MULTIDIMENSIONAL UPWINDING WITH GRID ADAPTATION

M.J. BAINES AND M.E. HUBBARD
ICFD, The University of Reading,
Department of Mathematics,
Whitknights, P.O. Box 220,
Reading, RG6 6AF.

1. Introduction

Over the past ten years multidimensional upwinding techniques have been developed with the intention of superseding traditional conservative upwind finite volume methods which rely on the solution of one-dimensional Riemann problems. The new methods attempt a more genuinely multidimensional approach to the solution of the Euler equations by considering a piecewise linear continuous representation of the flow with the data stored at the nodes of the grid. The schemes are then constructed from three separate stages: the decomposition of the system of equations into simple (usually scalar) components, the construction of a consistent, conservative discrete form of the equations and the subsequent solution of the decomposed system using scalar fluctuation distribution schemes. A detailed description of each of these stages can be found in [1, 2, 3].

As we shall show, the quality of the solution of a system of differential equations can be improved by means of grid adaptation. For example, multidimensional upwind schemes will capture shocks within two or three cells when they are aligned with the grid [3] and adaptation can be used to take advantage of this. On unstructured grids this can be accomplished by refinement, which reduces the size of the cells, and by edge swapping, which realigns the grid. However, both selective refinement and edge alignment can, to a large extent, be achieved by a third option, grid movement, which has the added advantage of avoiding the expensive process of changing the number of nodes or the connectivity of the grid.

In this paper one of the most recent and successful of the multidimensional upwind algorithms [1, 3] is described. Following this, a very simple and cheap algorithm for moving nodes is presented, which improves the accuracy of two-dimensional steady state solutions of the Euler equations on unstructured triangular grids.

2. Multidimensional upwinding

2.1. DECOMPOSITION OF THE EULER EQUATIONS

The two-dimensional Euler equations in conservative form are written

$$\underline{\mathbf{U}}_t + \underline{\mathbf{F}}_x + \underline{\mathbf{G}}_y = \underline{\mathbf{0}} \,, \tag{1}$$

where

$$\underline{\mathbf{U}} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix} \,, \quad \underline{\mathbf{F}} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(p+e) \end{pmatrix} \,, \quad \underline{\mathbf{G}} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(p+e) \end{pmatrix} \,, \tag{2}$$

are the vectors of conserved variables and the corresponding conservative fluxes, respectively, in which ρ is density, u and v are the x- and y-velocities, p is pressure, and e is total energy, related to the other variables by an equation of state which, for a perfect gas, is

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \,. \tag{3}$$

The decomposition stage of the algorithm dictates how the flux balance within each triangle of the grid, namely

$$\underline{\mathbf{\Phi}}_U = -\int\int_{\triangle} \left(\underline{\mathbf{F}}_x + \underline{\mathbf{G}}_y\right) \, dx\, dy \,, \tag{4}$$

is divided into simpler components. In this paper each of these components will be scalar (although in some cases transforming the equations results in a simple elliptic subsystem which could be considered without further decomposition). The decomposed flux balance takes the form of a sum of scalar contributions,

$$\underline{\mathbf{\Phi}}_U = \sum_{k=1}^{N} \phi^k \underline{\mathbf{r}}_U^k \,, \tag{5}$$

where

$$\phi^k = -S_\triangle \left(\vec{\lambda}^k \cdot \vec{\nabla} W^k + q^k\right) \tag{6}$$

is the general form of the fluctuation due to the k^{th} component of the decomposition, in which $S_\triangle$ is the area of the cell. $\underline{\mathbf{r}}_U^k$ is the vector which maps this flux balance contribution back to the conservative variables and N is the number of components (or waves) in the model.

Unlike in one dimension where a unique decomposition is available, many different wave models have been proposed for the two-dimensional Euler equations. These can be divided into groups: some decompose the flux balance into contributions corresponding to simple wave solutions of the Euler equations [4, 5], while others use a similarity transformation of the system of equations into 'characteristic' variables [6]. However, the most successful models which have been produced are based on the decomposition of a preconditioned form of the Euler equations [3, 7, 8].

