problem of Sod (1978) and the steady transonic flow past the NACA 0012 aerofoil: see Hirsch (1990). So we conclude this section by showing the results of some Euler calculations for another widely used model problem due to Woodward and Colella (1984). This concerns the supersonic flow of a gas past a forward-facing step, the details of which will be given in a later section. Two triangular meshes are used: a coarse mesh of 2016 triangles and a finer mesh of 8064 triangles, both shown in Figure 2.1. In Figure 2.2 we show contour plots of the Mach number obtained with the coarse mesh (above) and with the fine mesh (below), using two finite volume methods: on the left the plot is obtained

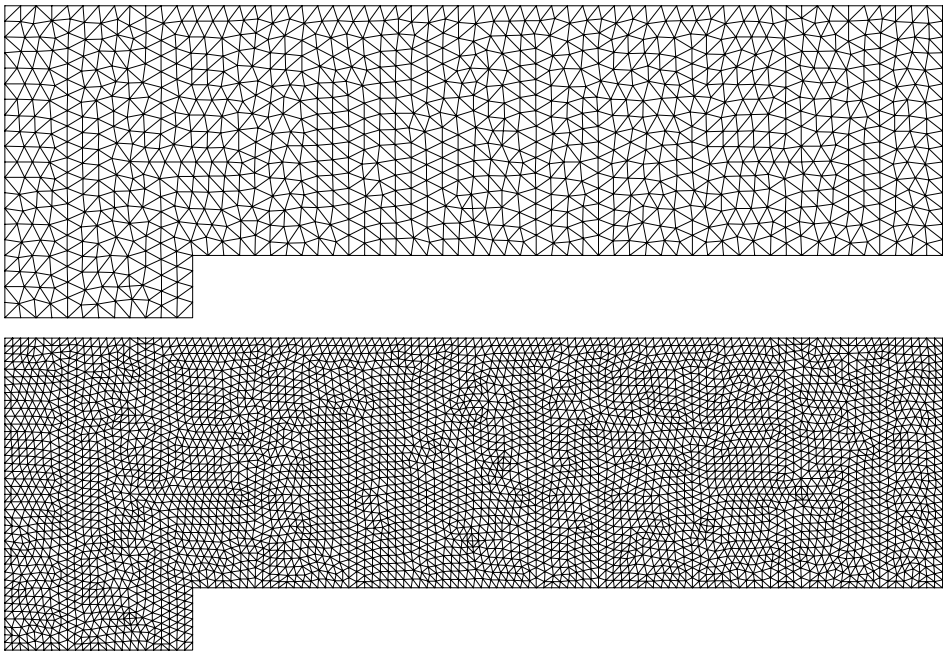


Figure 2.1. Coarse and fine grid for the Woodward and Colella test case.

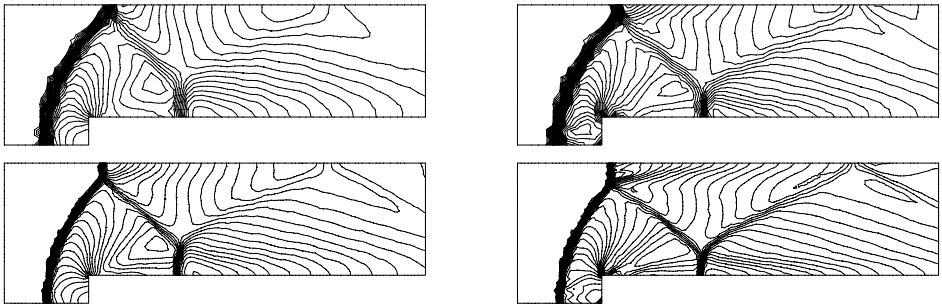


Figure 2.2. Mach number distribution on the coarse mesh (*above*) and on the fine mesh (*below*). Numerical scheme of Steger and Warming (*left*) and Osher and Solomon (*right*).

with a method due to Steger and Warming (1981), which generalizes that shown in (1.6) by exploiting the homogeneous property of the fluxes referred to above; and on the right the plot is obtained with a method due to Osher and Solomon (1982) which generalizes the Engquist–Osher scheme given in (1.7). Apart from the obvious improvement on the finer mesh, it is clear that the second scheme captures more details of the flow, though the two are of the same formal accuracy. We shall see later that the difference between these two schemes lies not so much in their starting points but in the way they generalize from the scalar problem to a system of equations: the former is an example of a *flux-vector splitting method* and the latter of a *flux-difference splitting method*.

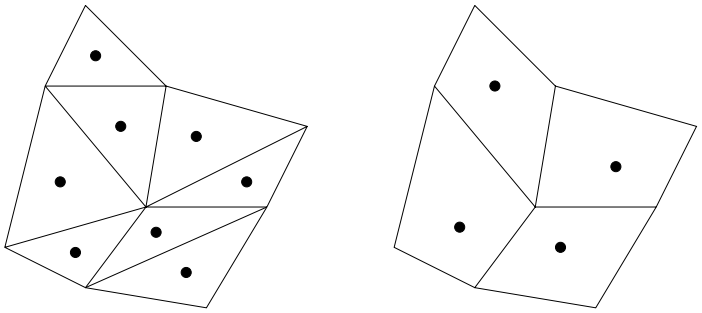
3. Finite volume formulations

There are so many finite volume schemes applied to such a variety of problems, in both the engineering and the numerical analysis literature, that we have to be quite selective in this review. We will concentrate on formulations that are tailored to the needs of the Euler equations of aeronautical gas dynamics because this is the field that has stimulated the most important developments; and we will focus on two space dimensions. However, we will refer to other fields and to three space dimensions when making choices about the methods that we describe in detail.

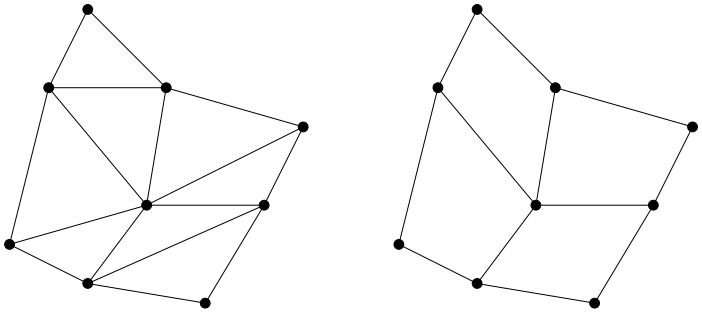
Many of the engineering and design problems in aeronautics concern steady flows; and even in the unsteady problems the rates of change are often very slow when compared with the characteristic sound speed. Thus the approximation employed in the spatial variables is usually quite distinct from that used for the time variable. Advantage can be taken of a finite element formulation in the spatial variables, usually of the Petrov–Galerkin form with the test space different from the trial space. With these points in mind, we will first review some of the choices that have to be made.

3.1. Overall view of alternatives

Triangles vs quadrilaterals. In early two-dimensional calculations of flows around aerofoils quadrilateral meshes were very popular: they are easy to generate and they have the nice property that, globally, there are the same number of vertices as cells. This has an advantage for a cell-vertex formulation, which is preferred on the grounds of accuracy on a stretched mesh (see Morton and Paisley (1989)); and such an approach is readily extended to approximating the Navier–Stokes equations (Crumpton, Mackenzie and Morton 1993). However, triangular meshes are more flexible in modelling complicated geometries and much easier to generalize to three dimensions. So our emphasis will be on triangles, with such meshes often being referred to as unstructured: see Figure 3.1.



(a) Cell-centre triangular (b) Cell-centre quadrilateral



(c) Cell-vertex triangular (d) Cell-vertex quadrilateral

Figure 3.1. Some typical finite volume meshes.

Cell-centre vs cell-vertex. In the former the unknowns are associated with the centres of the cells which act as the control volumes, as indicated in Figure 3.1(a) and (b); while in the latter the unknowns are associated with the vertices of the control volumes, as indicated in Figure 3.1(c) and (d). Thus, in a cell-vertex scheme the basic approximation would naturally be linear on a triangle and bilinear on a quadrilateral, while from a finite element viewpoint the test function would be piecewise constant. The local approximation can be good but difficulties can arise in setting up and solving the overall system of equations. In a cell-centre scheme the unknowns usually represent cell averages, so the local approximation is piecewise constant. Thus from a finite element viewpoint the test and trial spaces are the same, which simplifies setting up the equations: one merely has to interpret the very low-order approximations appropriately. In this respect they can be regarded as early forms of *discontinuous Galerkin methods*: see Cockburn, Karniadakis and Shu (2000).

Node-centred schemes. This is a third alternative, sometimes called *box schemes*, in which the control volumes are centred on the vertices of the primary mesh. When the primary mesh is quadrilateral, then the new mesh can be the same and there is then little difference from corresponding cell-

vertex schemes. But when the primary mesh is triangular, the secondary mesh control volumes are often constructed by joining the centroids to the mid-sides of the primary mesh – as shown in Figure 3.3 – although there are alternative choices for the box shape. This third choice turns out to have several advantages, which we will describe in the course of this article.

Semi-discrete vs time-integrated. In the former approach, often called the *method-of-lines* approach, approximations to only the spatial operators are sought so that an ODE solver then has to be applied to the resulting system of equations. At the other extreme, as in the box scheme of (1.2), integration over time is included in the finite volume formulation. Most commonly some intermediate approach is adopted: we will consider both extremes and relate them to some of the many alternatives.

3.2. Cell-centre schemes and node-centred schemes on triangles

We consider the approximation of the Euler (and later the Navier–Stokes) equations on a bounded open domain $\Omega \subset \mathbb{R}^2$. For the sake of simplicity we assume that the boundary $\partial\Omega := \overline{\Omega}\backslash\Omega$ is already a polygon. On $\overline{\Omega}$ we establish two types of tesselations, a primary and a secondary mesh or grid.

A *triangulation* $\mathcal{T}^h$ of $\overline{\Omega}$ is the set of finitely many triangular subsets $T_i \subset \overline{\Omega}$, $i = 1, \ldots, \#T$, such that the following conditions are satisfied:

- $\overline{\Omega} = \bigcup_{i\in\{1,\ldots,\#T\}} T_i$,
- every $T_i \in \mathcal{T}^h$ is closed and non-empty,
- for two $T_i, T_j \in \mathcal{T}^h$ with $i \neq j$ their interiors satisfy $\overset{\circ}{T}_i \cap \overset{\circ}{T}_j = \emptyset$.

A triangulation is called *conforming* if the following additional condition holds:

- every one-dimensional edge of any $T_i \in \mathcal{T}^h$ is either a subset of $\partial\Omega$ or the edge of another T_j, $j \neq i$.

The parameter h in the notation $\mathcal{T}^h$ corresponds to a typical geometrical length scale of the triangulation which may be represented by the length of the longest edge.

Note that conformity ensures that there can be no *hanging nodes*, *i.e.*, vertices lying in the interior of an edge of another triangle. Although conformity is not necessary in the context of finite volume approximations, it helps to simplify nearly every algorithmic detail, especially in the case of grid adaptivity. The definition of a triangulation given here is identical to that used in finite element methods: see Ciarlet (1987). We call such a conforming triangulation $\mathcal{T}^h$ a *primary grid*.

A *barycentric subdivision* can be used to define a *secondary grid*. Let

$$K_{h,i} := \{T \in \mathcal{T}^h \mid \text{node } i \text{ is vertex of } T\}$$

be the set of all triangles of a primary grid sharing node i. Denote the three edges of triangle T by $e_{T,k}$, $k = 1, 2, 3$. For each $T \in \mathcal{T}^h$ consider the following barycentric subdivision: join the barycentre or centroid of T with the mid-points of its three edges $e_{T,k}$, $k = 1, 2, 3$. This divides each triangle into three segments, as shown in Figure 3.2. The union of all those segments

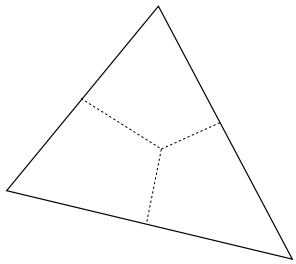


Figure 3.2. Barycentric subdivision of a triangle.

of $T \in K_{h,i}$ adjacent to node i is called the *box* B_i around node i. If node i belongs to the boundary $\partial\Omega$ the box is constructed with the two halves of the boundary edges of the boundary triangles having node i in common. The union $\mathcal{B}^h := \bigcup_{i=1,\dots,\#B} B_i$ of all the boxes is called the *secondary grid*. An example of a primary and secondary grid is shown in Figure 3.3; this includes the situation at the boundary.

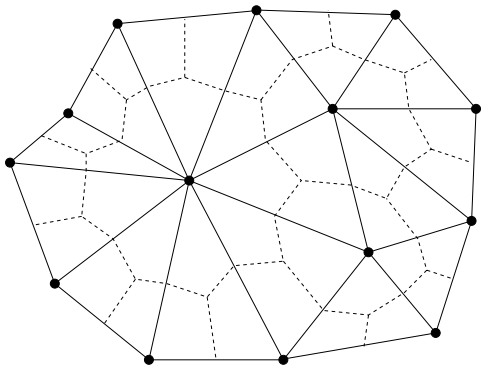


Figure 3.3. Primary and secondary grid for a node-centred scheme.

We next construct cell-centre finite volume approximations on such grids for systems of the type (2.1); we postpone consideration of the viscous fluxes to later in the section. We need the notion of the neighbourhood of a triangle T_i: so we denote the set of indices of its neighbouring triangles by

$$N(i) := \{j \in \mathbb{N} \mid T_i \cap T_j \text{ is an edge of } T_i\}.$$

Then we integrate the conservation law over the control volume formed by the triangle T_i to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{A}(T_i)\mathbf{u}(t) = -\frac{1}{|T_i|}\sum_{j\in N(i)}\int_{\partial T_j\cap\partial T_i}\sum_{\ell=1}^{2}\mathbf{f}_\ell(\mathbf{u})n_{ij,\ell}\,\mathrm{d}\sigma, \tag{3.1}$$

where we denote by $\mathcal{A}(T_i)\mathbf{u}(t)$ the average of $\mathbf{u}(t)$ over the triangle. Here the outer (with respect to T_i) unit normal vector at the edge $\partial T_i \cap \partial T_j$ is denoted by $\mathbf{n}_{ij} = (n_{ij,1}, n_{ij,2})^T$, and $|T_i|$ is the area of the triangle.

To allow maximum flexibility in the development of numerical schemes from this formula we replace the edge integrals by Gaussian quadrature formulae, which will assume a certain degree of smoothness. If we denote by $\mathbf{x}_{ij-}$ and $\mathbf{x}_{ij+}$ the coordinates of the vertex points of $\partial T_i \cap \partial T_j$, the edge is parametrized by $s \in [-1, 1]$ such that the general point on the edge is

$$\mathbf{x}_{ij}(s) := \tfrac{1}{2}[(1-s)\mathbf{x}_{ij-} + (1+s)\mathbf{x}_{ij+}],$$

and this can be introduced in the evolution equation (3.1) on T_i. Suppose we denote the number of Gauss points on $\partial T_i \cap \partial T_j$ by n_G, the actual Gauss points by $\mathbf{x}_{ij}(s_\nu)$, $\nu = 1, \ldots, n_G$, and the weights by ω_ν, then we get the system

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{A}(T_i)\mathbf{u}(t) = \\ -\sum_{j\in N(i)}\frac{|\partial T_i\cap\partial T_j|}{2|T_i|}\left\{\sum_{\nu=1}^{n_G}\sum_{\ell=1}^{2}\omega_\nu\mathbf{f}_\ell(\mathbf{u}(\mathbf{x}_{ij}(s_\nu),t))n_{ij,\ell} + \mathcal{O}\big(h^{2n_G}\big)\right\}.\end{aligned} \tag{3.2}$$

The lowest-order scheme corresponds to the mid-point rule and, although we will concentrate on this form of quadrature, it is clearly a simple matter to replace it by the trapezoidal rule, Simpson's rule or some other choice.

The first step in deriving a corresponding numerical approximation is to introduce $\mathbf{U}_i(t)$ as an approximation to $\mathcal{A}(T_i)\mathbf{u}(t)$. Then the crucial step is to choose a formula giving a *numerical flux function* that generalizes that given in (1.6) for the Roe scheme or (1.7) for the Engquist–Osher method. Because it takes the form of a mapping

$$(\mathbf{u}_L, \mathbf{u}_R; \mathbf{n}) \overset{\mathbf{H}}{\longmapsto} \mathbf{H}(\mathbf{u}_L, \mathbf{u}_R; \mathbf{n}) \in \mathbb{R}^m$$

from two constant states to a flux, in a direction $\mathbf{n}$, it is also called an *approximate Riemann solver*. Note that we will use these terms quite generally to refer to any choices for such a mapping, even to make comparisons with, *e.g.*, Lax–Friedrichs or Lax–Wendroff difference schemes. The essential condition it has to satisfy is a consistency condition with the differential

equation,

$$\forall \mathbf{u} \in \mathbb{R}^m : \quad \mathbf{H}(\mathbf{u}, \mathbf{u}; \mathbf{n}) = \sum_{\ell=1}^{2} \mathbf{f}_\ell(\mathbf{u}) n_\ell. \tag{3.3}$$

Because of this condition we can replace one of the sums over the fluxes in (3.2) by the corresponding $\mathbf{H}$ function. But more importantly, with the additional choice of a one-point quadrature rule, it leads to the following definition of the basic cell-centre finite volume scheme:

Find $\mathbf{U}_i(t), i = 1, \ldots, \#T, t \in [0, t^*], t^* > 0$, *as a solution of the system of ordinary differential equations*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_i(t) = -\frac{1}{|T_i|} \sum_{j \in N(i)} |\partial T_i \cap \partial T_j| \mathbf{H}(\mathbf{U}_i(t), \mathbf{U}_j(t); \mathbf{n}_{ij}), \tag{3.4}$$

$$\mathbf{U}_i(0) = \mathcal{A}(T_i)\mathbf{u}(0).$$

After a discretization in time this will form the basis of all the cell-centre finite volume schemes on triangular meshes that we shall discuss. The simplest time discretization is obtained with the explicit Euler scheme. Then we will have a direct generalization of (1.3) to systems of conservation laws in two dimensions. Such a choice will lead to a stability limit on the time step, which is in the form of a *CFL condition* (Courant, Friedrichs and Lewy 1928). In the scalar one-dimensional case, this requires that no characteristic can cross more than one cell in one time step: in the notation of Figure 1.1(b) this becomes

$$-\Delta x_{j-1} \le f'(u)\Delta t \le \Delta x_j \quad \text{for } u \text{ between } U^n_{j-1} \text{ and } U^n_j, \ \forall j. \tag{3.5}$$

For both the Roe flux (1.6) and the Engquist–Osher flux (1.7), it is shown in Morton (2001) that this condition is sufficient as well as necessary for stability. However, it is clear that such a condition will become more restrictive and more complicated in two dimensions on a triangular mesh. Other more sophisticated time discretizations are therefore needed, some of which will be described in Section 4.4.

One cannot expect to obtain better than first-order accuracy with a scheme that only uses a piecewise constant approximation to the unknown solution. To do better we introduce a recovery step: this produces a higher-order approximation $\widetilde{\mathbf{U}}(t)$ which preserves the cell averages,

$$\mathcal{A}(T_i)\widetilde{\mathbf{U}}(t) = \mathbf{U}_i(t), \ i = 1, \ldots, \#T;$$

and the values of this function at the quadrature points are then substituted into the calculation of the numerical flux functions in (3.4). The details of how this is done will be described in Section 5 but we will introduce some of the ideas here.

For the one-dimensional scheme (1.3), an approach that has led to the popular MUSCL algorithms introduced by van Leer (1979) makes use of discontinuous linear recovery for each variable: since the average is to be preserved in a cell, one need only choose a slope S_j for the variable in each cell. This is usually obtained by combining the divided differences D_+U_j and D_-U_j between the average in the cell and those to its right and left, but it is important to impose some restrictions on how this is done. Thus in the case of a scalar conservation law it is easily deduced from the TVD property (2.10) that monotone initial data remains monotone. Hence if a set of cell averages form a monotone sequence this property should be preserved and this makes the choice of recovery algorithm far from trivial even in this case. For example, suppose that the recovered function in cell j is given by

$$\widetilde{U}(x) = U_j + S_j(x - x_j), \tag{3.6}$$

where x_j is the centre of the cell. Then one might aim to ensure that whenever the sequence U_j is monotone increasing, then so is $\widetilde{U}(x)$; and it is easily seen that a sufficient condition for this is to have

$$0 \le S_j \le \min(D_+U_j, D_-U_j) \quad \forall j.$$

Formulae for S_j based on such considerations are referred to as *slope limiters*, with the best-known being that given by the minimum of $|D_\pm U_j|$ and called the *minmod limiter*:

$$\operatorname{minmod}(x,y) := \begin{cases} s\min(|x|,|y|) & \text{if} \quad \operatorname{sgn} x = \operatorname{sgn} y = s, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

However, such a choice is rather conservative, and could lead to clipping of local extrema, so many alternatives have appeared in the literature, a topic we will return to in Section 4.6 on ENO schemes. Meanwhile, a necessary condition for preserving a monotone increasing function that we will refer to later (and that originally suggested by van Leer) is the following:

$$U_{j-1} \le U_j - \tfrac{1}{2}S_j\Delta x_j \quad \text{and} \quad U_j + \tfrac{1}{2}S_j\Delta x_j \le U_{j+1}; \tag{3.8}$$

that is, the variation in the cell does not go beyond the averages in its neighbours.

To generalize this approach to a triangular mesh, we need to calculate a gradient for a variable in each cell from the cell averages in neighbouring cells. One way to do this is to use the general formula for obtaining an average gradient of a variable over a region Ω from values on its perimeter,

$$\mathcal{A}(\Omega)\nabla u = \frac{1}{|\Omega|}\int_{\partial\Omega} \begin{bmatrix} u\,\mathrm{d}x_2 \\ -u\,\mathrm{d}x_1 \end{bmatrix},$$

and applying this to the secondary grid cell around each vertex. This gives a choice of three gradients for each triangle. An alternative due to Durlofsky,

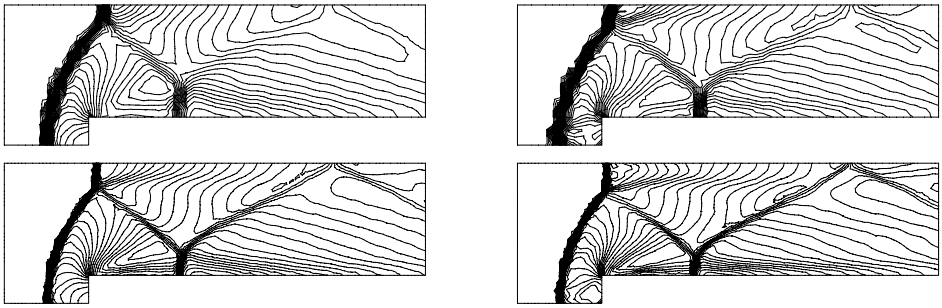


Figure 3.4. Mach number distribution on the coarse mesh (*above*) and on the fine mesh (*below*) for the Woodward and Colella test case. Numerical scheme of Steger and Warming (*left*) and Osher and Solomon (*right*), both with linear recovery.

Engquist and Osher (1992) is a direct generalization of the TVD construction in one dimension. It makes use of the three neighbouring triangles, *i.e.*, those in $N(i)$, which is often called the von Neumann neighbourhood of T_i. From any pair we can construct a linear function whose averages over the pair and T_i match the corresponding values of U. Again this gives a choice of three gradients.

One needs to combine or choose between these gradients in some way, for each component of $\mathbf{U}$, and then calculate the numerical fluxes along each edge of the triangular mesh, where the variables are not only discontinuous but also non-constant; and this should be done in such a way as to utilize known properties of the differential equation system, such as monotonicity preservation or the TVD property. For example, choosing the gradient of a variable with smallest absolute value except at an extremum would be a direct generalization of the minmod limiter (3.7). However, this is rather severe and generally more sophisticated choices are made: see Section 5. These issues highlight the key principles of the recovery process: any known properties of the unknown function $\mathbf{u}(t)$ can be exploited in constructing the higher-order approximation $\widetilde{\mathbf{U}}(t)$; but its projection onto the lower-order space has to be such as to reproduce $\mathbf{U}(t)$.