These 'preconditioned' decompositions are derived by considering the system (1) in the streamwise variables, ξ and η, and in terms of the symmetrising variables $\underline{\mathbf{Q}}$, defined by

$$\partial\underline{\mathbf{Q}} = \begin{pmatrix} \frac{\partial p}{\rho a} \\ \partial q \\ q\partial\theta \\ \partial p - a^2\partial\rho \end{pmatrix} , \tag{7}$$

where a is the local speed of sound, $q = \sqrt{u^2+v^2}$ is the flow speed and $\theta = \tan^{-1}\left(\frac{v}{u}\right)$ is the direction of the flow. When the symmetrised equations are preconditioned by the matrix $\mathbf{P}$ (see below) they can be written

$$\underline{\mathbf{Q}}_t + \mathbf{P}\left(\mathbf{A}^s_Q\underline{\mathbf{Q}}_\xi + \mathbf{B}^s_Q\underline{\mathbf{Q}}_\eta\right) = \underline{\mathbf{0}} , \tag{8}$$

where the new Jacobians are the symmetric matrices,

$$\mathbf{A}^s_Q = \begin{pmatrix} q & a & 0 & 0 \\ a & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & q \end{pmatrix} , \qquad \mathbf{B}^s_Q = \begin{pmatrix} 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \tag{9}$$

For the purposes of this work, the preconditioner $\mathbf{P}$ is chosen to be that of Deconinck and Paillère [7], a generalisation of the van Leer-Lee-Roe matrix [9], given by

$$\mathbf{P} = \frac{1}{q}\begin{pmatrix} \frac{\chi}{\beta_\epsilon^2}M^2 & -\frac{\chi}{\beta_\epsilon^2}M & 0 & 0 \\ -\frac{\chi}{\beta_\epsilon^2}M & \frac{\chi}{\beta_\epsilon^2}+1 & 0 & 0 \\ 0 & 0 & \chi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \tag{10}$$

in which β_ϵ and χ are

$$\beta_\epsilon = \sqrt{\max(\epsilon^2, |M^2-1|)} , \qquad \chi = \frac{\beta_\epsilon}{\max(M,1)} . \tag{11}$$

The singularity which would otherwise occur at the sonic line is avoided by choosing the constant $\epsilon > 0$ (typically, $\epsilon = 0.05$).

The preconditioned system (8) can now be completely (in the case of supersonic flow) or partially decoupled by transforming it into a set of characteristic equations, becoming

$$\underline{\mathbf{W}}_t + \mathbf{A}^s_W \underline{\mathbf{W}}_\xi + \mathbf{B}^s_W \underline{\mathbf{W}}_\eta = \underline{\mathbf{0}} \ , \tag{12}$$

where $\underline{\mathbf{W}}$ is given by

$$\partial \underline{\mathbf{W}} = \begin{pmatrix} \beta_\epsilon \frac{\partial p}{\rho a} + Mq\partial\theta \\ \beta_\epsilon \frac{\partial p}{\rho a} - Mq\partial\theta \\ \frac{\partial p}{\rho a} + M\partial q \\ \partial p - a^2 \partial \rho \end{pmatrix} . \tag{13}$$

The flux Jacobians resulting from this set of variables now have the very simple form

$$\mathbf{A}^s_W = \begin{pmatrix} \chi\nu^+ & \chi\nu^- & 0 & 0 \\ \chi\nu^- & \chi\nu^+ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \quad \mathbf{B}^s_W = \begin{pmatrix} \frac{\chi}{\beta_\epsilon} & 0 & 0 & 0 \\ 0 & -\frac{\chi}{\beta_\epsilon} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} , \tag{14}$$

where

$$\nu^+ = \frac{M^2 - 1 + \beta_\epsilon^2}{2\beta_\epsilon^2} , \qquad \nu^- = \frac{M^2 - 1 - \beta_\epsilon^2}{2\beta_\epsilon^2} . \tag{15}$$

Thus, the system can be written as four scalar equations of the form

$$W^k_t + \vec{\lambda}^k_s \cdot \vec{\nabla}_s W^k + q^k_s = 0 \ , \qquad k = 1, 2, 3, 4, \tag{16}$$

where the advection velocities in streamwise coordinates are given by

$$\vec{\lambda}^1_s = \begin{pmatrix} \chi \\ \frac{\chi}{\beta_\epsilon} \end{pmatrix}_s , \quad \vec{\lambda}^2_s = \begin{pmatrix} \chi \\ -\frac{\chi}{\beta_\epsilon} \end{pmatrix}_s , \quad \vec{\lambda}^3_s = \vec{\lambda}^4_s = \begin{pmatrix} 1 \\ 0 \end{pmatrix}_s , \tag{17}$$

and the coupling terms are

$$q^1_s = \chi\nu^- W^2_\xi \ , \quad q^2_s = \chi\nu^- W^1_\eta \ , \quad q^3_s = q^4_s = 0 \ . \tag{18}$$