We illustrate the effectiveness of such algorithms by means of the forward-facing step problem already referred to. In Figure 3.4 we show on the coarse mesh (above) and the fine mesh (below) the results obtained after linear recovery for the two schemes for which the corresponding results without recovery were shown in Figure 2.2. They clearly show the improvement due to the recovery, on both meshes.

We conclude this section by describing node-centred schemes on a triangular mesh, which we will gradually see have several advantages over the alternative cell-centre schemes, including the choice of a linear recovery

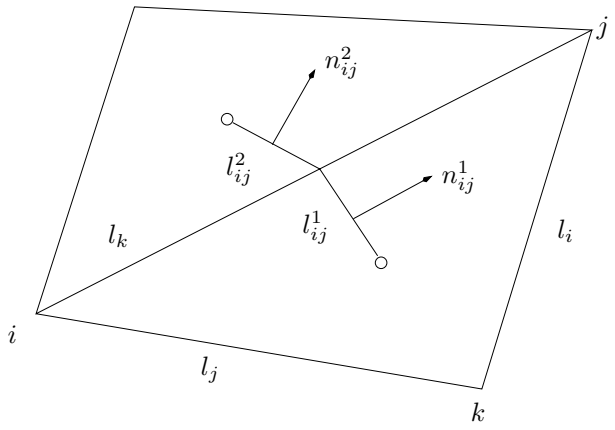


Figure 3.5. Geometry between boxes B_i and B_j.

procedure. The piecewise constant approximation in this case is given by the values $\mathbf{U}_i$ which represent averages over the boxes B_i centred on the vertices and forming the secondary grid. It is then straightforward to construct a piecewise linear function corresponding to each component variable U on each triangle by choosing the three required nodal values so that the average of the function over each box B_i corresponding to one of the nodes matches U_i. Then each triangle yields a gradient of this function; and in the box B_i we choose the gradient with the smallest magnitude from those that correspond to triangles which share the node i, *i.e.*, are in the set $K_{h,i}$. Thus we obtain a discontinuous piecewise linear approximation to the vector of unknowns which we denote by $\widetilde{\mathbf{U}}(t)$.

The node-centred update formula corresponding to (3.4) is a little more complicated because the boundary of each box B_i has more segments: we denote the set of indices of boxes that are neighbours to B_i by

$$N^B(i) := \{j \in \mathbb{N} \mid B_i \cap B_j \text{ is edge of } B_i\};$$

and the boundary between two neighbouring boxes consists of two segments with different normals, as shown in Figure 3.5. Then, with the notation shown in the figure and with one-point Gaussian quadrature at the corresponding mid-points $\mathbf{x}_{ij}^k$ of the segments, we obtain the system of ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_i(t) = -\frac{1}{|B_i|}\sum_{j\in N^B(i)}\sum_{k=1}^{2} |l_{ij}^k|\mathbf{H}\big(\widetilde{\mathbf{U}}_i(\mathbf{x}_{ij}^k,t), \widetilde{\mathbf{U}}_j(\mathbf{x}_{ij}^k,t); \mathbf{n}_{ij}^k\big), \tag{3.9}$$

$$\mathbf{U}_i(0) = \mathcal{A}(B_i)\mathbf{u}(0). \tag{3.10}$$

Choice of an integrator for this system completes the definition of the method.

With this scheme and its recovery procedure it is reasonably straightforward to include the viscous fluxes and thence approximate the Navier–Stokes equations (2.19). We have linear approximations to each variable on each triangle, so that their gradients are readily computed on each section of the boundary of a box B_i; and since they are constant on each triangle they are exactly integrated by any Gaussian quadrature rule. There is just one snag. For realistic values of the Reynolds number (say, Re $= \mathcal{O}(10^6)$) it is necessary to have a highly stretched mesh near the boundary of the domain, with long thin triangles aligned with the boundary; then the normals to an edge of a box will point mainly towards and away from the boundary, rather than along it. This is clearly inappropriate for the inviscid fluxes. The remedy is to replace the barycentres of each triangle by an appropriately weighted average of its vertex positions. Such a formula is given by

$$\mathbf{x}^s = \sum_{m\in\{i,j,k\}} \alpha_m^s \mathbf{x}_m \quad \text{where} \quad \alpha_m^s := \frac{1}{2(|l_i| + |l_j| + |l_k|)} \sum_{\substack{m'\in\{i,j,k\}\\ m'\neq m}} |l_{m'}|;$$

the effect on the mesh is illustrated in Figure 3.6.

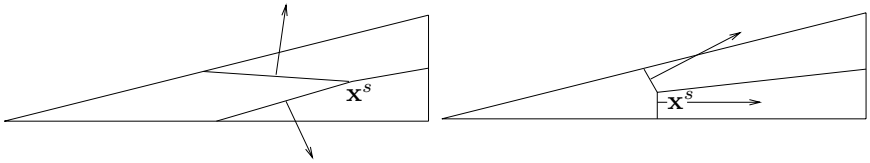


Figure 3.6. Stretched grid cell for Navier–Stokes; deformed boxes on the right.

3.3. Cell-vertex schemes on quadrilaterals

We can subdivide a bounded open domain $\Omega \subset \mathbb{R}^2$, with a polygonal boundary, into a set of quadrilaterals $Q_\alpha \subset \overline{\Omega}$, $\alpha = 1, \ldots, \#Q$ in exactly the same way as the triangulation described in the previous subsection. We will also assume it is conforming, in the same sense, and it is unnecessary to repeat here the formal detailed specification of the subdivision. However, we now use Greek subscripts $\alpha, \beta, \ldots$ to refer to the quadrilaterals and reserve Roman subscripts $i, j, \ldots$ to refer to their vertices, with which the variables $\mathbf{U}$ will be associated. In addition we include the viscous fluxes from the outset and seek a steady solution of the Navier–Stokes equations. So the formulation will be much closer to a finite element approach to a steady convection-diffusion problem: the distinction is that we limit the class of test functions to piecewise constants on the quadrilaterals.

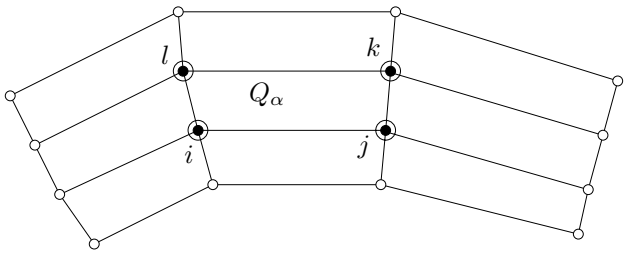


Figure 3.7. Typical quadrilateral mesh for a cell-vertex approximation to the Navier–Stokes equations, showing the vertices contributing to the cell residual: solid circles where fluxes are calculated, open circles at points needed for gradients.

A typical mesh is shown in Figure 3.7 with the cell Q_α over which the equations are to be integrated marked with solid circles. Thus, writing the integral of the hyperbolic conservation law (2.1) over this cell in a similar form to (3.1), we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{A}(Q_\alpha)\mathbf{u}(t) = -\frac{1}{|Q_\alpha|}\int_{\partial Q_\alpha}[\mathbf{f}_1\,\mathrm{d}x_2 - \mathbf{f}_2\,\mathrm{d}x_1], \tag{3.11}$$

where we have used the usual form for the boundary integral in two dimensions; and we have a similar form for the Navier–Stokes equations (2.19).

The objective is to generate a scheme of second-order accuracy for the steady problem and this can be achieved if each quadrilateral is within $\mathcal{O}(h)$ of a parallelogram, *i.e.*, the orientations of opposite sides differ by this order, and this we will assume. Then the approximation can be regarded as a bilinear form determined by the vertex values; and a recovery procedure is needed only to calculate the gradients that appear in the viscous fluxes. This may be done in several ways, but with this assumption on the mesh it is best to do so by integrating over each cell to give the gradients at the centroids, and then interpolating between these to obtain values at the vertices.

Writing $\mathbf{U}_i$ for the approximation to $\mathbf{u}$ at the vertex $\mathbf{x}_i$, we use the more compact notation in the Navier–Stokes equations (2.19),

$$\mathbf{F}_{i,\ell} = \mathbf{f}_\ell(\mathbf{U}(\mathbf{x}_i)) - (1/\mathrm{Re})\mathbf{g}_\ell^{(r)}(\mathbf{U}(\mathbf{x}_i))$$

where the superscript in $\mathbf{g}^{(r)}$ signifies the fact that the recovered gradient has been used in the calculation of the viscous fluxes. We also approximate the boundary integrals of (3.11) by the trapezoidal rule. Then it is one of the attractive features of the cell-vertex method in two dimensions that only the components of the quadrilateral diagonals appear in the residual. So, writing $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and with the vertex lettering in Figure 3.7, to solve the

steady problem we seek to satisfy the cell residual equations, which become

$$\mathbf{R}_\alpha := \tfrac{1}{2}\big[(\mathbf{F}_{i,1}-\mathbf{F}_{k,1})\mathbf{x}_{jl,2}+(\mathbf{F}_{j,1}-\mathbf{F}_{l,1})\mathbf{x}_{ki,2} - (\mathbf{F}_{i,2}-\mathbf{F}_{k,2})\mathbf{x}_{jl,1}-(\mathbf{F}_{j,2}-\mathbf{F}_{l,2})\mathbf{x}_{ki,1}\big]=\mathbf{0}. \tag{3.12}$$

This is a difficult system to solve and, although some of the techniques that are used are properly topics for the next section, several introductory remarks are in order here.

The most direct approach to solving the system (3.12) would be to apply Newton's method, or a quasi-Newton method. And for the incompressible Navier–Stokes equations or similar systems, and with closely related finite element approximations, this is widely used very successfully: see, for example, Winters, Rae, Jackson and Cliffe (1981). In addition, if the unsteady problem were modelled by approximating in the same way the full divergence form as in (2.12), we would have a system that directly generalizes the box scheme of (1.2), which is very successfully used in conjunction with Newton's method in one-dimensional river flow modelling: see Cunge, Holly and Verwey (1980). Aerodynamic applications involving high-speed compressible flows pose much more severe problems, however, particularly in the neighbourhood of shocks. Thus, even though some progress has been made in using Newton's method, as reported in Badcock and Richards (1995), steady cell-vertex approximations have generally been obtained with iteration schemes that relate closely to time-stepping; the one important distinction is that different time steps can be used at each point in order to improve convergence rates. Nevertheless, in future developments we may expect greater use of Newton methods, and progress in this direction is discussed in Section 4.5.

In most of the flow region the inviscid flux terms dominate, so modelling the Euler equations highlights many of the difficulties. The first of these is that the discrete equations, based on the cells, do not match up with the unknowns, based on the vertices. Even having the correct number of equations depends on the careful imposition of boundary conditions. Then the resulting equations are far from diagonally dominated when linearized. Thus most iteration procedures are based on combining the residuals from the cells surrounding a vertex to form a nodal residual, which is what is actually driven to zero. So we introduce *distribution matrices* $D_{\alpha,i}$ and define nodal residuals by

$$\mathbf{N}_i(\mathbf{U}) := \frac{\sum_{\alpha=1}^{p}|Q_\alpha|D_{\alpha,i}\mathbf{R}_\alpha}{\sum_{\alpha=1}^{p}|Q_\alpha|}, \tag{3.13}$$

where p is the number of cells meeting at node i, normally 4. An important particular choice of the distribution matrices is closely related to the most widely used two-dimensional form of the Lax–Wendroff method, and

corresponds to that used in the pioneering paper by Ni (1982): for node i in Figure 3.7, we have

$$D_{\alpha,i} = I + \nu_C \frac{\Delta t_\alpha}{|Q_\alpha|}[\mathbf{x}_{jl,2} A_{\alpha,1} - \mathbf{x}_{jl,1} A_{\alpha,2}], \tag{3.14}$$

where ν_C is a cell-based global CFL number, Δt_α a local time step and $A_{\alpha,\ell}\,, \ell = 1, 2$, are the Jacobian matrices of the inviscid fluxes evaluated at the centre of cell Q_α. Then the basic iteration can be written as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \nu_N \Delta t_i \mathbf{N}_i(\mathbf{U}), \tag{3.15}$$

where Δt_i is the minimum local time step from the surrounding cells and ν_N a node-based global CFL number. Such a scheme was applied successfully to solving the Navier–Stokes equations around an aerofoil in Crumpton *et al.* (1993), using a multigrid acceleration procedure based on a standard full approximation scheme. It was shown that for model problems the CFL parameters should satisfy

$$\nu_N \leq \nu_C \quad \text{and} \quad \nu_N \nu_C < 1;$$

and for the Navier–Stokes problems the role of ν_N was to control the rate of convergence of the iteration, while the value of ν_C affected the quality of the converged approximation: in particular, the extent to which the cell residuals were driven to zero rather than just the nodal residuals.

There are two further difficulties that affect these methods: the first is the presence of a spurious *chequer-board mode*; and the second is that the continuous form of the approximation is not well suited to representing shocks. The chequer-board mode arises from the averaging in the trapezoidal rule, and will be stimulated by the presence of shocks or other rapidly changing flow features to give severe oscillations in both directions of a typical mesh. These necessitate the addition of carefully chosen dissipation terms, which are critical to the success of the method. The introduction of procedures to recognize the presence of shocks, and to provide a global fit for them, can be used for simple problems such as the inviscid transonic flow around an aerofoil treated in Morton and Paisley (1989); however, this is not very feasible for general problems.

These difficulties have meant that cell-vertex methods are not as widely used at present as cell-centre and node-centred schemes. However, there has been considerable interest in the last few years in the development of cell-vertex methods on triangles. The attraction of quadrilateral meshes is that globally there are as many cells as vertices, so one can hope to drive most of the cell residuals to zero, and the nodal residuals are introduced only to achieve that end. With triangles this approach is no longer feasible. Instead, in the same way that approximate Riemann solvers use the discrepancy between two neighbouring flux values to update the two states, so the

flux residual in a triangular cell is used to update the states at its three vertices. In his influential paper Roe (1981), Roe was already expressing this viewpoint and expanded on it in Roe (1982); subsequent collaboration with Deconinck (Deconinck, Roe and Struijs 1993) took the ideas much further, and a recent survey, Ricciutto, Csik and Deconinck (2005), summarizes the present position.

4. Evolutionary algorithms

The various formulations of the finite volume approximation in the spatial variables require differing approaches to the approximation in time. At one extreme we have the system of ODEs in (3.4) that require a careful choice of ODE solvers. At another we have the system of nonlinear algebraic equations (3.12), which would be only slightly modified in an implicit time-stepping of an unsteady problem, and this needs special solution algorithms. And between these we have methods which use explicit time-stepping integrated into the spatial discretization. We will briefly survey each of these, starting with the need to define numerical flux functions for the cell-centre schemes, which leads naturally into considering first explicit, and then implicit, time-stepping algorithms.

4.1. Numerical flux functions

It is clearly not feasible to solve the generalized Riemann problem at all cell boundaries on a triangular or quadrilateral mesh, that is, to take account of mesh corners and the possible variation of the recovered approximation $\widetilde{\mathbf{U}}$ along each boundary. We therefore have to consider the construction of approximate flux functions to substitute into the typical scheme (3.4). A useful starting point is the scalar problem and Brenier's *transport collapse operator* (Brenier 1984). The exact solution of

$$u_t + f_1(u)_{x_1} + f_2(u)_{x_2} = 0$$

carries the initial data along the characteristics until shocks form: but in the transport collapse operator approximation this data is carried forward to give multivalued solutions at each point in space, and then these are combined to give the approximation. Brenier showed that repeated application of this process over small time steps converges to the the correct solution of the PDE as the time steps are refined; indeed, it provides one of the simplest means of establishing Theorem 2.1. He also showed that in one dimension it could lead directly to the Engquist–Osher scheme.

Suppose that in this one-dimensional case we have a (possibly recovered) approximation $\widetilde{U}^n(x)$ at time t^n, and that $f'(u) = a(u)$. Let

$$y = x + a(\widetilde{U}^n(x))\Delta t$$

denote the end point of the characteristic drawn from (x, t^n) through the time step Δt. Then it was shown in Lin, Morton and Süli (1993) that the transport collapse operator can be interpreted in terms of a Riemann–Stieltjes integral along the graph $[\widetilde{U}^n, y]$, and that it is thus equivalent to a characteristic-Galerkin method. Suppose the piecewise constant basis function on the mesh of Figure 1.1(b) is denoted by

$$\chi_j(x) = H_{j+1/2}(x) - H_{j-1/2}(x)$$

in terms of two Heaviside functions. Then an update algorithm that may include a recovery step can be written in the form

$$\begin{aligned} \Delta x_j (U_j^{n+1} - U_j^n) &= \int \widetilde{U}^n(x)[\chi_j(y)\,\mathrm{d}y - \chi_j(x)\,\mathrm{d}x] \qquad (4.1)\\ &= -\int \left[\int_x^{y(x)} \chi(s)\,\mathrm{d}s\right] \mathrm{d}\widetilde{U}^n. \end{aligned}$$

To interpret this as a finite volume method we have to make use of the relationship $a\mathrm{d}u = \mathrm{d}f$ and carry out the integral on the right to give a difference of flux functions: this has to be done with care when crossing a sonic point, for which $a(u) = 0$. Several examples can be found in Morton (2001), including the Engquist–Osher case in which there is no recovery stage.

Even more interesting is the two-dimensional case on a rectangular mesh. It is shown in Lin, Morton and Süli (1997) that, in addition to the one-dimensional flux differences along the sides of a mesh box arising from integrals of $a_1\mathrm{d}u$ and $a_2\mathrm{d}u$, there are also corner terms arising from integrals of $a_1(u)a_2(u)\mathrm{d}u$. With no recovery, on a uniform mesh and with the CFL conditions $0 \le a_1(U^n)\Delta t \le \Delta x_1$, $0 \le a_2(U^n)\Delta t \le \Delta x_2$ satisfied, the difference scheme that results from this method has the form

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} + \frac{\Delta_{-x_1} f_1(U_{i,j}^n)}{\Delta x_1} + \frac{\Delta_{-x_2} f_2(U_{i,j}^n)}{\Delta x_2} - \Delta t \frac{\Delta_{-x_1}\Delta_{-x_2} f_{12}(U_{i,j}^n)}{\Delta x_1 \Delta x_2} = 0, \qquad (4.2)$$

where the corner flux is given by

$$f_{12}(u) := \int^u a_1(v)a_2(v)\,\mathrm{d}v,$$

and Δ_{-x_ℓ} is the backward difference operator in the x_ℓ direction. The scheme is stable under the given conditions, but without the extra corner term the stability limit would be given by $a_1(U^n)\Delta t/\Delta x_1 + a_2(U^n)\Delta t/\Delta x_2 \le 1$.

A key point needs to be made about these formulae, which is particularly important in the two-dimensional case. They resulted from the development of unconditionally stable methods for hyperbolic problems in which shocks were not the key phenomena: thus they were not generally put in

their finite volume form and characteristics were tracked even into non-neighbouring cells before the projection was made to obtain the updated approximation. Atmospheric flows are a typical application area: see, *e.g.*, Staniforth and Côté (1991). Thus the significance of (4.2) is that the stable region in (x_1, x_2)-space is a mesh rectangle and other formulae are obtained for neighbouring rectangles so that the whole plane is covered to give an unconditionally stable scheme. This is clearly not so relevant to the general triangular and quadrilateral meshes that we are concentrating on. But what is relevant to note is that the corner terms represent an $\mathcal{O}(\Delta t)$ correction to the flux terms obtained along the edges: to omit them, as is normal with general finite volume methods, commits an error as well as limiting the stability: see LeVeque (2002) for a wider discussion on the importance of corner terms.

Formulae obtained from the transport collapse operator also provide valuable guidance on the modifications to the flux functions that arise from a recovery stage. For example, suppose the discontinuous linear recovery (3.6) is used in one dimension. Then substitution into (4.1) leads to a flux function of the same form as the Engquist–Osher flux (1.7) but with different terms: the quantities involved are obtained from the values of the recovered approximation just to the left and to the right of the interface so as to give

$$\widetilde{F}^{n+1/2}_{j+1/2} = \tfrac{1}{2}\big[(1+s^+_j)\widetilde{F}^+_j + (s^-_{j+1} - s^+_j)f(u_s) + (1-s^-_{j+1})\widetilde{F}^-_{j+1}\big]. \tag{4.3}$$

Here we will use the notation $\widetilde{U}^\pm_j$ for the value of the recovered variable at the right and left of cell j, and $s^\pm_j$ for the signs of the corresponding characteristic speeds $a(\widetilde{U}^\pm_j)$. It remains to define the two flux values. We consider only the simplest case, when the characteristic speeds are positive throughout cell j, and we need to calculate $\widetilde{F}^+_j$ from all the right-moving characteristics which reach the interface from the cell in one time step. For this purpose we assume that the characteristic speed also varies linearly throughout the cell, with a slope M^+_j. Then the speed at the point which is just carried to the interface at the end of the time step is given by

$$A^{*+}_j = a(\widetilde{U}^+_j)/[1 + M^+_j \Delta t]; \tag{4.4}$$

and a short calculation gives the following flux value:

$$\widetilde{F}^+_j = \frac{A^{*+}_j f(\widetilde{U}^+_j) + a(\widetilde{U}^+_j) f(\widetilde{U}^+_j - S_j A^{*+}_j \Delta t)}{A^{*+}_j + a(\widetilde{U}^+_j)}. \tag{4.5}$$

This average of two flux values gives a scheme which is second-order accurate in smooth flow regions. In Morton (2001) it is shown to be TV-stable, under CFL conditions which are principally of the standard form that characteristics cross no more than one cell in one time step, and where the recovery stage satisfies only the necessary monotonicity-preserving condition (3.8).

The same framework can be used to derive a third-order accurate method by means of a continuous piecewise parabolic recovery process similar to that introduced in the *Piecewise Parabolic Method* (PPM) of Colella and Woodward (1984). In the recovery process the key step is to deduce a value $\widetilde{U}_{j+1/2}$ at each interface, the parabola in each cell then following from the requirement that the cell average is preserved. One can then show that if the recovery stage is TVD then the scheme is TV-stable under our familiar CFL conditions: see Morton (2001).

The real challenge is to carry these ideas forward to systems of conservation laws, and in particular to the Euler equations. A breakthrough was achieved by Roe (1981) by using a local linearization that ensures that, from two states $\mathbf{u}_L, \mathbf{u}_R$ and a one-dimensional flux vector $\mathbf{f}(\mathbf{u})$, a matrix $\bar{A}(\mathbf{u}_L, \mathbf{u}_R)$ is constructed so as to satisfy

$$\bar{A}(\mathbf{u}_L, \mathbf{u}_R)(\mathbf{u}_R - \mathbf{u}_L) = \mathbf{f}(\mathbf{u}_R) - \mathbf{f}(\mathbf{u}_L). \tag{4.6}$$

The advantage of using any local linearization is that the interfacial flux can be computed straightforwardly from the waves corresponding to the eigenvalues and eigenvectors of the matrix $\bar{A}$. If these are given in the standard form $\bar{A} = R\Lambda R^{-1}$ and we define the *absolute value* of $\bar{A}$ by $|\bar{A}| = R|\Lambda|R^{-1}$, then we can write this flux very concisely as

$$\bar{F}(\mathbf{u}_L, \mathbf{u}_R) = \tfrac{1}{2}[\mathbf{f}(\mathbf{u}_R) + \mathbf{f}(\mathbf{u}_L)] - \tfrac{1}{2}|\bar{A}|[\mathbf{u}_R - \mathbf{u}_L], \tag{4.7}$$

although this is not the way in which it is usually coded. The advantage of a linearization satisfying (4.6) is that Riemann problems whose solution is a simple discontinuity (a shock or a contact) are solved exactly, because the Rankine-Hugoniot conditions are satisfied with the shock speed given by an eigenvalue of $\bar{A}$.

The *Roe matrix* is constructed for the Euler equations by observing that both $\mathbf{u}$ and $\mathbf{f}$ can be expressed as quadratic functions of a new variable $\mathbf{z}$ given by $\rho^{1/2}(1, v, H)^T$. Then one can exploit the identity

$$2(a_1 b_1 - a_2 b_2) \equiv (a_1 + a_2)(b_1 - b_2) + (b_1 + b_2)(a_1 - a_2)$$

to introduce matrices $\bar{B}, \bar{C}$ such that

$$\mathbf{u}_R - \mathbf{u}_L = \bar{B}(\mathbf{z}_R - \mathbf{z}_L) \quad \text{and} \quad \mathbf{f}(\mathbf{u}_R) - \mathbf{f}(\mathbf{u}_L) = \bar{C}(\mathbf{z}_R - \mathbf{z}_L),$$

from which one can define $\bar{A} = \bar{C}\bar{B}^{-1}$ to satisfy (4.6); the detailed form of these matrices can be found in texts such as Hirsch (1990). For obvious reasons such methods are called *flux difference splitting methods* and are computationally quite expensive; alternatively, numerical flux functions can be computed by *flux vector splitting methods* such as that due to Steger and Warming (1981) already referred to.

The Roe matrix gives a direct generalization of the upwind scalar scheme (1.6) and has similar disadvantages in its emphasis on capturing shocks.

In particular, convergence would not necessarily be to an entropy-satisfying solution of the PDE system. Thus, when such a flux function is used it is modified by some form of *entropy fix*. Alternatively, one may seek to approximate rarefaction waves as in the method proposed in Osher and Solomon (1982) which generalizes the Engquist–Osher flux of (1.7). For a comprehensive review of schemes that generalize upwind differencing to systems of conservation laws, described in the context of the underlying theory, see Harten, Lax and van Leer (1983). The standard CFD text Hirsch (1990) also describes many widely used schemes. There are many desirable properties that numerical flux functions should have, such as ensuring that the density and pressure are always non-negative, and a discussion of such issues can be found in the recent survey by Roe (2001).

4.2. Evolution-Galerkin methods and error analysis

A useful general framework for considering the approximation of an evolutionary problem by finite difference, finite element or finite volume methods has the following form. Suppose that in a given function space V the operator $\mathcal{E}_\Delta$ represents an approximation to the true evolution operator through a time step Δt; let U^n be an approximation in some discrete subspace S^h of V to the exact solution $u(\cdot, t^n)$ at time t^n, and $\mathcal{P}$ a projection from V to that discrete space; finally, let $\mathcal{R}$ be a recovery operator giving $\widetilde{U}^n = \mathcal{R}U^n$ as a recovered approximation in some larger discrete subspace of V. Then one step of an evolution-Galerkin method can be written in the alternative forms

$$U^{n+1} = \mathcal{P}\mathcal{E}_\Delta \mathcal{R} U^n \quad \text{or} \quad \widetilde{U}^{n+1} = \mathcal{R}\mathcal{P}\mathcal{E}_\Delta \widetilde{U}^n. \tag{4.8}$$

All of the methods we have described can be put into this form, although we have not specified the defining operators. We are interested in the error between the true solution and the recovered approximation, which we estimate by decomposing it into two parts through the introduction of a *target approximation* $u^n \in S^h$: we call $\eta^n = u(\cdot, t^n) - \mathcal{R}u^n$ the *projection error* and $\xi^n = \mathcal{R}u^n - \widetilde{U}^n$ the *evolutionary error*, so that we have

$$\begin{aligned} u(\cdot, t^n) - \widetilde{U}^n &= [u(\cdot, t^n) - \mathcal{R}u^n] + [\mathcal{R}u^n - \widetilde{U}^n] \\ &=: \eta^n + \xi^n. \end{aligned} \tag{4.9}$$

An appropriate choice of the target approximation can be important when comparing differing types of method on non-uniform meshes.

In order to estimate the evolutionary error, we make use of (4.8) to write

$$\xi^{n+1} \equiv \mathcal{R}u^{n+1} - \widetilde{U}^{n+1} = [\mathcal{R}u^{n+1} - \mathcal{R}\mathcal{P}\mathcal{E}_\Delta \mathcal{R}u^n] + [\mathcal{R}\mathcal{P}\mathcal{E}_\Delta \mathcal{R}u^n - \mathcal{R}\mathcal{P}\mathcal{E}_\Delta \widetilde{U}^n], \tag{4.10}$$

and define a *truncation error* as

$$\widetilde{T}^n := (\Delta t)^{-1}(\mathcal{R}u^{n+1} - \mathcal{R}\mathcal{P}\mathcal{E}_\Delta \mathcal{R}u^n). \tag{4.11}$$

Now suppose that the method is strongly stable in the sense that

$$\|\mathcal{R}\mathcal{P}\mathcal{E}_\Delta \widetilde{U} - \mathcal{R}\mathcal{P}\mathcal{E}_\Delta \widetilde{V}\| \le \|\widetilde{U} - \widetilde{V}\|. \tag{4.12}$$

Then we have the familiar dependence of the evolutionary error on the truncation error,

$$\|\xi^{n+1}\| \le \|\xi^n\| + \|\widetilde{T}^n\|\Delta t. \tag{4.13}$$

For a well-posed problem one should expect that $\mathcal{P}\mathcal{E}_\Delta$ will be strongly stable, so the stability assumption here is mainly a constraint on the recovery process.

The linear advection equation $u_t + au_x = 0$, with a a positive constant, is the most illuminating first test case for any proposed approximation scheme for hyperbolic problems. So it is here. Suppose that on a non-uniform mesh the upwind Roe scheme (1.3) with (1.6) is applied, using the piecewise constant approximation $U^n \equiv \{U_j^n\}$. Then, if the target approximation is based on cell averages, the truncation error will depend on the ratio of the distance between two cell centres and the length of one of the cells, so the scheme would be deemed inconsistent with the PDE, as has been observed by many researchers. On the other hand, suppose we take the target approximation $u^n \equiv \{u_j^n\}$ such that u_j^n is the value of the true solution at the upwind end of the cell, namely $u(x_{j+1/2}, t^n)$, which still gives a first-order projection error. Now the truncation error (4.11) with no recovery has the familiar form

$$T_j^n = \frac{u(x_{j+1/2}, t^{n+1}) - u(x_{j+1/2}, t^n)}{\Delta t} + \frac{a[(u(x_{j+1/2}, t^n) - u(x_{j-1/2}, t^n)]}{x_{j+1/2} - x_{j-1/2}}$$

and is clearly of first order.

In Morton (1998) this analysis is continued to show that discontinuous linear recovery gives a second-order error if the grading of the mesh and the change in solution slope are smooth; and it is also shown that continuous quadratic recovery as in the PPM scheme gives third-order accuracy under similar mesh restrictions. The target approximations in these cases are formed by truncating the Taylor expansion of the cell average of the true solution about the upwind end of the cell.

The linear advection equation also provides a convenient framework for considering both the order of accuracy best aimed for, and whether that should be attained from choice of the order of accuracy of the basic approximation U^n, or from the recovery process giving $\widetilde{U}^n$. Moreover, approximating this equation on a uniform mesh means that the powerful tool of Fourier analysis is available. This shows that schemes with an even order of accuracy propagate waves with an error dominated by dispersion; while schemes with an odd order have waves dominated by dissipation. For example, the best-known second-order scheme is the Lax–Wendroff method,

which notoriously suffers from a trail of oscillations when used to approximate the advection of a discontinuity; while the first-order upwind scheme suffers from severe damping for the same problem, but has surprisingly small dispersion errors; see Morton and Mayers (2005) for illustrations of these phenomena and Morton (1998) for a more general discussion.

Thus, for wave propagation problems, such as those arising in meteorology or oceanography, third-order accuracy has been advocated by many authors, such as Leonard (1991). Moreover, use of the *characteristic-Galerkin method* with the standard continuous piecewise linear finite element basis yields such a scheme (see Lesaint (1977) and Douglas and Russell (1982)); and such methods have been widely used in finite element approximations of the incompressible Navier–Stokes equations (see Pironneau (1982)). Thus, suppose we approximate the convection-dominated diffusion problem $u_t + au_x = \epsilon u_{xx}$ on a uniform mesh, with linear finite elements, in the following way. From each mesh point at the time level t^n we draw the characteristic forward to the time level t^{n+1}, and suppose that

$$\nu := a\Delta t/\Delta x = m + \hat{\nu} \quad \text{with} \quad m \in \mathbb{N},\ 0 < \hat{\nu} \leq 1.$$

Then the projection of the approximate evolution operator, denoted by $\mathcal{PE}_\Delta$ above, is defined by carrying values along these characteristics from one time level to the next, where the diffusion is applied, and using the Galerkin projection to determine the new piecewise linear approximation. In finite difference notation, with $\mu = \epsilon\Delta t/(\Delta x)^2$, we obtain the scheme

$$\begin{aligned}[1 + (\tfrac{1}{6} - \mu)\delta^2]U_j^{n+1} = [(1 + \tfrac{1}{6}\delta^2) - \hat{\nu}\Delta_0 \\ + \tfrac{1}{2}\hat{\nu}^2\delta^2 - \tfrac{1}{6}\hat{\nu}^3\delta^3\Delta_-]U_{j-m}^n;\end{aligned} \tag{4.14}$$

here Δ_-, Δ_0, δ^2 are the first-order backward, the first-order central and the second-order central differences, all undivided. This scheme is unconditionally stable and third-order accurate in the convection terms; and of course it is readily generalized to more space dimensions: indeed, with a triangular or quadrilateral mesh, though it would not then be expressed in difference form. It thus provides an extremely valuable yardstick against which to measure alternative schemes.

Now the piecewise constant basis function is a first-order B-spline, and on a uniform mesh higher-order B-splines are generated by a recurrence relation:

$$\chi^{(p)}(s) := \int \chi^{(p-1)}(\sigma - s)\chi(\sigma)\,\mathrm{d}\sigma, \tag{4.15}$$

where $\chi(\sigma) \equiv \chi^{(1)}(\sigma)$ is defined as the characteristic function of the interval $[-\frac{1}{2}, \frac{1}{2}]$; and the linear basis functions are second-order B-splines. Moreover, differentiating a spline of a given order generates splines of a lower

order; and integrating the product of two splines generates higher-order splines. Thus all the terms in (4.14) could be generated from other than linear basis functions: in particular, the convection terms giving the third-order accurate scheme could be generated from using a piecewise constant approximation U^n that is recovered by quadratic splines to give $\widetilde{U}^n$.

To make this last statement more precise, we are presuming that a non-adaptive recovery process is used that defines the quadratic spline recovered approximation $\widetilde{U}^n$ by maintaining the cell averages, that is, by specifying its inner products against first-order splines; and it is these inner products exactly equalling inner products between two linear splines that leads to the equivalence of the two formulations. So in this case there is no loss of accuracy in using a piecewise constant basic approximation followed by recovery with a high-order spline, compared with using higher-order basis functions for U^n and no recovery; and this is a general conclusion. Moreover, with the former approach, which is the basis of many of our finite volume methods, we have the opportunity to make the recovery stage adaptive so as to maintain key properties of the solution, as we have already described. A fuller discussion of these points and some numerical illustrations can be found in Morton (1996).

4.3. Finite volume evolution-Galerkin methods

The linear advection equation is a reasonable model problem for devising and analysing algorithms used to solve hyperbolic problems that are dominated by a single velocity field. But it is inadequate for the equations of unsteady gas dynamics. For these the two-dimensional wave equation system is more appropriate: we write it as

$$\begin{aligned} \phi_t + c(u_x + v_y) &= 0, \\ u_t + c\phi_x = 0, \qquad & v_t + c\phi_y = 0, \end{aligned} \tag{4.16}$$

where ϕ can be regarded as a pressure and u, v as the velocity components. The classical Kirchhoff solution of the wave equation, written as a single second-order equation, is in the form of an integral over the base of a characteristic cone with its apex at the sample point: the solution at the apex is given in terms of the data over the base. However, Butler (1960) developed an alternative form for the system (4.16), which he used to good effect in approximating the Euler equations. In his form the data on the perimeter of the base is used, but he also uses an integral over the mantle of the cone, that is, involving the solution at intermediate times. Unfortunately, his method did not make use of the data in a very consistent way and it was quickly superseded by the method of Lax and Wendroff (1960); but with the help of finite element and finite volume formulations it can be used as the basis of powerful methods.

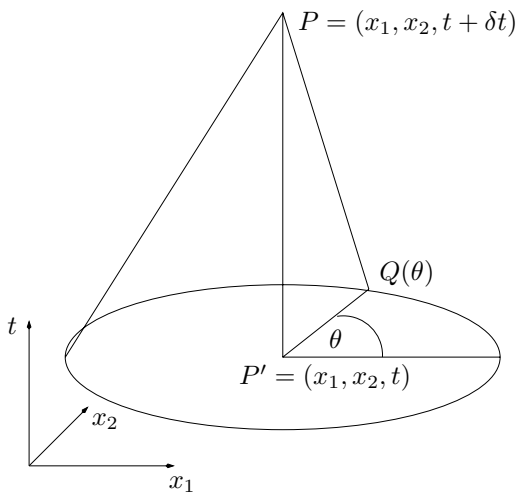


Figure 4.1. Characteristic cone for the wave equation in 2D.

Suppose one integrates a $(1, -\cos\theta, -\sin\theta)$ linear combination of the equations (4.16) along each bicharacteristic that generates the characteristic cone, centred at $P \equiv (\mathbf{x}, t + \delta t)$, and averages the result over θ; then one obtains an integral equation for ϕ that, in the notation of Figure 4.1, has the following form:

$$\phi_P = \frac{1}{2\pi}\int_0^{2\pi} [\phi_Q - u_Q\cos\theta - v_Q\sin\theta]\,\mathrm{d}\theta \qquad (4.17)$$
$$- \frac{1}{2\pi}\int_t^{t+\delta t}\int_0^{2\pi} S(t', \theta)\,\mathrm{d}\theta\,\mathrm{d}t',$$

where

$$S(t', \theta) = c[u_x \sin^2\theta - (u_y + v_x)\sin\theta\cos\theta + v_y\cos^2\theta] \qquad (4.18)$$

and (u, v) are evaluated at Q' at the intermediate time t'. Similar equations can be derived for (u_P, v_P). If the integral over t' is approximated by the rectangle rule at time level t, or the trapezoidal rule, one obtains an approximate evolution operator over a full time step by taking $\delta t = \Delta t$. This was used as the basis of various evolution Galerkin methods on a square mesh in Lukáčová-Medviďová, Morton and Warnecke (2000). The advantage of such methods was seen to lie in the possibility that, by using all characteristic directions, they would propagate wave fronts which were not distorted badly by the mesh orientation. With exact integrals over piecewise constant approximations, this hope was realized; but such methods could only be first-order accurate and did not have a good stability range.

To do better, four steps are necessary: recovery by discontinuous bilinear approximations on the square mesh, use of a more general evolution operator derived by Ostkamp (1997), adoption of a finite volume framework and better approximation of the mantle integrals. All of these steps are described in Lukáčová-Medviďová, Morton and Warnecke (2004) and were applied to the Euler equations in Lukáčová-Medviďová, Morton and Warnecke (2002). The finite volume form (2.11) was used, with the conservation form of the equations, which has the advantage that the evolution operator is needed only to evaluate the solution on the perimeter of the control volume; and in the case of the Euler equations the equations for the primitive variables were used for this purpose. To obtain a second-order accurate method one need use only the mid-point rule for the time integration, so the approximate evolution operator is needed only at time $t + \frac{1}{2}\Delta t$ and at the quadrature points used for the integral around the control volume perimeter.

To derive the general evolution operator for a hyperbolic system such as (2.2), we suppose that the linear combination $A(\boldsymbol{\nu})$ of Jacobian matrices in the direction $\boldsymbol{\nu}$, given by (2.3), has the matrix of right column eigenvectors $R(\boldsymbol{\nu})$. Then, if we apply the corresponding transformation to each individual Jacobian matrix, in general they will not all be diagonalized: writing D_ℓ for the diagonal part and B'_ℓ for the remainder, we have

$$R^{-1}A_\ell R = D_\ell + B'_\ell.$$

We also introduce the corresponding characteristic variables $\mathbf{w} \equiv \mathbf{w}(\boldsymbol{\nu})$ given by $\partial\mathbf{w} = R^{-1}\partial\mathbf{u}$. Then, operating on the differential equation with R^{-1} from the left, we get

$$\partial_t\mathbf{w} + \sum_{\ell=1}^{d} D_\ell\partial_{x_\ell}\mathbf{w} = -\sum_{\ell=1}^{d} B'_\ell\partial_{x_\ell}\mathbf{w} =: \mathbf{S}. \tag{4.19}$$

It is this equation that is integrated along the bicharacteristic corresponding to the direction $\boldsymbol{\nu}$ and the 'source' term on the right that leads to the mantle integral when the result is averaged over all directions.

For the wave equation the resulting formula for ϕ is the same as that given by Butler but that for the velocity components is different. Thus we have the following formulae, after an integration by parts in the mantle integrals over θ to remove the derivatives on the dependent variables:

$$\begin{aligned}\phi_P = \frac{1}{2\pi}\int_0^{2\pi} [\phi_Q - u_Q\cos\theta - v_Q\sin\theta]\,\mathrm{d}\theta \\ - \frac{1}{2\pi}\int_0^{\delta t}\frac{1}{\tau}\int_0^{2\pi} [u_{Q'}\cos\theta + v_{Q'}\sin\theta]\,\mathrm{d}\theta\,\mathrm{d}\tau,\end{aligned} \tag{4.20}$$

and

$$u_P = \frac{1}{2\pi}\int_0^{2\pi} [-\phi_Q \cos\theta + u_Q \cos^2\theta + v_Q \sin\theta\cos\theta]\,\mathrm{d}\theta \qquad (4.21)$$
$$+ \frac{1}{2\pi}\int_0^{\delta t}\frac{1}{\tau}\int_0^{2\pi} [u_{Q'}\cos 2\theta + v_{Q'}\sin 2\theta]\,\mathrm{d}\theta\,\mathrm{d}\tau$$
$$+ \tfrac{1}{2}u_{P^0} - \tfrac{1}{2}c\int_0^{\delta t} \partial_{x_1}\phi_{P'}\,\mathrm{d}\tau,$$

with a similar formula for v_P, where P^0 is the centre of the cone base. Note that in the scheme given below we take $\delta t = \frac{1}{2}\Delta t$.

The integrals at the old time level are carried out exactly for either the unrecovered piecewise constant approximation $\mathbf{U}^n$, or its discontinuous bilinear recovered counterpart $\widetilde{\mathbf{U}}^n$. It is the approximation of the mantle integrals, that involve values of the solution at intermediate times, that are crucial to both the accuracy and stability of methods based on these formulae. Fortunately, there is one case when these integrals can be evaluated exactly in terms of the known data: that is, when that data represents waves that are one-dimensional so that we can use the familiar d'Alembert formula. Lukáčová-Medvi̇ďová *et al.* (2004) used this, both for piecewise constant and continuous piecewise linear data, to relate the mantle integrals to those round the perimeter of the cone base. Substituting the result

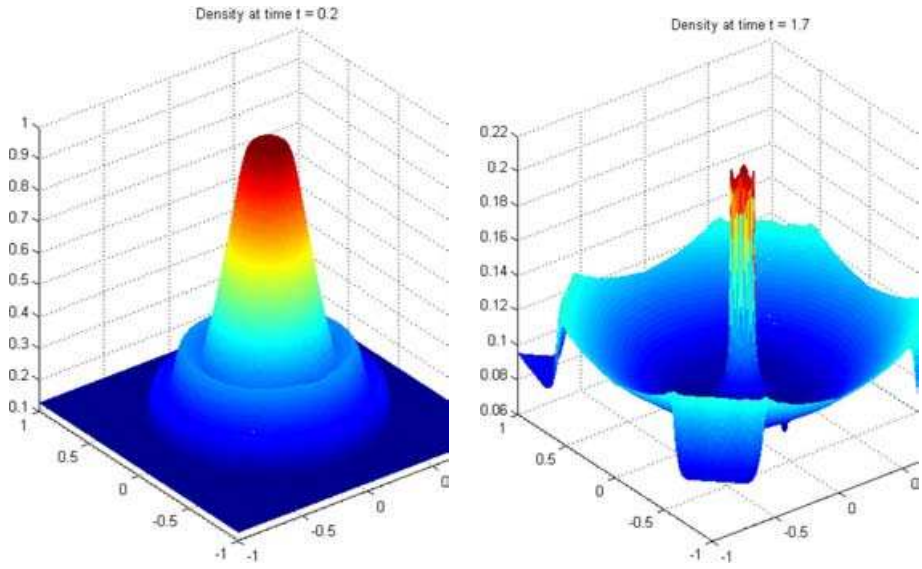


Figure 4.2. Solution to the 2D Sod test case.
Density at time $t = 0.2$ (*left*) and at $t = 1.7$ (*right*).

into the finite volume framework on a rectangular mesh, gives a scheme for updating the cell averages which corresponds to (2.11) and takes the form

$$|R_i|(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n) + \Delta t \oint_{\partial R_i} \mathcal{F}(\mathcal{E}_\delta \mathcal{R}\mathbf{U}^n) \cdot \mathbf{n}\, \mathrm{d}s = \mathbf{0}, \tag{4.22}$$

where R_i is a mesh rectangle and $\mathcal{E}_\delta$ is the approximate evolution operator over half a time step as just described, and $\mathcal{R}$ represents the recovery operator. Using Simpson's rule for approximating the integrals around the cell perimeter, the resulting methods are second-order accurate, being some five times more accurate than the comparable Lax–Wendroff method, and have good stability properties. More importantly, when applied to the Sod-2D test problem involving cylindrically symmetric wave propagation and reflection, it preserves the symmetry very precisely: see Figure 4.2 and the results in Lukáčová-Medvidová *et al.* (2004).

4.4. Semi-discrete explicit time-stepping algorithms

The traditional discretization of hyperbolic equations was based on methods using finite differences in space and time, with explicit time differencing. They are simple to implement and the CFL stability limit on the time step is commonly consistent with the requirements of accuracy. However, we have already seen several situations when these arguments break down: when a steady state is sought, or the flow rate of change is much slower than important characteristic speeds; and at sharp corners in the mesh where the ideal scheme corresponding to (4.2) cannot be used and the CFL limit is drastically reduced. Moreover, when grid adaptivity is introduced in Section 6 a uniform explicit time step may be quite inappropriate. In such situations it may be preferable to consider the space discretization quite separately from that in time, not attempting in any way to have the fluxes represent averages over a time step, as is implied by the notation introduced for the Godunov method in (1.3) and which led naturally to the complications faced by the FVEG methods described in the previous subsection; instead, we seek methods to solve large systems of ODEs such as that given by (3.4).

One-step methods, such as Runge–Kutta schemes, are clearly attractive for this purpose and their use goes back to the very influential paper of Jameson, Schmidt and Turkel (1981). The behaviour of the methods, and hence the selection of the most appropriate, can be studied by considering model hyperbolic systems and applying Fourier analysis in the spatial variables. Then, in a standard stability region plot, it is clear that it is the behaviour along the imaginary axis and just to its left that is most important. So in that paper Jameson *et al.* used the standard explicit fourth-order Runge–Kutta scheme which is stable for a CFL number up to $2\sqrt{2}$ when

upwind differencing is applied to the linear advection equation, and allows for a reasonable amount of damping to be added.

However, Shu and Osher (1988) observed that this scheme does not preserve monotonicity properties that may have been built into the spatial discretization. They therefore introduced special TVD-Runge–Kutta schemes that preserve the properties of ENO-type schemes at the cost of reduced stability ranges. If we write (3.4) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_i(t) = -\mathcal{N}_i(\mathbf{U}(t)),$$

and temporarily suppress the subscripts, then a typical third-order scheme has the form

$$\begin{aligned}
\mathbf{U}^{(0)} &:= \mathbf{U}(t), \qquad (4.23)\\
\mathbf{U}^{(1)} &= \mathbf{U}^{(0)} - \Delta t\mathcal{N}(\mathbf{U}^{(0)}),\\
\mathbf{U}^{(2)} &= \mathbf{U}^{(0)} - \tfrac{1}{4}\Delta t\mathcal{N}(\mathbf{U}^{(0)}) - \tfrac{1}{4}\Delta t\mathcal{N}(\mathbf{U}^{(1)}),\\
\mathbf{U}^{(3)} &= \mathbf{U}^{(0)} - \tfrac{1}{6}\Delta t\mathcal{N}(\mathbf{U}^{(0)}) - \tfrac{1}{6}\Delta t\mathcal{N}(\mathbf{U}^{(1)}) - \tfrac{2}{3}\Delta t\mathcal{N}(\mathbf{U}^{(2)}),\\
\mathbf{U}(t+\Delta t) &:= \mathbf{U}^{(3)}.
\end{aligned}$$

But then such a scheme has a CFL limit reduced to unity for the above model problem.