These indicate that the decomposition is optimal since the third and fourth equations are always decoupled from the rest of the system, indicating the invariance of entropy and enthalpy along the streamlines, while for supersonic flow $\nu^- = 0$ and the system is completely decoupled. In subsonic flow it is possible to consider the first and second equations as an elliptic subsystem which can be decomposed no further [1] but for the purposes of this work they are considered independently, as follows.

Each scalar component is now treated separately. The fluctuations ϕ^k are cell-based quantities which can be distributed to the nodes of the grid to bring the system closer to equilibrium. The resulting contributions to the nodes can be transformed into updates to the conservative variables using the matrix

$$\mathbf{R}_U = \frac{\partial \underline{\mathbf{U}}}{\partial \underline{\mathbf{Q}}} \mathbf{P}^{-1} \frac{\partial \underline{\mathbf{Q}}}{\partial \underline{\mathbf{W}}} \,, \tag{19}$$

the columns of which are the vectors which provide the mapping in (6) of the characteristic system back on to the conservative system. $\mathbf{R}_U$ is non-singular, so its columns are linearly independent and this four-component decomposition is linearity preserving, *i.e.* a zero flux balance implies a zero contribution to each node, and should not destroy the higher order accuracy of the scalar distribution schemes. Note that, though the decomposition is itself continuous, the distribution is not since the acoustic subsystem will be treated differently for supersonic and subsonic flows.

2.2. CONSERVATIVE LINEARISATION

The above decomposition is based on an analysis of a quasilinear form of the Euler equations,

$$\underline{\mathbf{U}}_t + \mathbf{A}_U \underline{\mathbf{U}}_x + \mathbf{B}_U \underline{\mathbf{U}}_y = \underline{\mathbf{0}} \,, \tag{20}$$

where $\mathbf{A}_U = \frac{\partial \underline{\mathbf{F}}}{\partial \underline{\mathbf{U}}}$ and $\mathbf{B}_U = \frac{\partial \underline{\mathbf{G}}}{\partial \underline{\mathbf{U}}}$ are the conservative flux Jacobian matrices. In order to guarantee that any discretisation involving such a decomposition will give rise to a conservative scheme, a conservative linearisation of this system is necessary.

An appropriate and very neat linearisation may be obtained [10] by assuming that Roe's parameter vector variables,

$$\underline{\mathbf{Z}} = \sqrt{\rho} \begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix} \,, \tag{21}$$

where $H = \frac{e+p}{\rho}$ is the total enthalpy, vary linearly within each cell. Under this assumption, each of the Jacobian matrices, $\frac{\partial \underline{\mathbf{U}}}{\partial \underline{\mathbf{Z}}}$, $\frac{\partial \underline{\mathbf{F}}}{\partial \underline{\mathbf{Z}}}$ and $\frac{\partial \underline{\mathbf{G}}}{\partial \underline{\mathbf{Z}}}$, depends linearly on the components of $\underline{\mathbf{Z}}$ and, because of this and the linear variation of $\underline{\mathbf{Z}}$, (4) can be integrated exactly using a one point quadrature, leading to a discrete form of the flux balance which is equal to the exact flux balance.

This ensures that, as long as the whole of each discrete flux balance is distributed to the nodes at each time-step, the sum of these contributions over the whole grid reduces to boundary contributions and the scheme is conservative.

The result is a consistent approximation to the flux balance of the form

$$\begin{aligned} \underline{\boldsymbol{\Phi}}_U &= -S_\triangle \left(\widehat{\underline{\mathbf{F}}_x} + \widehat{\underline{\mathbf{G}}_y} \right) \\ &= -S_\triangle \left(\mathbf{A}_U(\widehat{\underline{\mathbf{Z}}}) \widehat{\underline{\mathbf{U}}_x} + \mathbf{B}_U(\widehat{\underline{\mathbf{Z}}}) \widehat{\underline{\mathbf{U}}_y} \right) \ , \end{aligned} \tag{22}$$