4.5. Implicit time-stepping

Considerations such as those outlined in the previous subsection lead to the conclusion that implicit time-differencing is bound to play a larger role in future methods, either in the semi-discrete formulation introduced there or in a fully discrete formulation. This is despite the difficulties posed by the resulting large systems of highly nonlinear equations that such methods will lead to. There are two major hurdles to overcome: formulation of the equations to ensure convergence of the Newton or quasi-Newton iterations that are needed; and rapid solution of the linear equations at each iteration. Hyperbolicity of the equations being approximated ensures the underlying Jacobians are well behaved with finite eigenvalues and a full set of eigenvectors. So, if care is taken to reflect properly the properties of the differential equations in their discretization, attention can often be concentrated on the solution of the linear equation systems.

A natural starting point for considering these issues would seem to be provided by the Preissmann box scheme applied to the St. Venant equations for one-dimensional river flow, where solution of the global Newton system is the standard procedure for subcritical flows. However, it has long been recognized that this formulation runs into difficulties when flows develop a supercritical section. Thus we will start even more simply with the scalar

problem of the inviscid Burgers equation and initial data that leads to a shock; and we consider not only the box scheme, as the simplest cell-vertex method, but also later an implicit cell-centre or node-centred scheme.

Suppose that, on the interval $0 \leq x \leq 1$, initial data $u^0(x)$ are given with $u^0(0) = u_L > 0$ and $u^0(1) = u_R < 0$. Then boundary conditions need to be imposed at both these boundaries in order to solve the problem for $t > 0$; and if these values continue to be imposed, eventually a shock will form. Now divide the interval into J cells with the points $x_0 = 0, x_1, \ldots, x_J = 1$, and suppose first that we approximate the problem with the box scheme (1.2). As there are J cells, there will be J box equations and two boundary conditions to be satisfied by the $J + 1$ unknown nodal values; clearly, something has to be sacrificed. To clarify the choice, suppose that u^0 has the constant value u_L to the left of some point $x = x_S \in (x_k, x_{k+1})$ and u_R to the right: then, in carrying out the first time step, we can set $U_0^1 = u_L$ and work from left to right using each cell residual equation to calculate successively $U_1^1, U_2^1, \ldots, U_k^1$ until the shock is reached; and we can do the same from the right for $U_J^1, U_{J-1}^1, \ldots, U_{k+1}^1$. Thus all the nodal values at the new time level could be calculated in this way without having to use the box equation for the cell containing the shock. But this would violate the basic conservation law for u and is unacceptable; and, in fact, the same problem would occur if we were to discretize the initial data.

Apart from the boundary conditions, this overall conservation property is the most important consideration. We satisfy it by means of a general algorithm which we will refer to as a *residual distribution scheme*: these ideas were originally put forward in Roe (1982) and Deconinck *et al.* (1993) in the context of explicit time-stepping algorithms, and in Crumpton *et al.* (1993) to derive iteration procedures for steady flow problems; but they are equally applicable to the tasks of discretizing the initial data or setting up implicit time-stepping equations. For a general scalar one-dimensional problem, we suppose that we have for each cell a residual $R_{j+1/2}$ and an average or representative characteristic speed $a_{j+1/2}$. Then we execute the following two steps:

- for each cell, allocate the residual $R_{j+1/2}$ to node j if $a_{j+1/2} \leq 0$ or to node $j + 1$ if $a_{j+1/2} > 0$;
- for each node, set up the appropriate nodal equation using the sum of the residuals that have been allocated to it.

For our present Burgers' equation problem, let us first consider the discretization of the initial data containing a shock by a continuous piecewise linear approximation satisfying the two boundary conditions. The integral of $u^0(x)$ over each cell gives the cell residual; and in the situation described above, we would clearly have average characteristic speeds that were positive (*i.e.*, supercritical) for the cells to the left of x_k, and negative (*i.e.*, sub-

critical) for the cells to the right of x_{k+1}. The only issue is with the shock cell: should its residual be combined with that from the left or the right? Let us suppose the latter. Then, by setting the residuals to zero, we would set $U_j^0 = u_L$ for $j = 0, 1, \ldots, k$ and $U_j^0 = u_R$ for $j = J, J-1, \ldots, k+2$; and from the combined residual from the cells either side of $k+1$, we obtain the following equation for U_{k+1}^0:

$$\tfrac{1}{2}(u_L + U_{k+1}^0)(x_{k+1} - x_k) + \tfrac{1}{2}(U_{k+1}^0 + u_R)(x_{k+2} - x_{k+1}) \\ = u_L(x_S - x_k) + u_R(x_{k+2} - x_S).$$

This gives

$$U_{k+1}^0 = \frac{(2x_S - x_k - x_{k+1})u_L + (x_{k+1} + x_{k+2} - 2x_S)}{x_{k+2} - x_k}.$$

Now it is clearly important that U_{k+1}^0 should lie between u_L and u_R, a condition required to maintain the TVD property when setting up an evolution step; and this requires that

$$\tfrac{1}{2}(x_k + x_{k+1}) \le x_S \le \tfrac{1}{2}(x_{k+1} + x_{k+2}).$$

The first inequality implies that the shock cell residual should be combined with that from the cell to the right when the shock is closest to the boundary with that cell, at x_{k+1}; in other words, when the shock cell is dominated by u_L values and therefore considered to be supercritical. Correspondingly, the second inequality implies that these two cell residuals should be combined when the shock is in the cell (x_{k+1}, x_{k+2}) and closest to the left-hand boundary, so that it is subcritical.

These later interpretations are the key to determining how to carry out an evolution step both for this problem and more generally. For each cell we calculate the box scheme residual and allocate it to the node to its left or right according to whether it is regarded as being a subcritical or supercritical cell: and this is determined by an average characteristic speed calculated from the current solution approximation. Thus in the neighbourhood of a shock we always combine the residual of a cell which is deemed to be subcritical with one deemed supercritical.

There is also the complementary situation to consider in which the initial data, or current solution, is subcritical on the left and supercritical on the right, so that there is a *sonic point* or *critical point* at some point of the interval. Then, for example with $u_L < 0$ and $u_R > 0$, boundary conditions are not imposed on the left or the right so that now there are too few equations provided by the residuals to determine all the unknowns. The solution is to split the residual for the cell containing the sonic point at that point: for the Burgers' equation problem, either to discretize the initial data or to evolve the solution, values are obtained successively by working out from the sonic point to the boundaries. Incidentally, it is worth noting that

this device corresponds to how the Engquist–Osher scheme of (1.7) breaks up the flux differences near a sonic point.

In Freitag and Morton (2007) these two techniques were applied to extend the Preissmann box scheme to solve St. Venant equation problems with a supercritical section. The mass residual was that chosen to be either split at a sonic point or combined at a shock; and the resultant system of equations was shown to be well behaved when solved by Newton's method combined with the standard Thomas algorithm applied to the resultant block tridiagonal system of linear equations.

However, these are simple problems in only one space dimension. For the compressible flow equations in two dimensions, residuals for cells crossed by shocks are in general poorly computed by the trapezoidal rule applied to the cell faces; and this can trigger violent chequer-board and washboard oscillations which are normally damped by artificial viscosity terms, of both second-order and fourth-order type. The distribution matrices, introduced in (3.13) and (3.14), to match the cell residuals with the unknowns, are also difficult to define. So, although an appeal to feedback control techniques as in Morton and Stringer (1998) offers a way forward with both these difficulties, more development is still needed for the application of cell-vertex methods to these problems.

We turn instead to cell-centre and node-centred methods, where the equations and unknowns have a more natural matching with the number of equations always equal to the number of unknowns. The node-centred method is easiest to apply to the Burgers' equation problem and the mesh described above: there are $J+1$ unknowns corresponding to a piecewise constant representation in which the discontinuities occur at the J cell mid-points; and, for the shock problem which had ingoing characteristics at both boundaries, the end values are given by the boundary conditions, while for $j = 1, 2, \ldots, J-1$ a typical equation will be of the Crank–Nicolson form

$$(x_{j+1/2} - x_{j-1/2})[U_j^{n+1} - U_j^n] + \tfrac{1}{2}(t^{n+1} - t^n)[F_{j+1/2}^{n+1} + F_{j+1/2}^n - F_{j-1/2}^{n+1} - F_{j-1/2}^n] = 0,$$

where the fluxes have to be obtained from an approximate Riemann solver, and $x_{j+1/2}$ is a cell mid-point. For a problem with an outgoing characteristic at a boundary, on the other hand, a corresponding equation is constructed over a boundary half-cell.

A Newton solver needs to be applied to this system of equations, which has some implications for the choice of Riemann solver for the fluxes: they should be smooth functions of the unknowns $\{U_j^{n+1}\}$ which yield a Jacobian matrix to which fast solvers can be applied. For our simple expository problem let us suppose we use simple upwind fluxes. Then, where the flow

is supercritical, the Jacobian will have diagonal elements of the form

$$(x_{j+1/2} - x_{j-1/2}) + \tfrac{1}{2}(t^{n+1} - t^n)U_j^{n+1};$$

and where it is subcritical, of the form

$$(x_{j+1/2} - x_{j-1/2}) - \tfrac{1}{2}(t^{n+1} - t^n)U_j^{n+1}.$$

Hence these will always be increased by the flux terms; and, in general, a reasonable choice for the Riemann solver will lead to a diagonally dominant Jacobian, thus assisting with the choice of a fast solver.

Of course, the disadvantage of this approach is that the immediate outcome is only a first-order accurate approximation: a recovery procedure is needed to obtain higher-order accuracy. However, Fezoui and Stoufflet (1989) successfully used such an approach to approximating the Euler equations on a triangular mesh with simplified Jacobians, reporting almost quadratic convergence with a first-order scheme and quite acceptable results for a second-order scheme. More recently, in a series of papers (Meister and Oevermann 1996, Meister 1998, Meister and Vömel 2001), Meister and his collaborators further developed such methods and applied them successfully to the solution of both the Euler and Navier–Stokes equations. Some of the issues that need to be resolved in such an approach are as follows:

- whether to use the simple Crank–Nicolson (or more general theta-method) form of equation used above, or an implicit Runge–Kutta scheme such as a BDF method (see Hairer and Wanner (1996));
- choice of the recovery procedure, Riemann solver and approximate Jacobian;
- choice of the preconditioner and iteration scheme to solve the linearized equations.

In the papers cited above, Meister used a node-centred scheme based on a Delaunay triangulation of the flow region, and concentrated on steady flow problems. He used discontinuous linear recovery, as described below in Section 5, to obtain second-order accuracy, with numerical fluxes computed by the AUSMDV scheme due to Liou and Steffen (1993) and Wada and Liou (1994) that combines the accuracy of flux-difference splitting methods with the economy of flux-vector splitting schemes. Much of the later sections of this review will be devoted to recovery procedures; and in the next two subsections we summarize some of the other key ideas that are needed to implement such methods.

4.6. ENO and WENO schemes

In the mid-1980s it became clear that the strict imposition of Harten's TVD condition at the recovery stage of an algorithm would lead to a loss of accuracy at solution extrema; and we have already referred in Section 3.2 to the

clipping of peaks when slope limiters are applied in the case of discontinuous linear recovery. New algorithms have therefore been introduced in a very influential series of papers, in which the total variation is allowed to increase to a limited extent in such a way that the order of accuracy is maintained uniformly throughout the domain: these are the *essentially non-oscillatory* or *ENO* schemes.

In the first paper, Harten and Osher (1987), the main ideas are introduced for a second-order approximation called *UNO2* to a scalar one-dimensional problem on a uniform mesh. The recovery stage uses the MUSCL discontinuous linear approximation (3.6) but the novelty lies in the choice of the slopes S_j. A non-oscillatory piecewise parabolic interpolant $Q(x;U)$ of the cell averages is constructed which, between U_j and U_{j+1}, uses either U_{j-1} or U_{j+2} (whichever gives the smaller second difference in absolute value) as the third value to determine the quadratic interpolant: it is shown that this choice results in an interpolant with no more local extrema than the set of U values. Then it is the derivatives of this function that are used to define the slopes as

$$S_j = \text{minmod}(Q(x_j - 0; U), Q(x_j + 0; U)). \tag{4.24}$$

The interface fluxes are calculated by approximating the constancy of the solution along its characteristics: using the same average characteristic speed $a^n_{j+1/2}$ as in the Roe scheme (1.6), when this is positive the flux given for that scheme is changed, with $\lambda = \Delta t/\Delta x$, to

$$F^{n+1/2}_{j+1/2} = f(U^n_j) + \frac{\frac{1}{2}a^n_{j+1/2}(1 - a^n_{j-1/2}\lambda)S^n_j}{1 + \lambda(a^n_{j+1/2} - a^n_{j-1/2})}. \tag{4.25}$$

The result is shown to be a scheme which is uniformly second-order accurate wherever the solution is smooth, including extrema and sonic points.

There are many necessary generalizations of this algorithm. Extensions to a non-uniform mesh are reasonably straightforward, since both the parabolic interpolation and the calculation of the fluxes from the discontinuous linear approximation are readily carried out on an arbitrary mesh. Even extensions to higher-order recovery procedures using Newton divided differences can be formulated in a natural way. The key step there is to define a sequence of cells running out from a given cell, such that at each stage the choice from the left or the right is made so that the resultant divided difference is the smaller in absolute value of the two available. However, Harten, Osher, Engquist and Chakravarthy (1986) showed that, at higher than second order, the process allows spurious oscillations to appear, of a magnitude limited by the order of accuracy – prompting the adoption of the general name ENO. A further point of choice is whether the cell values are regarded as point values, as in UNO2, or the cell averages are matched. The latter

is more in the spirit of the optimal recovery process, and in one dimension is best done by forming the interpolation formula for the primitive function of U and then differentiating the result; but any such procedure requires more computation.

Harten, Engquist, Osher and Chakravarthy (1987) further extended these ideas to systems of equations in one dimension, in particular to the Euler equations. The recovery procedures are applied to locally defined characteristic variables; and the fluxes are calculated by using local Cauchy–Kowalewski formulae, similar to those developed by Ben-Artzi and Falcovitz (1984). In addition, it has long been recognized that *shock recovery* procedures can be applied to finite volume or characteristic-Galerkin methods in order to calculate the position and jump parameters of a shock: see, *e.g.*, Morton and Rudgyard (1988), Morton and Paisley (1989) and Childs and Morton (1990), and references therein. This is taken a step further in Harten (1989), where the ENO recovery procedures are used to detect the presence of contact discontinuities and to recover them so as to prevent the smearing that is normal for most methods. Applying all these developments leads to a very powerful set of one-dimensional schemes: and numerical experiments on standard Euler test problems, such as those from Woodward and Colella (1984), demonstrate impressive results for both second-order and fourth-order methods.

However, the major challenge is the development of such schemes on multidimensional unstructured meshes. We have already described in Section 3.2 some procedures for discontinuous linear recovery on triangular meshes; and much has been done to extend these ideas in connection with the development of ENO schemes: see Harten and Chakravarthy (1991) and Durlofsky *et al.* (1992). But it was Abgrall (1994*b*) who pointed out the advantage of using node-centred rather than cell-centre schemes in this recovery process, and we will take up this topic again in Section 5.

A problem that has generally been encountered with ENO schemes is a lack of convergence to steady flow solutions, because of oscillatory switching that can take place in the choice of recovery stencils. This has led to the development of WENO schemes by Liu, Osher and Chan (1994), in which a smoothness indicator is used to compute a weighted average of all the local recovery polynomials. Extensions to unstructured meshes have been developed by Friedrich (1998), which we will describe in Section 5.

4.7. Multigrid and Krylov subspace methods

In the thirty years in which finite volume methods have been used there have been major developments in methods for solving the large systems of algebraic equations that they generate. Two lines of development have been particularly important: one approaches the problem from the viewpoint of

minimizing an appropriate function, as originated by the conjugate gradient method of Hestenes and Stiefel (1952) and generalized to unsymmetric problems in the GMRES algorithm of Saad and Schultz (1986); and the other exploits the fact that a PDE is being approximated on a mesh and the properties of the equation system depend strongly on the mesh size, as exploited in multigrid algorithms by Brandt (1977), following the earlier work of Federenko (1964) and others.

The two techniques have been applied in tandem to great effect in the solution of incompressible flow problems: see, in particular, Elman, Silvester and Wathen (2005). In such problems, the incompressibility condition brings a degree of ellipticity to the problems that enables the vast experience from Varga (1962) onwards to be built upon. There is a natural progression from discretizing the Poisson equation, and solving the resultant algebraic system, to tackling linear convection-diffusion problems, the Stokes system of equations and thence on to the incompressible Navier–Stokes equations. Algorithms such as GMRES are included in the general class of Krylov subspace algorithms: see the review by Eiermann and Ernst (2001) and the book by van der Vorst (2003). And these methods together with multigrid techniques have been progressively refined, generalized and optimized for this sequence of problems.

Compressible flow problems, with greater nonlinearity and a dominant hyperbolic character, pose greater difficulties. Thus, in early studies, point iteration methods dominated the scene. Multigrid methods were first introduced for steady transonic flow problems by Jameson (1979) and have made a massive impact on the field. The choice of appropriate smoothers is always important in multigrid applications, and the choice of alternating direction smoothers in this early study indicated how the flow direction influenced the behaviour of the methods. For a general reference on multigrid methods in the context of fluid flow problems, see Wesseling (1992); for a description of applications to compressible flow computations on unstructured meshes see Mavriplis (1995), and for a more recent review see Mavriplis (2002).

Krylov subspace methods have also been developed for these problems: see, *e.g.*, Nielsen, Anderson, Walters and Kayes (1995). In particular, their application to the finite volume methods we have described has been explored by Meister (1998). For any such algorithm, efficient preconditioning is essential and this has been developed by Meister and Vömel (2001) for the discretization of hyperbolic conservation laws.

5. Optimal recovery: theory and practice

The two preceding sections will have shown the reader the extent to which the successful development of finite volume methods depends on the recovery or reconstruction stage. Only the cell-vertex approach can give second-order

accuracy without some such step, however rudimentary. We prefer the term 'recovery' because of its reference to the much more general field of *optimal recovery* which we will now outline, and because we feel that it is necessary to take advantage of this greater generality. Without such a general framework there is a natural tendency to use only local polynomials, as we have seen with the ENO methods, and these can easily lead to ill-conditioned procedures, with breakdown occurring there in the one-dimensional case when the polynomial degree exceeds six. The initial ideas for this theory are due to Golomb and Weinberger (1959) and were subsequently greatly developed in Micchelli and Rivlin (1977).

5.1. Theory of optimal recovery

Suppose we are trying to deduce some property of an unknown function u from some data given for it, and we can set up the problem in the following way. We suppose that u lies in a Hilbert space H with a bound on its norm $\|u\|_H$, and the data are given in the form of the values of a set of bounded linear functionals on that space, the *information operator*

$$\mathcal{I} := \{F_1(\cdot), F_2(\cdot), \ldots, F_M(\cdot)\};$$

furthermore, what we seek is the value of another bounded linear functional $F(\cdot)$, the *feature operator*. Then Golomb and Weinberger (1959) observed that the given values of the linear functionals define a hyperplane in H and the bound on $\|u\|_H$ a hypersphere, with their intersection defining a *hypercircle* (see Synge (1957)); moreover, the centre of this hypercircle defines a function $u_c \in H$ and it gives a value of the sought-after functional $F(u_c)$ which is optimal, and this is the case independently of the information functional that is sought. We will illustrate why this is so by means of one of the most important examples of the theory.

Let $a(\cdot,\cdot)$ be a symmetric, bilinear form on $H \times H$ that defines an elliptic boundary value problem on a region Ω, with homogeneous Dirichlet boundary conditions: find $u \in H$ such that

$$a(u, v) = (f, v), \quad \forall v \in H;$$

and let the norm on H be defined by $\|v\|_H^2 := a(v, v)$. Now consider the approximation of this problem by a finite element method which uses the basis functions $\phi_1, \phi_2, \ldots, \phi_M$ lying in H, and suppose we regard the data functionals to be defined by

$$F_i(u) := (f, \phi_i) = a(u, \phi_i), \quad i = 1, 2, \ldots, M.$$

Indeed, whatever the given data functionals, by the Riesz representation theorem we could find their *representers* ϕ_i which would be defined by these equations; and we could continue the following construction. We define $H_M := \text{span}\{\phi_i,\, i = 1, 2, \ldots, M\}$; and then the centre of the hypercircle is

the orthogonal projection, with respect to $a(\cdot,\cdot)$, of u onto H_M. That is, it is the familiar finite element approximation $U_M \in H_M$ to u using this set of basis functions:

$$F_i(U_M) \equiv a(U_M, \phi_i) = (f, \phi_i) \equiv F_i(u), \quad i = 1, 2, \ldots, M.$$

Then we suppose that the linear functional whose value for u we seek to estimate is

$$F(\cdot) \equiv a(\cdot, \psi),$$

where ψ is its representer.

Suppose now that ψ_M is the orthogonal projection of ψ onto H_M. It follows that

$$\begin{aligned} |F(u) - F(U_M)| &= |a(u - U_M, \psi)| = |a(u - U_M, \psi - \psi_M)| \\ &\leq \|u - U_M\|_H \|\psi - \psi_M\|_H. \end{aligned} \tag{5.1}$$

Moreover, let us denote by $\Delta_M u$ and $\Delta_M \psi$ the two (positive real) factors on the right and consider two alternative choices for estimating F, namely

$$u_\pm := U_M \pm \frac{\Delta_M u}{\Delta_M \psi}(\psi - \psi_M),$$

for which we can readily check that $F_i(u_\pm) = F_i(u)$, $\forall i$. Then it is easily seen that

$$F(U_M) - F(u_\pm) = \pm(\Delta_M u)(\Delta_M \psi); \tag{5.2}$$

that is, $F(U_M)$ lies at the centre of the interval $[F(u_-), F(u_+)]$ of possible values for $F(u)$, hence giving the optimal estimate for this quantity.

It is a key property of Galerkin and Petrov–Galerkin finite element approximations that they are optimal or near-optimal approximations in an energy norm. Thus in Barrett, Moore and Morton (1988*a*) the framework outlined above was used to derive techniques for recovering point values of functions from their weighted L^2 best fits, using both local and global recovery procedures; while in Barrett, Moore and Morton (1988*b*) global recovery techniques were developed from low-order finite element approximations to ODE problems to obtain higher-order approximations, techniques which thus correspond to defect correction methods. These introduce higher-order approximations in a very similar way to those needed for the finite volume methods we will discuss below.

The theory of optimal recovery has been developed in a very general setting and the ideas applied to a wide range of problems: see Micchelli and Rivlin (1977). For finite volume methods the linear functionals that provide the data are the cell averages; and the information that is required consists of the function and derivative values from which higher-order approximations can be constructed – and, in particular, from which fluxes on the element boundaries can be computed. In the next two subsections we will describe

how the ENO recovery techniques may be generalized in a very natural way to two-dimensional triangular meshes. Then we will describe some less conventional approaches to this problem, based on spline functions and radial basis functions.

5.2. Recovery on primary triangular grids

In generalizing the ENO recovery techniques to obtain higher-order approximations we will need to consider how to select the mesh for successively higher orders, what form of expansion to use for the approximations, how to solve the algebraic equations for the expansion coefficients and how each of these interact. For instance, it is well known that a linear approximation in two dimensions cannot be derived from three point values given on a straight line. So for the cell-centre scheme we would not want to use three triangles whose centroids almost lie on a line in order to generate our linear recovery approximation. In what follows we shall generally refer to the recovery of the scalar quantity U: in application to such as the Euler equations, this will denote one component of the vector of unknowns or, more commonly, chosen to be one of the primitive variables.

A very simple but surprisingly successful technique for linear recovery was described by Durlofsky *et al.* (1992). The possible node sets for recovery on the triangle T_i in Figure 5.1 consist of the barycentre of T_i and those of two of its neighbours:

$$\begin{aligned} K_1(T_i) &:= \{T_j, j \in \{i, i_2, i_3\}\}, \\ K_2(T_i) &:= \{T_j, j \in \{i, i_3, i_1\}\}, \\ K_3(T_i) &:= \{T_j, j \in \{i, i_1, i_2\}\}. \end{aligned}$$

On each node set $K_k(T_i)$ a linear polynomial

$$\pi_i^{(k)} := a_{00}^{(k)} + a_{10}^{(k)}(x_1 - c_{i,1}) + a_{01}^{(k)}(x_2 - c_{i,2}), \quad k = 1, 2, 3,$$

where $\mathbf{c}_i$ is the barycentre of T_i, is computed by solving the linear systems

$$\mathcal{A}(T_j)\pi_i^{(k)} = U_j, \quad \forall (k, j); \tag{5.3}$$

here the range of values for (k, j) are given in the above definition of the node sets. If there is an extremum at triangle T_i with respect to its three neighbours, then none of the polynomials is chosen and the value on T_i is not recovered. Otherwise we consider the steepest of the three linear polynomials and check whether its use would result in a new extremum. If that is not the case this polynomial is taken to be the recovery on T_i. If a new extremum is created the polynomial with the next steepest gradient is considered, and so on. If all three polynomials would result in a new extremum then no recovery is used on T_i. Note that this procedure corresponds more to the classical TVD approach than to an ENO approach.

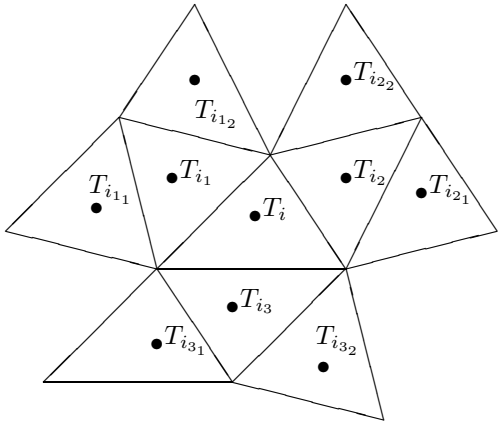


Figure 5.1. Neighbourhood of T_i.

An even simpler algorithm consists of choosing the node set that yields the linear polynomial with the gradient of smallest absolute value. This is a true generalization of the ENO idea in that small oscillations are then allowed to occur. However, this can lead to quantities such as density or pressure taking on negative values; so a remedial step would be needed. To reduce this possibility one might consider a larger number of node sets: for example, using the neighbours of the neighbours of T_i, that is, all the triangles shown in Figure 5.1. But such a large stencil is more appropriate for quadratic recovery.

To carry out quadratic recovery, for each node set we need to compute the coefficients in an expansion of the form

$$\pi_i(x) := a_{00} + a_{10}(x_1 - c_{i,1}) + a_{01}(x_2 - c_{i,2}) + a_{11}(x_1 - c_{i,1})(x_2 - c_{i,2}) + \tfrac{1}{2}a_{20}(x_1 - c_{i,1})^2 + \tfrac{1}{2}a_{02}(x_2 - c_{i,2})^2$$

by matching its average over each triangle in the node set with that of U. Although the integrals occurring in the coefficient matrix of this system can be computed exactly it makes more sense to use an appropriate quadrature rule.

The real problem is the selection of the node sets. In Harten and Chakravarthy (1991) a sectorial search strategy is advocated in which the region outside the central triangle T_i is divided into sectors by continuing the triangle sides in both directions, giving three based on the triangle sides alternating with three based on its vertices; then these are treated in a way that corresponds to the two directions in the one-dimensional case, and typically will give 18 possible ways of adding the three triangles needed for the quadratic node set. Alternatively, as in Abgrall (1994*a*) we can argue

as follows: referring to Figure 5.1 suppose that the chosen linear node set consisted of $\{T_i, T_{i_1}, T_{i_2}\}$; then there are three pairs of edges on the outer perimeter that correspond to a pair of triangles, such as $\{T_i, T_{i_1}\}$, and on each edge we consider the neighbouring triangle and its two neighbours, such as T_{i_3} with $T_{i_{3_1}}$ and $T_{i_{3_2}}$ on the outer edge of T_1 as shown in the figure; this will give us six choices of three triangles: T_{i_3} and its two neighbours, $T_{i_{1_1}}$ and its two neighbours, and both T_{i_3} and $T_{i_{1_1}}$ with one of their four neighbours. Again we have 18 choices!

Once a set of possible quadratic polynomials is computed, one has to be selected by some set of criteria. To ensure that the choice is the least oscillatory in some sense, we could use the criterion

$$W(\pi) := \sqrt{\sum_{\mu=1}^{2} \sum_{|\alpha|=\mu} a_\alpha^2} \tag{5.4}$$

and choose the polynomial which gives a minimal value of W. However, there are many other possible criteria; and it is far from clear that the Taylor series used above is best suited for defining these criteria, or for computing the quadratic and higher-order approximations. We will take up these points in more detail in the next subsection.

5.3. Recovery on secondary grids

It is argued by Abgrall (1994*b*) that there are fewer node sets to consider for higher-order recovery algorithms in this case, and this will lead to important advantages of node-centred schemes over their cell-centre counterparts: we will consider them by reference to Figure 5.2, where the nodes are now the vertices of the triangular grid. In the first stage, for linear recovery, we consider all the triangles which share a given vertex, say i_0, and construct a linear function for each such that its average over each box centred on one of its vertices matches the corresponding average of U. Then we choose that with the smallest gradient: in the figure this is labelled $T_{\min}$. We will describe the quadratic recovery stage in more detail before we come back to this linear stage.

There are several very important developments presented in Abgrall (1994*b*) which elaborate on the advantages of the node-centred methods. The first has to do with the selection of successive node sets: as can be seen from Figure 5.2, the three further vertices needed for quadratic recovery can be obtained from the triangle (and its two further neighbours) that shares one of its sides with $T_{\min}$: this gives a choice of three node sets $K(B_{i_0})$ for this stage. Indeed, it is claimed in the paper that at each further stage only three possible node sets need to be considered.

An even more important aspect of defining a true generalization of the ENO process is to select a representation of the approximation at each

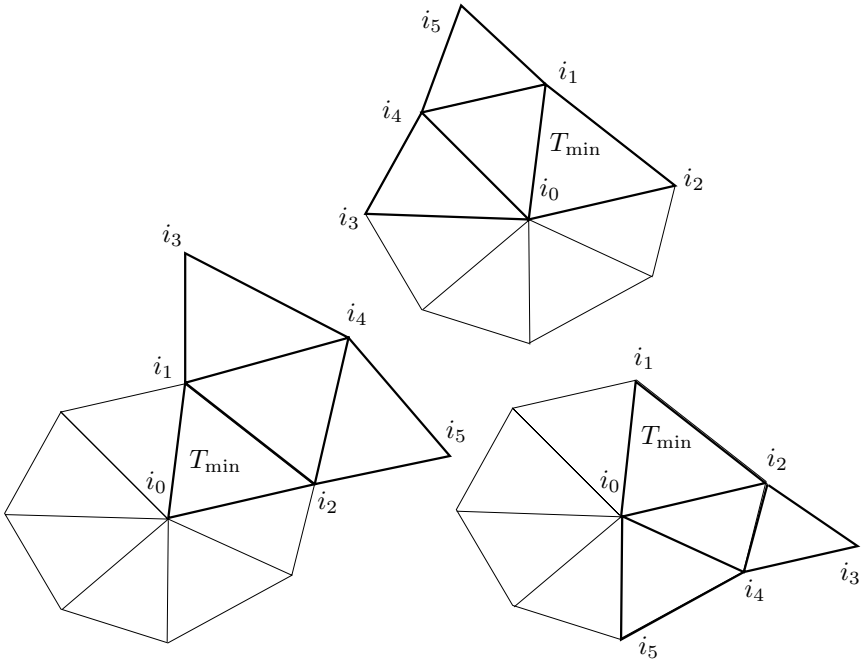


Figure 5.2. Three possible sets $K(B_{i_0})$ for quadratic recovery.

stage which has some of the important properties of the Newton divided differences used in the one-dimensional schemes. In Abgrall (1994*b*) the barycentric coordinates of $T_{\min}$ are used for the quadratic expansion; that is, with a cyclic ordering of $(1, 2, 3)$, and now with $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ the vertices of $T_{\min}$, we have the coordinates

$$\lambda_1(\mathbf{x}) = \frac{1}{2|T_{\min}|}\big[(x_{3,1} - x_{2,1})(x_2 - x_{2,1}) - (x_1 - x_{2,1})(x_{3,2} - x_{2,2})\big].$$

Then a quadratic polynomial for the box centred at i_0 can be written as

$$\pi_{i_0}(\mathbf{x}) = \sum_{m=1}^{3}\left(a_m\lambda_m(\mathbf{x}) + \sum_{n>m} A_{mn}\lambda_m(\mathbf{x})\lambda_n(\mathbf{x})\right);$$

and Abgrall could prove that the coefficient matrix of the system

$$\mathcal{A}(B_j)\pi_{i_0} = U_j, \qquad B_j \in K(B_{i_0}) \tag{5.5}$$

has condition number of order 1. This also holds for higher-order recovery, which is in sharp contrast to the poor conditioning obtained with the Taylor series expansion. In addition, by following the analysis of Ciarlet and Raviart (1972) for interpolation by finite element approximations, he was able to show that the derivatives of smooth functions were approximated

similarly, with a similar dependence on the mesh quality. Thus the recovered functions should provide reliable indicators of the smoothness of the unknown solution.

For the quadratic recovery Abgrall also showed that the system can be factored into two subsystems of size 3×3, with one of them corresponding to the system needed for the linear recovery stage. So this captures another key feature of the one-dimensional algorithm. Subsequently, in Abgrall and Sonar (1997) it is shown that this property also holds at all orders by exploiting the generalization of Newton divided differences developed by Mühlbach (1978). We will not give any details here, but it is useful to outline the main ideas.

Mühlbach introduced the idea of a *complete Chebyshev system* of functions $(f_1, f_2, \ldots, f_k, \ldots, f_n)$, which for simplicity we can take to be real functions on $\mathbb{R}$, for which

$$V\begin{pmatrix} f_1, \ldots, f_k \\ x_1, \ldots, x_k \end{pmatrix} := \det f_j(x_i) \neq 0$$

is true for any distinct set of points $(x_1, \ldots, x_n)$ and $k = 2, 3, \ldots, n$. For such a system it is clear that a linear interpolatory formula can be constructed for another function $f(\cdot)$ which, with its error, he denotes as follows:

$$p_n f \equiv pf\begin{bmatrix} f_1, \ldots, f_n \\ x_1, \ldots, x_n \end{bmatrix}, \quad r_n f := f - p_n f.$$

Moreover, this can be expressed in a series whose terms are *generalized divided differences* of the function f, with that of order k given by

$$\begin{bmatrix} f_1, \ldots, f_k, \\ x_1, \ldots, x_k \end{bmatrix} f := \frac{V\begin{pmatrix} f_1, \ldots, f_{k-1}, f \\ x_1, \ldots, x_{k-1}, x_k \end{pmatrix}}{V\begin{pmatrix} f_1, \ldots, f_{k-1}, f_k \\ x_1, \ldots, x_{k-1}, x_k \end{pmatrix}};$$

the series can then be written in the form

$$p_n f \equiv pf\begin{bmatrix} f_1, \ldots, f_n \\ x_1, \ldots, x_n \end{bmatrix} = \sum_{k=1}^{n} \begin{bmatrix} f_1, \ldots, f_k, \\ x_1, \ldots, x_k \end{bmatrix} f\, g_k, \tag{5.6}$$

where

$$g_1 := f_1, \quad g_k := r_{k-1} f_k, \quad \text{for} \quad k = 2, \ldots, n.$$

Finally, a recurrence relation for the divided differences

$$\begin{bmatrix} f_1, \ldots, f_k, \\ x_1, \ldots, x_k \end{bmatrix} f = \frac{\begin{bmatrix} f_1, \ldots, f_{k-1}, \\ x_2, \ldots, x_k \end{bmatrix} f - \begin{bmatrix} f_1, \ldots, f_{k-1}, \\ x_1, \ldots, x_{k-1} \end{bmatrix} f}{\begin{bmatrix} f_1, \ldots, f_{k-1}, \\ x_2, \ldots, x_k \end{bmatrix} f_k - \begin{bmatrix} f_1, \ldots, f_{k-1}, \\ x_1, \ldots, x_{k-1} \end{bmatrix} f_k}$$

was shown by Mühlbach to follow from a general Neville–Aitken recurrence formula.

What was shown in Abgrall and Sonar (1997) was that all of this could be generalized to the recovery problem in more than one space dimension and using linear functionals on the solution space, with the given functionals corresponding to the given function values and the unknown function to the sought-after linear functional. Then the Vandemonde determinants that occur in the definition of the generalized divided differences correspond to recovery equations such as (5.5) that have to be solved; and the recurrence relation for these divided differences expresses the fact that at each stage the system can be solved by solving similar systems corresponding to earlier stages. In particular, quadratic recovery can be implemented by twice solving the sort of 3×3 system needed for linear recovery.

Thus ENO schemes using quadratic and higher-order recovery become a very practical proposition. Moreover, so do the WENO schemes which require the calculation of more recovery approximations. On each box B_i we have to compute a set of recovery polynomials $\pi_i^{(k)}$ where k denotes the number of the stencil; then we compute a weighted sum

$$\pi_i := \sum_k \Omega_k \pi_i^{(k)}$$

where the Ω_k are weights with $\sum_k \Omega_k = 1$. An *oscillation indicator* OI is used to compute the weights: for example, we may use a Sobolev seminorm

$$OI(\pi_i^{(k)}) := \|\nabla \pi_i^{(k)}\|_{L^2(B_i)}$$

as an oscillation indicator; then the weights are computed from

$$\Omega_k := \frac{\omega(k)(\varepsilon + OI(\pi_i^{(k)}))^{-\beta}}{\sum_j \omega(j)(\varepsilon + OI(\pi_i^{(j)}))^{-\beta}}.$$

Here, $\omega(j), \omega(k)$ are weights which allow a different weighting of different stencils. The parameter ε is chosen to avoid the division by zero and β is a measure of sensitivity of the weights on the oscillation indicator. We set $\varepsilon = 10^{-16}$, $\beta := 8$ and $\omega(k) = 12$ for a central stencil while $\omega(j) = 1$ for a one-sided stencil. Such WENO schemes were developed by Friedrich (1998) and an example of their effectiveness is shown in Sonar (2002).

5.4. Splines and radial basis functions

In one dimension, splines are commonly introduced through a variational principle: the linear spline interpolant of a function at a given set of knots is that interpolant that minimizes the L^2-norm of its derivative: and a cubic spline interpolant is similarly an interpolant that minimizes the norm of

the second derivative: see, for instance, de Boor (2001). More generally they can be characterized as centres of hypercircles in certain semi-Hilbert spaces, that is, a space with seminorm $|\cdot|_V$ for which there may exist non-trivial functions $w \in V$ for which $|w|_V = 0$ holds (*i.e.*, the seminorm has a 'hole': its kernel $\ker|\cdot|_V$ contains more than the null function). A spline in a semi-Hilbert space is then defined to be a function $\Phi \in V$ which minimizes the seminorm: that is, for the given information operator $\mathcal{I}$,

$$|\Phi|_V = \inf_{\substack{v\in V \\ \mathcal{I}v=\mathcal{I}u}} |v|_V$$

is to be satisfied.

Thus a one-dimensional cubic spline interpolant minimizes the seminorm given by the L^2-norm of the second derivative, which has a kernel consisting of linear polynomials, and these have to be specified by some side conditions; for example, the natural cubic spline is determined by setting to zero the second derivative at each boundary. In two dimensions the cubic spline generalizes to the *thin plate spline*, so-called because it is the solution of the biharmonic equation. In our setting of recovery from cell average data, Sonar (1996) has shown that it is given by

$$\Phi(\mathbf{x}) = \sum_{j=0}^{M-1} \alpha_j \mathcal{A}^{(\mathbf{y})}(\sigma_{i_j})\big[|\mathbf{x}-\mathbf{y}|^2 \ln(|\mathbf{x}-\mathbf{y}|)\big] + a_{01}x_1 + a_{10}x_2 + a_{00}, \quad (5.7)$$

where $\mathcal{A}^{(\mathbf{y})}$ denotes application of the cell average operator with respect to the variable $\mathbf{y}$, and the additional linear polynomial is the contribution from the kernel of the seminorm. We now have to determine $M+3$ coefficients $\alpha_0, \ldots, \alpha_{M-1}$, a_{10}, a_{01}, a_{00} but we have only M conditions given by the information $\mathcal{I}u = \{\overline{u}\}$, the cell averages on the node set. If we require the condition

$$\forall q \in \ker|\cdot|_V : \quad \sum_{j=0}^{M-1} \alpha_j \mathcal{A}(\sigma_{i_j}) q = 0$$

we get the remaining three conditions needed to determine all coefficients: we can think of this condition as 'fixing the hole' in the seminorm.

One can easily prove that the thin plate spline reproduces linear polynomials, so that recovering from three given cell averages just gives the linear polynomial which is constructed to fix the hole in the seminorm. Hence we need to have more than three cells in a node set. In applying this recovery to the cell T_i in Figure 5.1 we therefore use node sets comprised of four neighbouring triangles: there is one central node set $K_0(T_i) := T_i \cup T_{i_1} \cup T_{i_2} \cup T_{i_3}$; and the three one-sided node sets $K_1(T_i) := T_i \cup T_{i_1} \cup T_{i_{1_1}} \cup T_{i_{1_2}}$, $K_2(T_i) := T_i \cup T_{i_2} \cup T_{i_{2_1}} \cup T_{i_{2_2}}$, and $K_3(T_i) := T_i \cup T_{i_3} \cup T_{i_{3_1}} \cup T_{i_{3_2}}$. On each of the node sets we solve the

linear system

$$\mathcal{A}(T)\Phi = \mathcal{A}(T)u =: \overline{u}_j, \quad T \in K_j(T_i), \quad j = 0, 1, 2, 3, \tag{5.8}$$

together with the conditions

$$\sum_{j=0}^{3} \alpha_j \mathcal{A}(T) = \sum_{j=0}^{3} \alpha_j \mathcal{A}(T)x_1 = \sum_{j=0}^{3} \alpha_j \mathcal{A}(T)x_2 = 0. \tag{5.9}$$

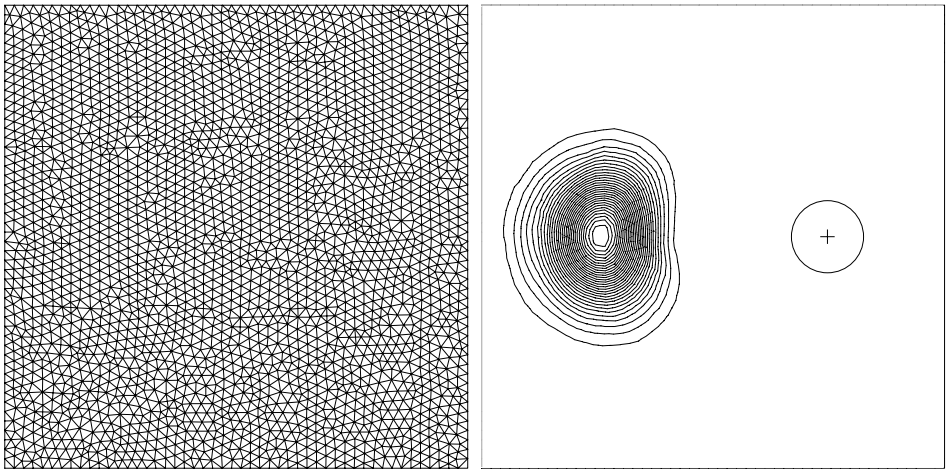
Denoting respectively by M and N the matrices in these two systems, we can write the equations for the seven unknowns in this thin plate recovery spline as

$$\begin{bmatrix} M & N^T \\ N & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_3 \\ a_{10} \\ a_{01} \\ a_{00} \end{bmatrix} = \begin{bmatrix} \overline{u}_0 \\ \vdots \\ \overline{u}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.10}$$

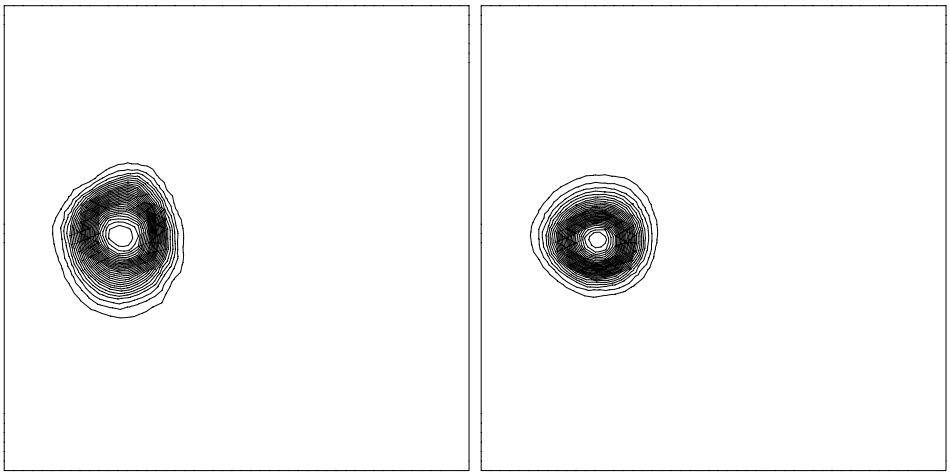
Explicit expressions for the matrices M and N are relatively easy to calculate. Then, after computing the four recovery splines on the four node sets, that with smallest total variation over T_i is chosen to be the recovery spline on T_i.

It is shown in Iske and Sonar (1996) that the thin plate spline is just one example of a *radial basis function* that may be used in a cell average recovery algorithm. Conditions which have to be satisfied by any such function of the form $\psi(|\mathbf{x} - \mathbf{y}|)$ are given, as well as several other examples. They all require considerably more computation than the more conventional polynomial recovery. The thin plate spline has been experimented with most widely, but alternatives are easier to use and equally effective.

A simple model problem that can be used to compare some of the recovery procedures that we have discussed is the following: the initial data consists of a straight-sided cone, of unit height and with the radius of its base 0.15, whose centre is at (0.5,0); it is then convected in a circle about the origin, and we plot the results of various computations at a time corresponding to half a revolution. In Figure 5.3(a), the left plot shows the mesh (of 2500 nodes and 4802 triangles), and the right plot shows the initial data and contours of the solution without recovery after half a revolution, using the Engquist–Osher flux and a simple explicit time-stepping. Figure 5.3(b) shows the corresponding result obtained with the linear recovery procedure due to Durlofsky *et al.* (1992); and (c) is that obtained with a thin plate spline recovery step by Sonar (1996). The figure shows that the thin plate spline reproduces the cone much more accurately than the alternatives, both



(a) Computational grid (*left*), base and centre of initial cone and solution without recovery



(b) Solution with linear recovery (c) Solution with thin plate spline recovery

Figure 5.3. The rotating cone problem.

Table 5.1.

$\overline{u}$	basic	DEO	mingrad	quad1	quad2	TPS
min	0.0	0.0	-1.4×10^{-5}	-2.0×10^{-6}	0.0	0.0
max	0.382	0.635	0.753	1.04	0.764	0.974

as regards its compactness and its circular shape. Table 5.1, which shows minimum and maximum cell average heights, emphasizes the comparisons.

In the headings, *basic* denotes the basic unrecovered scheme, *DEO* linear recovery using the Durlofsky *et al.* (1992) algorithm, while *mingrad* denotes the simpler algorithm which chooses the linear recovery with smallest gradient; similarly *quad1* denotes quadratic recovery using the simple criterion (5.4), *quad2* is a more sophisticated quadratic recovery using a sector search algorithm and, finally, *TPS* denotes the thin plate spline recovery just described.

6. Grid adaptivity: *a posteriori* error control

Except in rather simple and special cases it is impractical to use any form of shock fitting to achieve sharp definition of shocks. The practical alternative is to use local mesh refinement. This is simplest with a triangular or tetrahedral mesh, and a large literature on both the practical and theoretical techniques has developed for application to elliptic problems: for a general introduction which leads towards our present CFD problems, see Eriksson, Estep, Hansbo and Johnson (1995) and the references therein. In order to build on this for our finite volume methods, it is best to use node-centred methods: then the cell averages over the boxes centred on each vertex are recovered to give local polynomial approximations on each box, as described in the previous section; restricting these to the vertices of the primary triangular mesh gives a continuous piecewise linear approximation on which to base criteria for mesh refinement or coarsening. This will form the basis of the methods described below.

There are, however, many differences in what is required for a compressible flow calculation from that for the approximation of a scalar elliptic problem. Estimating the *a posteriori* error is probably the major difference: which component or combination of components should be used to measure the error; what norm should be used; how to distinguish the measured error from its source; and then how best to do all of this when the solution may be changing rapidly with time? We shall pay less attention to this last aspect: we will consider problems of mesh refining and coarsening

arising from shock movement; but we will not consider the estimation and use of variable time steps.

6.1. Mesh refinement and recoarsening

We will not give here all the details of any particular procedures, but it is important to outline the key ideas in order to understand the properties of the resultant approximations. At the end of each refinement or recoarsening we will ensure that we have a conforming triangulation; and we start with a conforming triangulation that is everywhere the coarsest that will be used. The algorithms summarized below follow the general strategies of Bank, Sherman and Weiser (1983); details can be found in Sonar (2002).

There are two main types of refinement: the so-called *red-refinement* of a triangle in which the mid-points of the sides are joined so that the triangle is divided into four similar triangles as in Figure 6.1; this will make the neighbouring triangles non-conforming, so that a *green-refinement* would be needed in which a mid-side is joined to the opposite node so as to divide the triangle into two as in the figure. The triangles resulting from a red-refinement are called *red triangles*, and are termed the *daughters* of the original *mother* triangle: similar terminology is used for the green-refinement.

These basic refinements of a single triangle can be used to define a refinement procedure for a conforming triangulation $\mathcal{T}$ in which a subset of its triangles have been marked for refinement:

`Algorithm 1`

1 Eliminate all green-refinements in $\mathcal{T}$ by restoring the mothers of green triangles. If a green triangle was specified for refinement the restored mother is also marked for refinement.
2 Red-refine all triangles which are marked for refinement.
3 While there exist triangles in $\mathcal{T}$ with more than one non-conforming node, they are red-refined.
4 Apply the green-refinement for all triangles which have exactly one non-conforming node.

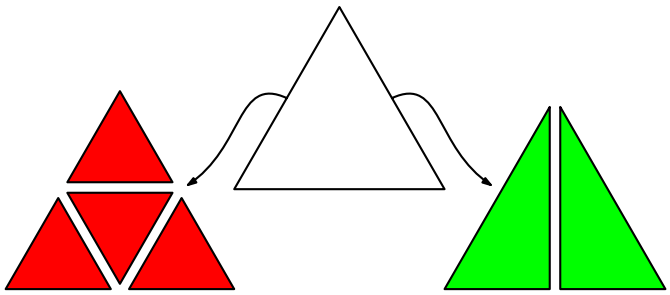


Figure 6.1. Red-refinement and green-refinement of a triangle.

The algorithm terminates with a conforming triangulation. Since all children of red-refinements are similar to their mothers and green triangles will be removed in the next refinement step, the refinement procedure is *stable*: that is, the inner angles of the triangles are bounded from below in any sequence of grid refinements.

In order to carry out a recoarsening procedure it is necessary to carry with each triangle a compact data structure called `History`. This will include whether the triangle is the result of a green-refinement, and if so the identifier of its sister; also it must hold the number of red-refinements that led to this triangle together with a data stack specifying its sisters and its antecedents. Then the following algorithm can be applied to a *resolvable patch*:

`Algorithm 2`

for all triangles $T \in \mathcal{T}$ which are specified for recoarsening
  for all three vertices P of T
    if the pair (P, T) spans a *resolvable patch* $\mathcal{P}$
      recoarsen this resolvable patch $\mathcal{P}$.

Figure 6.2 illustrates on the left-hand side some configurations for resolvable patches around a vertex, and on the right-hand side possible recoarsenings of the patch without producing hanging nodes. The bottom part of the figure illustrates a situation at the boundary of a triangulation.

The actual recoarsening of the triangles in a resolvable patch $\mathcal{P}$, spanned by triangle $T \in \mathcal{T}$ and one of its vertices P, is carried out as follows:

`Algorithm 3`

1 Remove all triangles in $\mathcal{P}$ and restore their mothers.
2 Remove all green triangles which produce hanging nodes in the mothers of $\mathcal{P}$, and restore the mothers of these green triangles.
3 Green-refine all triangles which have one non-conforming node.

The result of such recoarsening is a conforming triangulation that would be obtainable from the original triangulation by a sequence of refinement steps. Indeed, a sequence of such recoarsenings could lead back to the original triangulation.

6.2. Weighted L^2-norm error control

It was Johnson and his collaborators – see, *e.g.*, Hansbo and Johnson (1991) and Eriksson and Johnson (1993) – who introduced the idea of residual-based error indicators to CFD from their well-developed use with elliptic equations. We denote by $\mathbf{u}^h$ the continuous piecewise linear approximation constructed from a finite volume computation, and if $\mathcal{L}$ is a first-order differential operator, the corresponding *residual* for the problem $\mathcal{L}\mathbf{u} = \mathbf{0}$ can

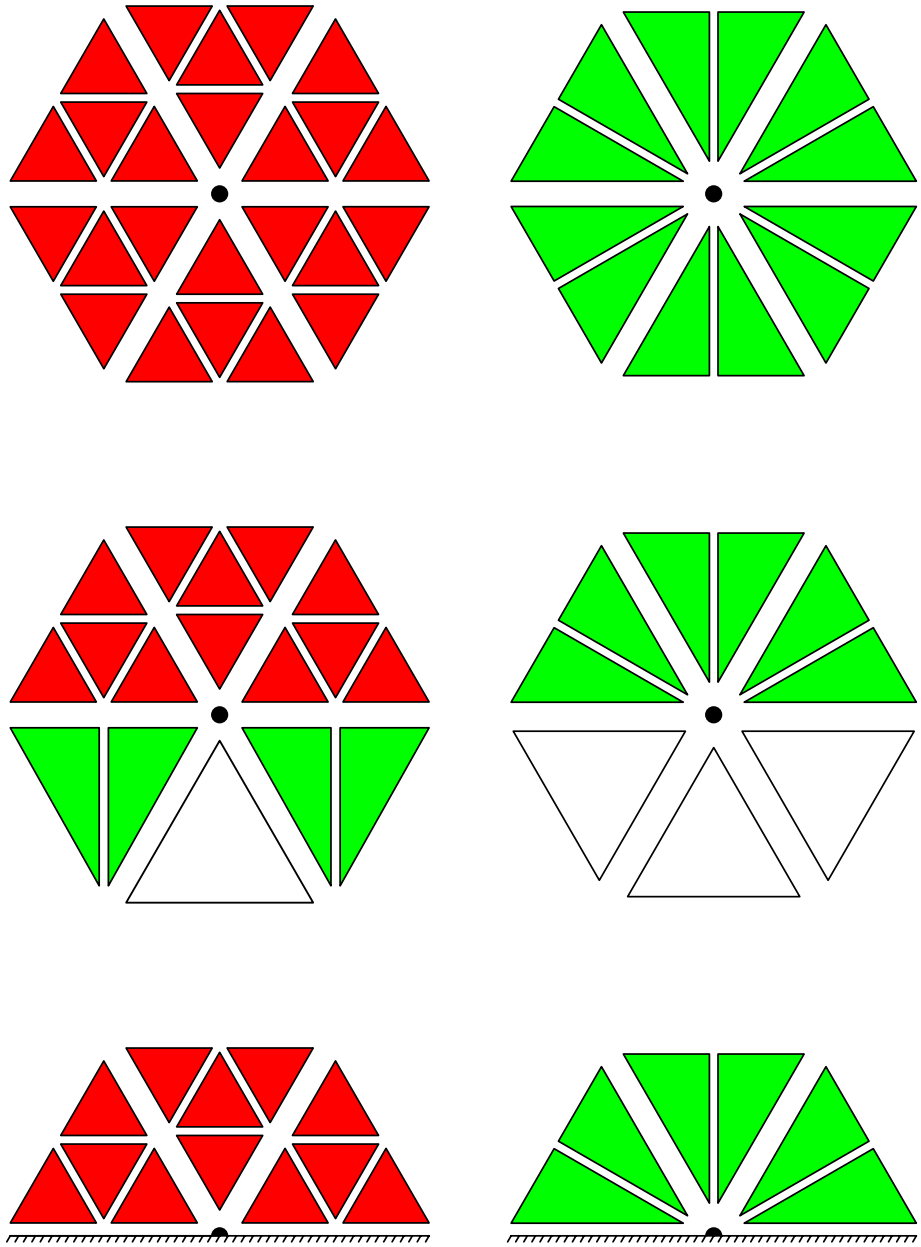


Figure 6.2. Resolvable patches (*left*) and recoarsened triangles (*right*).

be calculated as

$$\mathbf{r}^h := \mathcal{L}\mathbf{u}^h.$$

Our objective is local error control based on such a residual. In application to the Euler equations we have four components, and might first consider using the sum of the L^2-norms of each component over each triangle T, *i.e.*,

$$\|\mathbf{r}^h\|_{L^2(T)} := \sum_{i=1}^{4} \|r_i^h\|_{L^2(T)}$$

defined on the triangles of the primary grid, with the components of the residual corresponding to the continuity equation, the two momenta and energy equations.

Although our initial aim is to use this residual to guide the selection of triangles for refinement or recoarsening, the more ambitious target (which is achievable for the finite element approximation of elliptic problems) would be to define a residual which provides an efficient and reliable bound on the actual error in the approximation, $\mathbf{e}^h := \mathbf{u}^h - \mathbf{u}$: if the operator $\mathcal{L}$ has a bounded inverse from some space Y to a space X, we would like to establish bounds of the form

$$C_1\|\mathbf{r}^h\|_Y \le \|\mathbf{e}^h\|_X \le C_2\|\mathbf{r}^h\|_Y \tag{6.1}$$

for some computable constants C_1, C_2. Although we cannot expect to achieve this for the nonlinear Euler equations, it should be borne in mind as an eventual aim and thus give some guidance on how to measure and weight the contribution from each triangle. Moreover, we will below get quite close to realizing this aim for closely related PDE systems.

As a first step, let us consider using the unweighted L^2-norm of a shocked flow: in particular, suppose that the continuous piecewise linear numerical approximation, on a uniform mesh of size h, given by

$$u^h(x) = \begin{cases} 0 & 0 \le x < x_i, \\ (x - x_i)/h & x_i \le x < x_{i+1}, \\ 1 & x_{i+1} \le x \le 1, \end{cases}$$

approximates the function u which jumps from 0 to 1 at the mid-point of the interval $[x_i, x_{i+1}]$; and suppose also that the first-order differential operator $\mathcal{L}$ is just ∂_x. Then the L^2-norm of the residual on the interval $[x_i, x_{i+1}]$ is easily calculated to be

$$\|r^h\|_{L^2([x_i,x_{i+1}])} = \sqrt{\int_{x_i}^{x_{i+1}} |r^h|^2 \, \mathrm{d}x} = \sqrt{\int_{x_i}^{x_{i+1}} \frac{1}{h^2} \, \mathrm{d}x} = \frac{1}{\sqrt{h}},$$

and this quantity blows up at discontinuities as the grid is refined. Although in two dimensions on a square mesh this would be avoided for the norm on

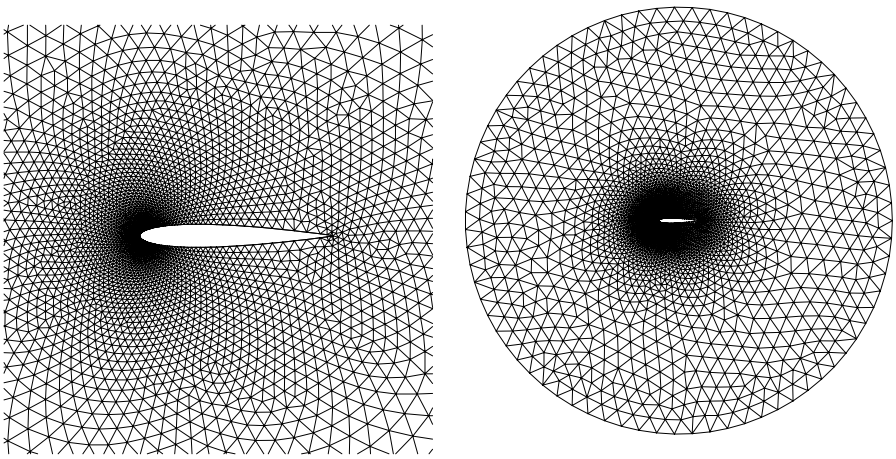


Figure 6.3. Initial grid for the NACA0012 aerofoil.

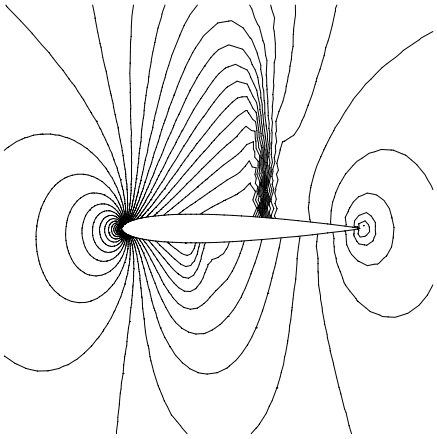


Figure 6.4. Pressure distribution on the initial grid.

an individual mesh square or triangle, for a shock extending a finite distance the norm over the region covering it would blow up in the same way. So the unweighted norm would give excessive refinement near such a shock.

On the other hand, in the finite element methods of Hansbo and Johnson (1991) it was found that the local triangle diameter should be used as a weight factor:

$$\|\mathbf{r}^h\|_{L^2_h(T)} := h_T \|\mathbf{r}^h\|_{L^2(T)}, \tag{6.2}$$

where h_T denotes the length of the longest side of T. In Sonar (2002) its use as a refinement indicator for finite volume methods was compared with the use of other powers of h_T, and the use of more heuristic alternatives

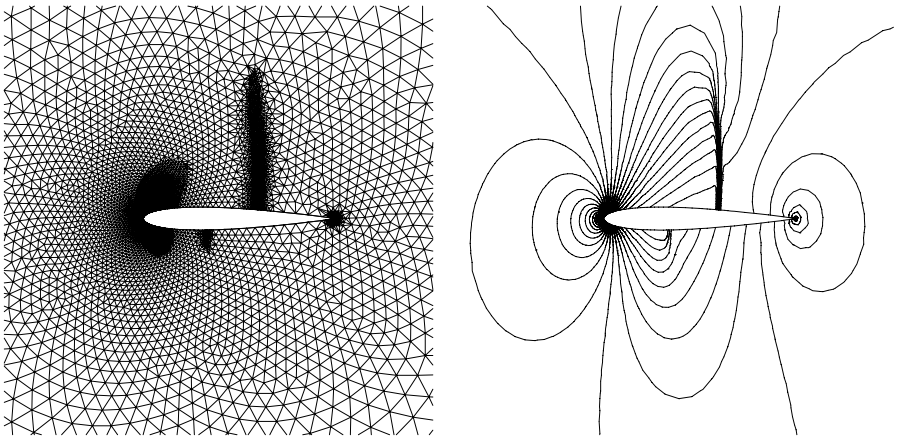


Figure 6.5. Grid after three refinement cycles with the unweighted norm indicator (*left*). Pressure distribution on this grid (*right*).

which had been advocated by other authors, for a number of flow problems. The simplest test problem was the very standard problem of the flow about the NACA0012 aerofoil, in which the Mach number of the incoming flow is $\mathrm{Ma} = 0.8$ and the angle of attack is $\alpha = 1.25°$. The flow should contain a strong shock on the upper side of the aerofoil and a weak one on the lower side. The initial grid in the vicinity of the profile and also the whole grid are shown in Figure 6.3. Note that the leading edge region of the profile is already overly refined in this grid, which the present algorithms are unable to correct. Using a second-order box method (*i.e.*, a node-centred scheme with continuous piecewise linear recovery, as described in earlier sections) one obtains for the pressure distribution the results shown in Figure 6.4. Note that although both shocks are visible, the weak lower side shock is quite badly resolved.

Applying three refinement cycles using the unweighted L^2-norm as the refinement indicator results in the mesh shown in Figure 6.5. The corresponding pressure distribution is shown on the right-hand side. The results are undoubtedly much improved but there has been a lot of unnecessary refinement around the leading edge.

For comparison, the grid after three refinement cycles with the weighted L^2-norm indicator and the corresponding Mach number distribution are shown in Figure 6.6. There is clearly a much more appropriate refinement of the mesh, resulting in a much better defined solution.

However, it was the use of the dual problem in the *a posteriori* analysis of finite element approximations to elliptic equations that initially led to the special place of the L^2-norm. Developments for convection-diffusion

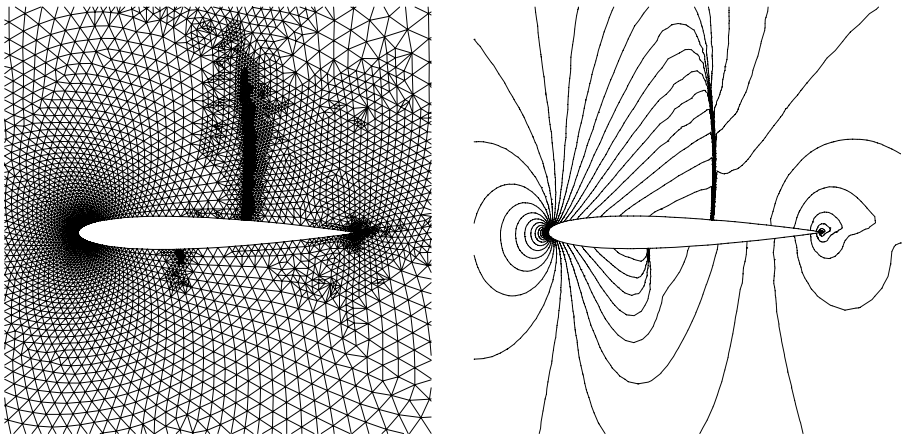


Figure 6.6. Grid after three refinement cycles with the weighted norm indicator; and the calculated Mach number distribution.

problems in Eriksson and Johnson (1993) and earlier papers then led to the weighted norm (6.2). But it was less clear whether this should be carried over to hyperbolic problems approximated by finite volume methods. Sonar (1993*b*) therefore experimented with various alternatives to this norm: in particular, there are theoretical arguments for using the weak norm

$$\|\mathbf{r}^h\|_{H^{-1}(T)} := \sup_{\Phi\in H_0^1} \frac{|(\mathbf{r}^h,\Phi)_T|}{\|\Phi\|_{H_0^1(T)}},$$

where the supremum is taken over all H_0^1-functions on triangle T. To approximate this, each edge of each triangle was divided into four equal parts and straight lines parallel to the edges drawn between them; their intersections define three interior points of the triangle and thence three hat functions Φ_i, $i = 1, 2, 3$, which take the value 1 at the subdivision node i and 0 elsewhere. Then the weak norm is approximated by taking the maximum over these choices, rather than the supremum over all $\Phi \in H_0^1(T)$. Results obtained with this norm were never better than with the weighted L^2-norm, were more sensitive to chosen tolerances and in the NACA0012 problem led to unwarranted mesh refinement well away from the profile.

More recently, Süli and Houston (1997) have given a very clear account of the Johnson (1994) paradigm, and then adopted an alternative but related approach for general finite element approximations to hyperbolic equations, together with an application to the cell-vertex method. These results together with the success of the weighted L^2-norm refinement indicator have prompted further theoretical developments which we will describe next before coming back to more extensive numerical tests. The analysis was first

developed for the symmetric positive PDEs studied by Friedrichs (1958), and now called *Friedrichs systems*; so we begin by putting the Euler equations in this form.

6.3. Symmetrizing the Euler equations

It is part of the general theory of hyperbolic systems that they may be symmetrized (*i.e.*, put in a form in which the flux Jacobian matrices are symmetric) by changing to *entropy variables* (Moch 1980), and this has been exploited for the Euler equations by Hughes, Franca and Mallet (1986). In the case of an ideal gas the entropy density is given by

$$\eta(\mathbf{u}) := -\rho s,$$

where $s := \ln(p\rho^{-\gamma})$ denotes the thermodynamic entropy. We then introduce new variables, the *entropy variables*, by means of the transformation

$$\mathbf{u} \longmapsto \mathbf{U}(\mathbf{u}) := \nabla_{\mathbf{u}}\eta(\mathbf{u}). \tag{6.3}$$

Explicitly, with the pressure p given by (2.13) and in terms of the primitive variables, this has the form

$$\mathbf{U}(\mathbf{u}) = \frac{\gamma-1}{p}\begin{bmatrix} \frac{p}{\gamma-1}(\gamma+1-s) - \rho E \\ \rho v_1 \\ \rho v_2 \\ -\rho \end{bmatrix} =: \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \tag{6.4}$$

and the inverse mapping $\mathbf{U} \longmapsto \mathbf{u}$ is given by

$$\mathbf{u}(\mathbf{U}) = \frac{p}{\gamma-1}\begin{bmatrix} -U_4 \\ U_2 \\ U_3 \\ 1 - \frac{1}{2}\frac{U_2^2+U_3^2}{U_4} \end{bmatrix}, \tag{6.5}$$

where we now need to write $p = p(\mathbf{U})$ in terms of the entropy variables. Substituting for $\mathbf{u}$ in the Euler equations from (6.5), applying the chain rule and using the notation $A_0(\mathbf{U}) := \nabla_{\mathbf{U}}\mathbf{u}$, we can then write them as

$$A_0(\mathbf{U})\partial_t\mathbf{U} + \sum_{i=1}^{2}(\nabla_{\mathbf{u}}\mathbf{f}_i(\mathbf{u}(\mathbf{U})))A_0(\mathbf{U})\partial_{x_i}\mathbf{U} = \mathbf{0}, \tag{6.6}$$

which is in the form we are seeking.

Explicit expressions for A_0 and its inverse A_0^{-1} were derived by Hughes *et al.* (1986) and are given in Sonar and Süli (1998); they are quite complicated and it is simplest to consider them via the intermediate system of primitive variables, using the transformation (2.16). Fortunately, they are not needed in order to show that all the coefficient matrices in this new

form of the equations are symmetric: that is, if we write the system in the compact notation

$$\sum_{i=0}^{2} A_i(\mathbf{U})\partial_{x_i}\mathbf{U} = \mathbf{0}, \tag{6.7}$$

with $x_0 := t$ and in which $A_i(\mathbf{U}) := \nabla_{\mathbf{u}}\mathbf{f}_i(\mathbf{u}(\mathbf{U}))A_0(\mathbf{U}) \equiv \nabla_{\mathbf{U}}\mathbf{f}_i$, $i = 1, 2$, these matrices and $A_0(\mathbf{U})$ are all symmetric 4×4 matrices. Following Sonar and Süli (1998), to show that this is so we introduce the scalar quantities

$$r(\mathbf{U}) = \mathbf{U}^T\mathbf{u} - \eta, \quad \text{and} \quad s_i(\mathbf{U}) = \mathbf{U}^T\mathbf{f}_i - q_i, \quad i = 1, 2,$$

where the q_i are the entropy fluxes. Then it is easily seen that

$$\nabla_{\mathbf{U}} r(\mathbf{U}) = \mathbf{u} + (\nabla_{\mathbf{U}}\mathbf{u})^T\mathbf{U} - (\nabla_{\mathbf{U}}\mathbf{u})^T\nabla_{\mathbf{u}}\eta = \mathbf{u};$$

and in the same way, by making use of the relation (2.5) satisfied by the entropy fluxes, we have

$$\nabla_{\mathbf{U}} s_i(\mathbf{U}) = \mathbf{f}_i + (\nabla_{\mathbf{U}}\mathbf{f}_i)^T\mathbf{U} - (\nabla_{\mathbf{U}}\mathbf{u})^T\nabla_{\mathbf{u}} q_i = \mathbf{f}_i.$$

It follows that $A_0(\mathbf{U}) \equiv \nabla_{\mathbf{U}}\mathbf{u}$ is the Hessian of the scalar r and is therefore symmetric: similarly, for $i = 1, 2$, $A_i(\mathbf{U}) \equiv \nabla_{\mathbf{U}}\mathbf{f}_i$ is the Hessian of the scalar s_i and is therefore symmetric.

In order to carry forward the error analysis developed for Friedrichs systems, we next need to carry out a local linearization. This is done about a constant mean state in each cell: that is, we assume the existence of a constant state $\mathbf{U}_c \in \mathbb{R}^4$ such that the decomposition

$$\mathbf{U} = \mathbf{U}_c + \mathbf{V}$$

holds for a small non-constant perturbation function $\mathbf{V}$. It follows that

$$\begin{aligned}\mathbf{u}(\mathbf{U}) &= \mathbf{u}(\mathbf{U}_c + \mathbf{V}) = \mathbf{u}(\mathbf{U}_c) + \nabla_{\mathbf{U}}\mathbf{u}(\mathbf{U}_c)\mathbf{V} + \mathcal{O}(|\mathbf{V}|^2)\\ &= \mathbf{u}(\mathbf{U}_c) + A_0(\mathbf{U}_c)\mathbf{V} + \mathcal{O}(|\mathbf{V}|)^2;\end{aligned}$$

and in a similar way, with $\mathbf{F}_i(\mathbf{U}) := \mathbf{f}_i(\mathbf{u}(\mathbf{U}))$, we have

$$\mathbf{f}_i(\mathbf{u}(\mathbf{U})) = \mathbf{F}_i(\mathbf{U}_c) + \nabla_{\mathbf{u}}\mathbf{f}_i(\mathbf{u}(\mathbf{U}_c))A_0(\mathbf{U}_c)\mathbf{V} + \mathcal{O}(|\mathbf{V}|^2). \tag{6.8}$$

Writing $\mathbf{u}_c := \mathbf{u}(\mathbf{U}_c)$ and dropping the $\mathcal{O}(|\mathbf{V}|^2)$ terms, we then obtain the symmetric system

$$A_0(\mathbf{U}_c)\partial_t\mathbf{V} + \sum_{i=1}^{2} \nabla_{\mathbf{u}}\mathbf{f}_i(\mathbf{u}_c)A_0(\mathbf{U}_c)\partial_{x_i}\mathbf{V} = \mathbf{0}, \tag{6.9}$$

in which all matrix elements are constant. Finally, we write this in standard form as

$$L_E\mathbf{V} := \sum_{i=0}^{2} A_i(\mathbf{U}_c)\partial_{x_i}\mathbf{V} = \mathbf{0}, \tag{6.10}$$

where the A_i are given by

$$A_i(\mathbf{U}_c) := \nabla_{\mathbf{u}} \mathbf{f}_i(\mathbf{u}_c) A_0(\mathbf{U}_c), \quad i = 1, 2. \tag{6.11}$$

6.4. Dual graph-norm error indicators for Friedrichs systems

The *a posteriori* error analysis developed in Houston, Mackenzie, Süli and Warnecke (1999) and earlier papers was for a more general Friedrichs system than (6.10), as was that in Sonar and Süli (1998) where it was applied to the Euler equations. In summarizing these presentations, we therefore consider the system

$$L\mathbf{U} := \sum_{j=0}^{2} A_j(\mathbf{x}) \partial_{x_j} \mathbf{U} + C(\mathbf{x})\mathbf{U} = \mathbf{0}, \tag{6.12}$$

where $\mathbf{x} = (x_0, x_1, x_2)^T := (t, x_1, x_2)^T$ is a space-time coordinate. Here the matrices A_j are symmetric with Lipschitz-continuous elements, the matrix C has continuous elements and we will assume that A_0 is positive definite. By assuming that it is symmetric positive definite in a region Ω we mean that there exists a positive constant $c_0 \equiv c_0(\Omega)$ such that

$$\tfrac{1}{2}\big(K(\mathbf{x}) + K^*(\mathbf{x})\big) \geq c_0 I, \tag{6.13}$$

for all $\mathbf{x} \in \overline{\Omega}$, where for some $\xi \in \mathbb{R}^3$, with $|\xi| = 1$, the matrix K is defined by

$$K := C - \tfrac{1}{2} \sum_{j=0}^{2} \partial_{x_j} A_j + \sum_{j=0}^{2} \xi_j A_j.$$

For the error analysis of such a system we need to distinguish the error generated within each cell and that transported from one cell to its neighbours. So for the unsteady problem integrated over one time step we consider a space-time prism $P_i^n := (n\Delta t, (n+1)\Delta t) \times T_i$ based on a triangle $T_i \in \mathcal{T}^h$, and introduce the matrix

$$B(\mathbf{x}) := \sum_{j=0}^{2} n_j A_j(\mathbf{x}),$$

where $\mathbf{n} = (n_0, n_1, n_2)^T$ denotes the unit outward normal vector to its boundary ∂P_i^n at a point $\mathbf{x}$. We suppose that B is non-singular at each such point, *i.e.*, ∂P_i^n is a non-characteristic hypersurface for the operator L. Now we split B into a negative semi-definite part B^- and a positive semi-definite part $B^+ = B - B^-$. We call $B^-\mathbf{U}$ the inflow part of the vector field $\mathbf{U}$ and $B^+\mathbf{U}$ its outflow part. Then it was shown in Friedrichs (1958) and Lax and Phillips (1960) that symmetric hyperbolic systems have unique strong solutions subject to a boundary condition that specifies $B^-\mathbf{U}$.

Now suppose that our numerical approximation $\mathbf{u}^h$ is converted into entropy variables to give $\mathbf{U}^h$. We consider the following boundary value problem in P_i^n:

$$L\hat{\mathbf{U}}^h = \mathbf{0} \quad \text{on} \quad P_i^n$$
$$B^-\hat{\mathbf{U}}^h\big|_{\partial P_i^n} = B^-\mathbf{U}^h\big|_{\partial P_i^n}.$$

We interpret the function $\hat{\mathbf{U}}^h$ as the exact solution of (6.12) in P_i^n with inflow boundary data contaminated by the *transported error* carried by $\mathbf{U}^h$. Hence we define the *cell error* by

$$\mathbf{e}_c \equiv \mathbf{e}_{P_i^n}^{\text{cell}} := \mathbf{U}^h - \hat{\mathbf{U}}^h, \tag{6.14}$$

being the error in the numerical solution which is produced on P_i^n; while the transported error is given by

$$\mathbf{e}_t \equiv \mathbf{e}_{P_i^n}^{\text{trans}} := \hat{\mathbf{U}}^h - \mathbf{U}. \tag{6.15}$$

The sum of these two is the total error $\mathbf{e}_{P_i^n} \equiv \mathbf{U}^h - \mathbf{U}$.

It is clear that the residual calculated in a given prism has no control over the transported error, which is just advected into the cell from upwind: while the cell error is governed directly by the residual via the relation

$$\mathbf{r}^h = L\mathbf{e}_{P_i^n} = L\mathbf{e}_{P_i^n}^{\text{cell}} \equiv L\mathbf{e}_c \quad \text{on} \quad P_i^n, \tag{6.16}$$

which is subject to a zero inflow boundary condition. Our next objective then is to obtain two-sided bounds of the form (6.1) for the cell error and cell residual. Note that this will only give some confidence in the overall accuracy of a computation if the cell errors dominate the transported errors; but in any case it should give a reliable indicator for local mesh refinement.

To develop such error bounds, it was shown in Houston *et al.* (1999) that one can define the following spaces and their associated norms: for certain weight functions w_i^n, the weighted graph-norm $\|\cdot\|_{D(L,P_i^n)}$ on

$$D_-(L,P_i^n) := \{\phi \in L^2(P_i^n) \,|\, L\phi \in L^2(P_i^n), \quad B^-\phi = 0 \quad \text{on} \quad \partial P_i^n\},$$

is defined by

$$\|\phi\|_{D(L,P_i^n)} = \left[\|w_i^n\phi\|^2_{L^2(P_i^n)} + \|w_i^n L\phi\|^2_{L^2(P_i^n)}\right]^{1/2},$$

and the associated dual graph-norm by

$$\|v\|_{D'(L,P_i^n)} := \sup_{\phi\in D_-(L,P_i^n)} \frac{|(v,\phi)_{P_i^n}|}{\|\phi\|_{D(L,P_i^n)}},$$

where $(\cdot,\cdot)_{P_i^n}$ denotes the usual L^2 inner product on P_i^n. Similarly, by

introducing the formal adjoint

$$L^*\phi := -\sum_{j=0}^{2} \partial_{x_j}(A_j\phi) + C^*\phi,$$

with

$$\phi \in D_+(L^*, P_i^n) := \{\phi \in L^2(P_i^n) \,|\, L^*\phi \in L^2(P_i^n), \quad B^+\phi = 0 \ \text{ on } \ \partial P_i^n\},$$

we can equip $D_+(L^*, P_i^n)$ with a corresponding graph-norm and associated dual graph-norm.

Establishing the necessary trace theorems in these spaces is a nontrivial part of the analysis which goes on to establish the following local *a posteriori* error bound.

Theorem 6.1. For the symmetric hyperbolic system (6.12), the cell error satisfies the following inequalities:

$$\begin{aligned} \big(\min_{P_i^n} w_i^n\big)\, \|\mathbf{r}^h\|_{D'(L^*,P_i^n)} &\le \|\mathbf{e}_c\|_{L^2(P_i^n)} \qquad (6.17)\\ &\le \left(1 + \frac{1}{c_0^2}\right)^{1/2} \big(\max_{P_i^n} w_i^n\big)\, \|\mathbf{r}^h\|_{D'(L^*,P_i^n)}, \end{aligned}$$

where $c_0 \equiv c_0(P_i^n)$ is as defined in (6.13).

Proof. See Sonar (2002) or Sonar and Süli (1998). □

In order to exploit this theorem by calculating the dual graph-norm of a residual on a space-time prism, we need first to consider the inflow and outflow boundary conditions for the Euler equations. It is more convenient to work with the unsymmetric form of the equations in the entropy variables obtained by premultiplying (6.9) by A_0^{-1}. Thus, denoting the resultant matrices by $\tilde{A}_j$, we have

$$\tilde{B}(P_i^n) = \sum_{j=0}^{2} n_j \tilde{A}_j \equiv n_0 I + \sum_{j=1}^{2} n_j A_0^{-1}(\mathbf{U}_{ci}) \nabla_{\mathbf{u}} \mathbf{f}_j(\mathbf{u}_{ci}) A_0(\mathbf{U}_{ci}), \qquad (6.18)$$

where $\mathbf{U}_{ci}, \mathbf{u}_{ci}$ denote constant mean states of entropy and conservative variables, respectively, within the prism P_i^n. Then, as above, we have the inflow/outflow subdivision

$$\tilde{B}(P_i^n) = \tilde{B}^-(P_i^n) + \tilde{B}^+(P_i^n).$$

In particular, it is clear that the bottom of the prism is an inflow boundary while the top is an outflow boundary.

For the sides, we note that the matrices $\tilde{A}_j$ are similar to $\nabla_{\mathbf{u}}\mathbf{f}_j$, for $j = 1, 2$, and $\tilde{A}_0$ is similar to I; so each eigenvalue of $\tilde{B}(P_i^n)$ is given by

$$\operatorname{eig}(\tilde{B}(P_i^n)) = n_0 + \operatorname{eig}\left(\sum_{j=1}^{2} n_j \nabla_{\mathbf{u}}\mathbf{f}_j(\mathbf{u}_{ci})\right),$$

in terms of the eigenvalues of $\nabla_{\mathbf{u}}\mathbf{f}_j$ which were given in Section 2.3. Thus we write

$$\Lambda(\mathbf{u}_{ci}, n_1, n_2) := \operatorname{diag}\big\{v_n, v_n, v_n + c_{ci}|(n_1, n_2)|, v_n - c_{ci}|(n_1, n_2)|\big\},$$

where $v_n = \sum_{j=1}^{2} n_j v_{ci,j}$ is the flow speed in the normal direction and c_{ci} is the mean constant speed of sound in P_i^n. We split Λ into a matrix Λ^+ containing the positive eigenvalues, and Λ^- containing the negative eigenvalues. So we finally have a representation for the boundary matrix $\tilde{B}(P_i^n)$ in the form

$$\tilde{B}(P_i^n) = n_0 I + A_0^{-1}(\mathbf{U}_{ci})\big[P\Lambda^+ P^{-1}(\mathbf{u}_{ci}, n_1, n_2)\big]A_0(\mathbf{U}_{ci}) \tag{6.19}$$

$$+ A_0^{-1}(\mathbf{U}_{ci})\big[P\Lambda^- P^{-1}(\mathbf{u}_{ci}, n_1, n_2)\big]A_0(\mathbf{U}_{ci}), \tag{6.20}$$

where $P(\mathbf{u}_{ci}, n_1, n_2)$ is the matrix which diagonalizes $\sum_{j=1}^{2} n_j \nabla_{\mathbf{u}}\mathbf{f}_j(\mathbf{u}_{ci})$. Note that this subdivision essentially corresponds to the flux vector splitting of Steger and Warming (1981).

There are several ways in which these formulae may be approximated to yield a practical refinement indicator. The simplest, whose use is reported on in Sonar (2002), is based on using an explicit time-stepping procedure so that divided differences are used instead of space-time basis functions in the prisms P_i^n. Then the graph-norm calculation is approximated by subdividing each triangle into 16 equal subtriangles, in the manner described above in connection with calculating the weak H^{-1}-norm. In this case we obtain three interior nodes and twelve boundary nodes for each triangle, giving 15 different test functions. Details of the calculation are given in Sonar (2002).

The adaptive procedure using this graph-norm refinement indicator makes use of two tolerances $\mathrm{TOL}_{\mathrm{refine}}$ and $\mathrm{TOL}_{\mathrm{coarse}}$ for the refinement and coarsening algorithms described above, and their choice in this case is quite critical. A typical mesh obtained in the case of the transonic NACA0012 flow problem described earlier is shown in Figure 6.7. The flow features are very well captured and the indicator has started to detect the supersonic region on the upper side. Another nice feature in comparison with the

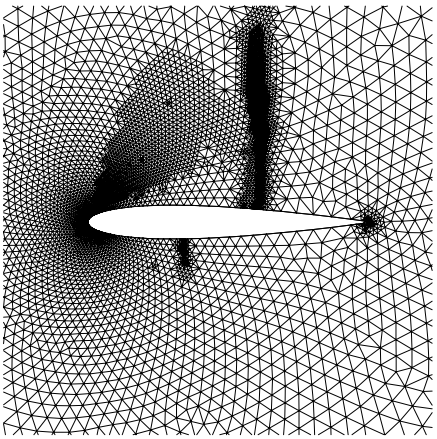


Figure 6.7. Grid after three refinement/coarsening cycles with the graph-norm indicator for transonic flow about the NACA0012 aerofoil.

H^{-1}-indicator described above is the absence of any noise spoiling the grid far away from the obstacle.

Although these and other results look quite promising there are several problems associated with the dual graph-norm refinement indicator. First of all, it is very expensive to compute in comparison with the weighted L^2-norm indicator. The second problem concerns its sensitivity: it turns out that in some calculations a small change in tolerance can influence the adapted grid enormously, which makes it hard to use in practice. This seems to arise from the inability of the indicator to detect contact discontinuities. Results which illustrate these points may be found in Sonar (2002).

6.5. Closing the loop and further tests

As claimed in Sonar and Süli (1998), the dual graph-norm error indicator which was described in the previous subsection seems to have been the first effective refinement indicator for the Euler equations with a sound mathematical foundation. On the other hand it has several practical disadvantages which were also indicated there. The more practical indicator would seem to be that based on the weighted L^2-norm, which gave the results in Figure 6.6.

It was therefore an important step towards resolving this dilemma when the following result was proved in Houston *et al.* (1999): for positive constants C and C' we have

$$C'h_0\|P_{h_0}\mathbf{r}^h\|_{L^2(P_i^n)} \leq \|\mathbf{e}_c\|_{L^2(P_i^n)} \leq Ch\|\mathbf{r}^h\|_{L^2(P_i^n)}, \tag{6.21}$$

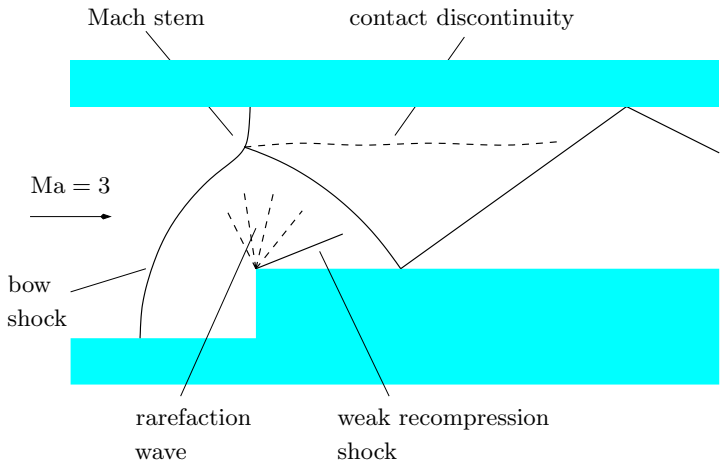


Figure 6.8. Flow phenomena in the channel with forward facing step.

where h denotes the diameter of P_i^n, and P_{h_0} is the orthogonal projector onto a finite element space on a micropartition of P_i^n of diameter h_0. This micropartition corresponds to that used for approximating the graph-norms of the bounds in (6.17).

With this result we finally have a practical refinement indicator which rests on a firm theoretical base. So we conclude this survey by giving some results for the test problem due to Woodward and Colella (1984), which we used earlier. This is actually an unsteady flow problem in which the forward-facing step is inserted into a steady, uniform Ma = 3.0 flow down the channel. A complicated shock system develops whose steady state is sketched in Figure 6.8: note too the contact discontinuity which emerges from the point where the bow shock joins the Mach stem attached to the upper boundary.

Computations for this problem were carried out with the unsteady DLR-τ-code of Sonar (1993*a*), Sonar, Hannemann and Hempel (1994) and Meister (1994), and reported in Sonar (2002). In Figure 6.9 we show the meshes that were generated at various times using the weighted L_2-norm refinement indicator. It is seen to have detected all the relevant flow phenomena: the shock system is clearly visible, as is the contact discontinuity starting at the Mach stem. Note that the corner point of the step is a true corner singularity since it corresponds to the centre of a rarefaction wave: the indicator has also detected phenomena associated with this special point and refined the region in its vicinity. In Figure 6.10 we show the density distributions obtained on these meshes.

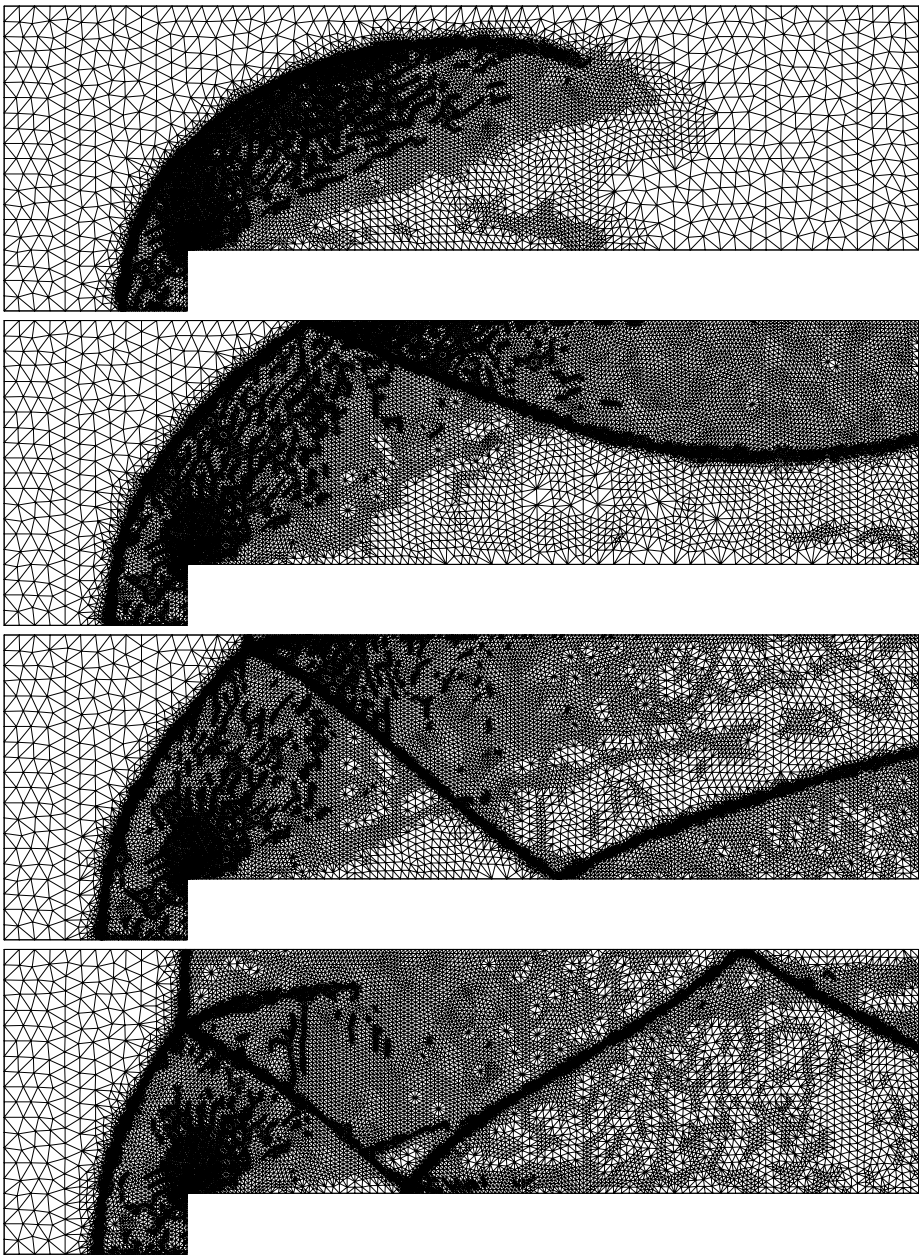


Figure 6.9. Adapted grids at different times
for the Woodward and Colella problem.

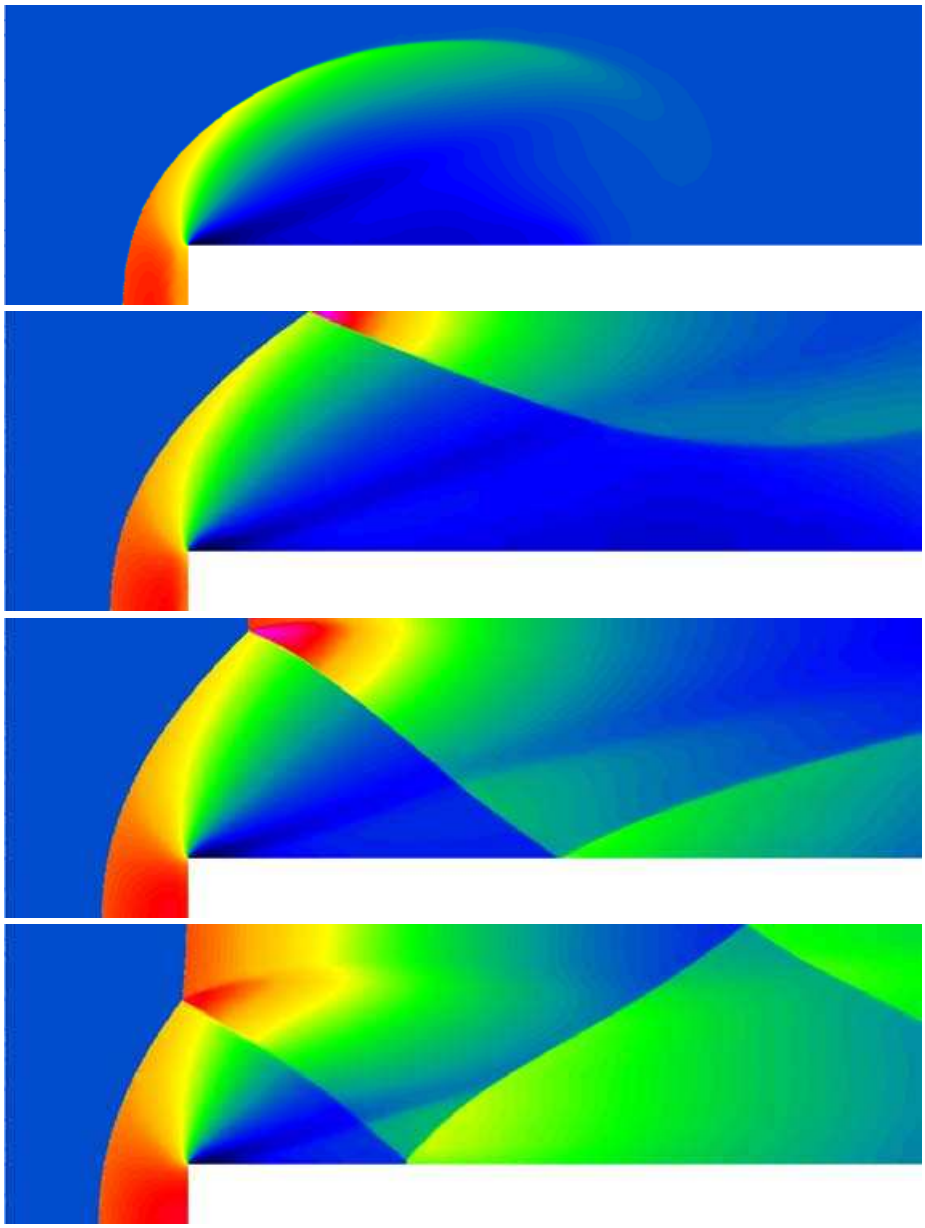


Figure 6.10. Density distributions corresponding to the grids in Figure 6.9.

7. Concluding remarks

- Finite volume methods share with finite element methods the viewpoint that it is primarily the solution that is being approximated, rather than the equation or any operator in it.
- Their key guiding principle is exact satisfaction of the integral conservation laws; so they are at their most effective where solutions contain shocks or other discontinuities.
- Their advantage over evolution-Galerkin methods is that even explicit methods need a less good approximation to the evolution operator defined by the PDE as it is used only to calculate the fluxes.
- Limited as they are to using only piecewise constant functions as test functions, with the consequential heavy dependence on the recovery stage, they may well be superseded by discontinuous Galerkin methods, but only when these methods recognize finite volume methods as their proper antecedents and learn from them.
- The jury was out for a long time in the judgement between cell-centre and cell-vertex methods; but it now seems that node-centred schemes have acquired an edge over either. Cell-centre schemes still hold centre stage as regards practical codes, because of their more reliable representation of shocks as compared with cell-vertex methods. But node-centred methods have advantages over cell-centre methods in regard to generating hierarchies of recovery procedures.
- Although the ideal of a guaranteed error bound derived from an *a posteriori* residual is still not achievable for finite volume computations of nonlinear hyperbolic conservation laws, progress to that end during the past decade has been quite remarkable: in particular, soundly based practical mesh refinement indicators are now available.
- Further progress with these methods is likely to lie with implicit algorithms, exploiting the techniques developed in the optimization field for the rapid solution of large nonlinear systems of equations. This is consistent with the practical requirement in aerodynamics that flow calculations should be fully incorporated into the design cycle.

Acknowledgements

We are grateful to Phil Roe, Gil Strang and Endre Süli for their comments on a draft of the paper.

REFERENCES

R. Abgrall (1994*a*), 'An essentially non-oscillatory reconstruction procedure on finite-element type meshes: Application to compressible flows', *Comput. Methods Appl. Mech. Engrg.* **116**, 95–101.

R. Abgrall (1994*b*), 'On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation', *J. Comput. Phys.* **114**, 45–58.

R. Abgrall and T. Sonar (1997), 'On the use of Mühlbach expansions in the recovery step of ENO methods', *Numer. Math.* **76**, 1–27.

K. J. Badcock and B. E. Richards (1995), 'Implicit time stepping methods for the Navier–Stokes equations', *AIAA Journal* **34**, 555–559.

R. E. Bank, A. H. Sherman and A. Weiser (1983), Refinement algorithms and data structures for local regular mesh refinements, in *Scientific Computing* (R. Stepleman *et al.*, eds), IMACS North-Holland, pp. 3–17.

J. W. Barrett, G. Moore and K. W. Morton (1988*a*), 'Optimal recovery in the finite element method, Part I: Recovery from weighted L^2 fits', *IMA J. Numer. Anal.* **8**, 149–184.

J. W. Barrett, G. Moore and K. W. Morton (1988*b*), 'Optimal recovery in the finite element method, Part II: Defect correction for ordinary differential equations', *IMA J. Numer. Anal.* **8**, 527–540.

M. Ben-Artzi and J. Falcovitz (1984), 'A second order Godunov-type scheme for compressible fluid dynamics', *J. Comput. Phys.* **55**, 1–32.

G. Birkhoff (1983), 'Numerical fluid dynamics', *SIAM Rev.* **25**, 1–34.

C. de Boor (2001), *A Practical Guide to Splines*, revised edn, Springer, New York.

A. Brandt (1977), 'Multilevel adaptive solutions to boundary-value problems', *Math. Comp.* **31**, 333–390.

Y. Brenier (1984), 'Average multi-valued solutions for scalar conservation laws', *SIAM J. Numer. Anal.* **21**, 1013–1037.

G. Bruhn (1985), 'Erhaltungssätze und schwache Lösungen in der Gasdynamik', *Math. Methods Appl. Sci.* **7**, 470–479.

D. S. Butler (1960), 'The numerical solution of hyperbolic systems of partial differential equations in three independent variables', *Proc. Roy. Soc.* **255A**, 233–252.

P. N. Childs and K. W. Morton (1990), 'Characteristic Galerkin methods for scalar conservation laws in one dimension', *SIAM J. Numer. Anal.* **27**, 553–594.

P. G. Ciarlet (1987), *The Finite Element Method for Elliptic Problems*, 2nd edn, North-Holland.

P. G. Ciarlet and P. R. Raviart (1972), 'General Lagrange and Hermite interpolation in R^n with applications to the finite element method', *Arch. Rat. Mech. Anal.* **46**, 177–199.

B. Cockburn, G. E. Karniadakis and C. W. Shu (2000), The development of discontinuous Galerkin methods, in *Proc. First International Symposium on Discontinuous Galerkin Methods*, Springer, New York, pp. 3–50.

P. Colella and P. R. Woodward (1984), 'The piecewise parabolic method (PPM) for gas-dynamical simulations', *J. Comput. Phys.* **54**, 174–201.

R. Courant, K. O. Friedrichs and H. Lewy (1928), 'Über die partiellen Differenzengleichungen der Physik', *Math. Ann.* **100**, 32–74.

P. I. Crumpton, J. A. Mackenzie and K. W. Morton (1993), 'Cell vertex algorithms for the compressible Navier–Stokes equations', *J. Comput. Phys.* **109**, 1–15.

J. A. Cunge, F. M. Holly and A. Verwey (1980), *Practical Aspects of Computational River Hydraulics*, Pitman, London.

H. Deconinck, P. L. Roe and R. Struijs (1993), 'A multidimensional generalization of Roe's flux difference splitter for the Euler equations', *Comput. Fluids* **22**, 215–222.

J. Douglas, Jr. and T. F. Russell (1982), 'Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures', *SIAM J. Numer. Anal.* **19**, 321–352.

L. J. Durlofsky, B. Engquist and S. Osher (1992), 'Triangle based adaptive stencils for the solution of hyperbolic conservation laws', *J. Comput. Phys.* **98**, 64–73.

M. Eiermann and O. G. Ernst (2001), Geometric aspects of the theory of Krylov subspace methods, in *Acta Numerica*, Vol. 10 (A. Iserles, ed.), Cambridge University Press, pp. 251–312.

H. C. Elman, D. J. Silvester and A. J. Wathen (2005), *Finite Elements and Fast Iterative Solvers*, Oxford University Press.

B. Engquist and S. Osher (1981), 'One-sided difference approximations for nonlinear conservation laws', *Math. Comp.* **36**, 321–352.

K. Eriksson and C. Johnson (1993), 'Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems', *Math. Comp.* **60**, 167–188.

K. Eriksson, D. Estep, P. Hansbo and C. Johnson (1995), Introduction to adaptive methods for differential equations, in *Acta Numerica*, Vol. 4 (A. Iserles, ed.), Cambridge University Press, pp. 105–158.

R. P. Federenko (1964), 'The speed of convergence of one iterative process', *USSR Comp. Math. and Math. Phys.* **4**, 227–235.

L. Fezoui and B. Stoufflet (1989), 'A class of implicit upwind schemes for Euler simulations with unstructured meshes', *J. Comput. Phys.* **84**, 174–206.

M. A. Freitag and K. W. Morton (2007), 'The Preissmann box scheme and its modification for transcritical flows', *Internat. J. Numer. Methods Engrg.*, to appear.

O. Friedrich (1998), 'Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids', *J. Comput. Phys.* **144**, 194–212.

K. O. Friedrichs (1958), 'Symmetric positive linear differential operators', *Comm. Pure Appl. Math.* **11**, 333–418.

E. Godlewski and P.-A. Raviart (1991), *Hyperbolic Systems of Conservation Laws*, Ellipses, Paris.

S. K. Godunov (1959), 'A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics', *Mat. Sb.* **47**, 271–306.

M. Golomb and H. F. Weinberger (1959), Optimal approximation and error bounds, in *Symposium on Numerical Approximation* (R. E. Langer, ed.), University of Wisconsin Press, Madison, pp. 117–190.

A. Haar (1919), 'Über die Variation der Doppelintegrale', *J. Reine Angew. Math.* **149**, 1–18.

E. Hairer and G. Wanner (1996), *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer, Berlin/Heidelberg.

P. Hansbo and C. Johnson (1991), 'Adaptive streamline diffusion methods for compressible flow using conservation variables', *Comput. Methods Appl. Mech. Engrg.* **87**, 267–280.

A. Harten (1983), 'High resolution schemes for conservation laws', *J. Comput. Phys.* **49**, 357–393.

A. Harten (1984), 'On a class of high resolution total-variation-stable finite-difference schemes', *SIAM J. Numer. Anal.* **21**, 1–23.

A. Harten (1989), 'ENO schemes with subcell resolution', *J. Comput. Phys.* **83**, 148–184.

A. Harten and S. R. Chakravarthy (1991), Multi-dimensional ENO schemes for general geometries. Report no. 91-76, ICASE.

A. Harten and S. Osher (1987), 'Uniformly high-order nonoscillatory schemes I', *SIAM J. Numer. Anal.* **24**, 279–309.

A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy (1987), 'Uniformly high order accurate essentially non-oscillatory schemes III', *J. Comput. Phys.* **71**, 231–303.

A. Harten, P. Lax and B. van Leer (1983), 'On upstream differencing and Godunov-type schemes for hyperbolic conservation laws', *SIAM Rev.* **25**, 35–61.

A. Harten, S. Osher, B. Engquist and S. R. Chakravarthy (1986), 'Some results on uniformly high-order accurate essentially nonoscillatory schemes', *Appl. Numer. Math.* **2**, 347–377.

M. R. Hestenes and E. Stiefel (1952), 'Methods of conjugate gradients for solving linear problems', *J. Res. Nat. Bur. Stand.* **49**, 409–436.

C. Hirsch (1988), *Numerical Computation of Internal and External Flows, Vol. 1: Fundamentals of Numerical Discretization*, Wiley.

C. Hirsch (1990), *Numerical Computation of Internal and External Flows, Vol. 2: Computational Methods for Inviscid and Viscous Flows*, Wiley.

P. Houston, J. M. Mackenzie, E. Süli and G. Warnecke (1999), '*A posteriori* error analysis for numerical approximation of Friedrichs systems', *Numer. Math.* **82**, 433–470.

T. J. R. Hughes, L. P. Franca and M. Mallet (1986), 'A new finite element formulation for compressible fluid dynamics I: Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics', *Comput. Methods Appl. Mech. Engrg.* **54**, 223–234.

A. Iske and T. Sonar (1996), 'On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions', *Numer. Math.* **74**, 177–201.

A. M. Jaffe (2006), 'The Millennium Grand Challenge in Mathematics', *Notices Amer. Math. Soc.* **53**, 652–660.

A. Jameson (1979), Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method, in *AIAA 4th Computational Fluid Dynamics Conference,* Paper 79-1458.

A. Jameson, W. Schmidt and E. Turkel (1981), Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes, in *AIAA 14th Fluid and Plasma Dynamics Conference,* Paper 81-1259.

C. Johnson (1994), A new paradigm for adaptive finite element methods, in *The Mathematics of Finite Element Methods and Applications: Highlights 1993* (J. R. Whiteman, ed.), Wiley, pp. 105–120.

R. Klötzler (1970), *Mehrdimensionale Variationsrechnung*, Birkhäuser.

S. Kružkov (1970), 'First-order quasilinear equations in several variables', *Math. USSR Sb.* **10**, 217–243.

P. D. Lax and R. S. Phillips (1960), 'Local boundary conditions for dissipative symmetric linear differential operators', *Comm. Pure Appl. Math.* **13**, 427–455.

P. D. Lax and B. Wendroff (1960), 'Systems of conservation laws', *Comm. Pure Appl. Math.* **13**, 217–237.

B. van Leer (1979), 'Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method', *J. Comput. Phys.* **32**, 101–136.

B. P. Leonard (1991), 'The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection', *Comput. Methods Appl. Mech. Engrg.* **88**, 17–74.

P. Lesaint (1977), Numerical solution of the equation of continuity, in *Topics in Numerical Analysis III* (J. J. H. Miller, ed.), Academic Press, pp. 199–222.

R. J. LeVeque (2002), *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press.

P. Lin, K. W. Morton and E. Süli (1993), 'Euler characteristic Galerkin scheme with recovery', *Math. Modelling Numer. Anal.* **27**, 863–894.

P. Lin, K. W. Morton and E. Süli (1997), 'Characteristic Galerkin schemes for scalar conservation laws in two and three space dimensions', *SIAM J. Numer. Anal.* **34**, 779–796.

M.-S. Liou and C. J. Steffen, Jr. (1993), 'A new flux splitting scheme', *J. Comput. Phys.* **107**, 23–39.

X.-D. Liu, S. Osher and T. Chan (1994), 'Weighted essentially non-oscillatory schemes', *J. Comput. Phys.* **115**, 200–212.

M. Lukáčová-Medvid'ová, K. W. Morton and G. Warnecke (2000), 'Evolution-Galerkin methods for hyperbolic systems in two space dimensions', *Math. Comp.* **69**, 1355–1384.

M. Lukáčová-Medvid'ová, K. W. Morton and G. Warnecke (2002), 'Finite volume evolution-Galerkin methods for Euler equations of gas dynamics', *Internat. J. Numer. Methods Fluids* **40**, 425–434.

M. Lukáčová-Medvid'ová, K. W. Morton and G. Warnecke (2004), 'Finite volume evolution-Galerkin methods for hyperbolic systems', *SIAM J. Sci. Comput.* **26**, 1–30.

P. W. McDonald (1971), The computation of transonic flow through two-dimensional gas turbine cascades, in *ASME Proc.*, Paper 71-GT-89, ASME, New York.

D. J. Mavriplis (1995), Multigrid techniques for unstructured meshes, in *VKI Lecture Series* VKI-LS 1995-02, von Karman Institute for Fluid Dynamics, Belgium.

D. J. Mavriplis (2002), 'An assessment of linear versus nonlinear multigrid methods for unstructured meshes', *J. Comput. Phys.* **175**, 302–325.

A. Meister (1994), Ein Beitrag zum DLR-τ-Code: Ein explizites und implizites Finite-Volume Verfahren zur Berechnung instationärer Strömungen auf unstructurierten Gittern. DLR Internal Report IB 223-94 A 36, Institute for Fluid Mechanics, DLR Göttingen.

A. Meister (1998), 'Comparison of different Krylov subspace methods embedded in an implicit finite volume scheme for the computation of viscous and inviscid flow fields on unstructured grids', *J. Comput. Phys.* **140**, 311–345.

A. Meister and M. Oevermann (1996), Computation of laminar and turbulent flow fields on unstructured grids with a finite volume scheme, in *Proc. 2nd Seminar on Euler and Navier–Stokes Equations, Prague*, pp. 61–62.

A. Meister and C. Vömel (2001), 'Efficient preconditioning of linear systems arising from the discretization of hyperbolic conservation laws', *Adv. Comput. Math.* **14**, 49–73.

C. A. Micchelli and T. J. Rivlin (1977), A survey of optimal recovery, in *Optimal Estimation in Approximation Theory* (C. A. Micchelli and T. J. Rivlin, eds), Plenum, pp. 1–54.

M. S. Moch (1980), 'Systems of conservation laws of mixed type', *J. Diff. Equations* **37**, 70–88.

C. B. Morrey (1960), 'Multiple integral problems in the calculus of variations and related topics', *Ann. Scuola Norm. Pisa* (III) **14**, 1–61.

K. W. Morton (1996), *Numerical Solution of Convection-Diffusion Problems*, Chapman and Hall.

K. W. Morton (1998), 'On the analysis of finite volume methods for evolutionary problems', *SIAM J. Numer. Anal.* **35**, 2195–2222.

K. W. Morton (2001), 'Discretization of unsteady hyperbolic conservation laws', *SIAM J. Numer. Anal.* **39**, 1556–1597.

K. W. Morton and D. F. Mayers (2005), *Numerical Solution of Partial Differential Equations*, 2nd edn, Cambridge University Press.

K. W. Morton and M. F. Paisley (1989), 'A finite volume scheme with shock fitting for the steady Euler equations', *J. Comput. Phys.* **80**, 168–203.

K. W. Morton and M. A. Rudgyard (1988), Shock recovery and the cell vertex scheme for the steady Euler equations, in *11th International Conference on Numerical Methods in Fluid Dynamics* (D. L. Dwoyer, M. Y. Hussaini and R. G. Voigt, eds), Springer, pp. 424–428.

K. W. Morton and S. M. Stringer (1998), Artificial dissipation as a feedback control with application to the cell vertex method, in *Computational Fluid Dynamics Review 1998*, Vol. 1 (M. Hafez and K. Oshima, eds), World Scientific, pp. 262–279.

G. Mühlbach (1978), 'The general Neville–Aitken algorithm and some applications', *Numer. Math.* **31**, 97–110.

C. Müller (1957), *Grundprobleme der Mathematischen Theorie Elektromagnetischer Schwingungen*, Springer.

E. M. Murman and J. D. Cole (1971), 'Calculation of plane steady transonic flows', *AIAA Journal* **9**, 114–121.

R. N. Ni (1982), 'A multiple grid system for solving the Euler equations', *AIAA Journal* **20**, 1565–1571.

E. J. Nielsen, W. K. Anderson, R. W. Walters and D. E. Kayes (1995), Application of Newton–Krylov methodology to a three-dimensional Euler code, in *Proc. 12th IAAA CFD Conf.,* Paper 95-1733-CP.

O. Oleinik (1957), 'Discontinuous solutions of nonlinear differential equations', *Usp. Mat. Nauk.* (NS) **12**, 3–73.

S. Osher and F. Solomon (1982), 'Upwind difference schemes for hyperbolic systems of conservation laws', *Math. Comp.* **38**, 339–374.

S. Ostkamp (1997), 'Multidimensional characteristic Galerkin schemes and evolution operators for hyperbolic systems', *Math. Methods Appl. Sci.* **20**, 1111–1125.

O. Pironneau (1982), 'On the transport-diffusion algorithm and its application to the Navier–Stokes equations', *Numer. Math.* **38**, 309–332.

A. Preissmann (1961), Propagation des intumescences dans les canaux et rivières, in *1st Congr. de l'Assoc. Française de Calcul*, Association Française de Calcul, Grenoble, France, pp. 433–442.

A. Quarteroni and A. Valli (1994), *Numerical Approximation of Partial Differential Equations*, Springer, Heidelberg.

M. Ricciutto, A. Csik and H. Deconinck (2005), 'Residual distribution for general time-dependent conservation laws', *J. Comput. Phys.* **209**, 249–289.

R. D. Richtmyer and K. W. Morton (1967), *Difference Methods for Initial-Value Problems*, Interscience, New York.

A. W. Rizzi and M. Inouye (1973), 'Time split finite volume method for three dimensional blunt-body flows', *AAIA Journal* **11**, 1478–1485.

P. L. Roe (1981), 'Approximate Riemann solvers, parameter vectors and difference schemes', *J. Comput. Phys.* **43**, 357–372.

P. L. Roe (1982), Fluctuations and signals: A framework for numerical evolution problems, in *Numerical Methods for Fluid Dynamics* (K. W. Morton and M. J. Baines, eds), Academic Press, pp. 219–257.

P. L. Roe (2001), Chapter 6, Numerical Methods, in *Handbook of Shockwaves*, Vol. 1 (G. Ben-dor, O. Igra and T. Elperin, eds), Academic Press, pp. 788–876.

Y. Saad and M. H. Schultz (1986), 'A generalized minimal residual algorithm for solving nonsymmetric linear systems', *J. Sci. Statist. Comput.* **7**, 856–869.

C. Shu and S. Osher (1988), 'Efficient implementation of Essentially Non-Oscillatory shock-capturing schemes', *J. Comput. Phys.* **77**, 439–471.

J. Smoller (1983), *Shock Waves and Reaction-Diffusion Equations*, Springer, New York.

G. A. Sod (1978), 'A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws', *J. Comput. Phys.* **27**, 1–31.

T. Sonar (1993*a*), 'On the design of an upwind scheme for compressible flow on general triangulations', *Numer. Algorithms* **4**, 135–149.

T. Sonar (1993*b*), 'Strong and weak norm refinement indicators based on the finite element residual for compressible flow computation', *IMPACT of Comp. in Science and Eng.* **5**, 111–127.

T. Sonar (1996), 'Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws', *IMA J. Numer. Anal.* **16**, 549–581.

T. Sonar (2002), Chapter 3, 'Methods on unstructured grids, WENO and ENO recovery techniques', in *Hyperbolic Partial Differential Equations: Theory, Numerics and Applications* (A. Meister and J. Struckmeier, eds), Vieweg, pp. 115–232.

T. Sonar and E. Süli (1998), 'A dual graph-norm refinement indicator for finite volume approximations of the Euler equations', *Numer. Math.* **78**, 619–658.

T. Sonar, V. Hannemann and D. Hempel (1994), 'Dynamic adaptivity and residual control in unsteady compressible flow computation', *Math. Comput. Modelling* **20**, 201–213.

A. Staniforth and J. Côté (1991), 'Semi-Lagrangian integration schemes and their application to environmental flows', *Mon. Weather Rev.* **119**, 2206–2223.

J. L. Steger and R. F. Warming (1981), 'Flux vector splitting of the inviscid gas-dynamic equations with applications to finite difference methods', *J. Comput. Phys.* **40**, 263–293.

E. Süli and P. Houston (1997), Finite element methods for hyperbolic problems: *a posteriori* error analysis and adaptivity, in *The State of the Art in Numerical Analysis* (I. S. Duff and G. A. Watson, eds), Clarendon Press, Oxford, pp. 441–471.

J. L. Synge (1957), *The Hypercircle in Mathematical Physics*, Cambridge University Press.

R. S. Varga (1962), *Matrix Iterative Analysis*, Prentice-Hall International, London.

H. A. van der Vorst (2003), *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press.

Y. Wada and M.-S. Liou (1994), A flux splitting scheme with high resolution and robustness for discontinuities. AIAA Paper 94-0083.

P. Wesseling (1992), *An Introduction to Multigrid Methods*, Wiley, Chichester.

K. H. Winters, J. Rae, C. P. Jackson and K. A. Cliffe (1981), 'The finite element method for laminar flow with chemical reaction', *Internat. J. Numer. Methods Engrg.* **17**, 239–253.

P. Woodward and P. Colella (1984), 'The numerical solution of two-dimensional fluid flow with strong shocks', *J. Comput. Phys.* **54**, 115–173.