where the Jacobians are evaluated at the Roe-average state,

$$\widehat{\underline{\mathbf{Z}}} = \frac{\underline{\mathbf{Z}}_1 + \underline{\mathbf{Z}}_2 + \underline{\mathbf{Z}}_3}{3} \ , \tag{23}$$

the mean of the values of $\underline{\mathbf{Z}}$ at the vertices of the cell, and the approximation to the gradient of the conservative variables, $\widehat{\vec{\nabla}\underline{\mathbf{U}}}$, is evaluated consistently from the discrete gradient of $\underline{\mathbf{Z}}$,

$$\widehat{\vec{\nabla}\underline{\mathbf{Z}}} = \frac{1}{2S_\triangle} \sum_{j=1}^{3} \underline{\mathbf{Z}}_j \vec{n}_j \ , \tag{24}$$

where $\vec{n}_j$ is the inward pointing normal to the edge opposite vertex j, scaled by the length of the edge. Importantly, the form of the linearisation implies that the analysis of the continuous system in the previous section will also hold at the discrete level without alteration, provided that all variables are evaluated at the average state $\widehat{\underline{\mathbf{Z}}}$ and all discrete gradients are calculated consistently from $\widehat{\vec{\nabla}\underline{\mathbf{Z}}}$.

2.3. SCALAR FLUCTUATION DISTRIBUTION SCHEMES

Once the wave model has been used to decompose the system of equations into scalar components (5), the behaviour of these components (if the source terms are ignored for the moment) can be modelled by the scalar advection equation,

$$u_t + f_x + g_y = 0 \qquad \text{or} \qquad u_t + \vec{\lambda} \cdot \vec{\nabla} u = 0 \ , \tag{25}$$

where $\vec{\lambda} = (\frac{\partial f}{\partial u}, \frac{\partial g}{\partial u})^T$ defines the velocity of the advected variable u.

A scheme can be constructed for the solution of this equation by calculating the fluctuation,

$$\begin{aligned} \phi &= -\int\!\!\int_\triangle \vec{\lambda} \cdot \vec{\nabla} u \, dx \, dy \\ &= -S_\triangle \widehat{\vec{\lambda}} \cdot \vec{\nabla} u \ , \end{aligned} \tag{26}$$

within each cell and then distributing it to the nodes of the grid, giving rise to a form of cell-vertex scheme. The integration, which can be done exactly because u is assumed to be linear within the cell, introduces the factor of $S_\triangle$, the area of the triangle, and a cell-averaged wave speed,

$$\widehat{\vec{\lambda}} = \frac{1}{S_\triangle} \int\!\!\int_\triangle \vec{\lambda}\, dx\, dy\ . \tag{27}$$

For simplicity and compactness, a cell is allowed to contribute its fluctuation only to its own vertices. Since summing the fluctuations over the whole domain reduces to a sum of boundary contributions, a conservative scheme is assured as long as the whole of each fluctuation is distributed.

If explicit forward Euler time-stepping is used, this leads to a scheme of the form

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{S_i} \sum_{\cup\triangle_i} \alpha_i^j \phi_j\ , \tag{28}$$

where S_i is the area of the median dual cell for node i (one third of the total area of the triangles with a vertex at i), α_i^j is the distribution coefficient which indicates the proportion of the fluctuation ϕ_j to be sent from cell j to node i, and $\cup\triangle_i$ represents the set of cells adjacent to node i. Since each fluctuation is a linear function of the data, the scheme is of the form

$$u_i^{n+1} = \sum_k c_{ik} u_k^n\ . \tag{29}$$

If the coefficients c_{ik} are allowed to depend on the data, the scheme becomes nonlinear and can be designed to satisfy the following four criteria:

- Upwind - the fluctuation within a cell is only sent to the downstream vertices of that cell *i.e.* vertices opposite inflow edges for which $\vec{\lambda}\cdot\vec{n} > 0$, where $\vec{n}$ is the inward pointing normal to the edge.
- Positive - the coefficients c_{ik} are positive, so the scheme cannot produce new extrema in the solution at the new time-step, spurious oscillations will not appear in the solution and the scheme is stable for an appropriate time-step restriction.
- Linearity preservation - no update is sent to the nodes when a cell fluctuation is zero, so the scheme is second order accurate at the steady state on a regular mesh with a uniform choice of diagonals.
- Continuity - the contributions to the nodes, $\alpha_i^j \phi_j$, depend continuously on the data, avoiding limit cycling as convergence is approached.

A linear scheme cannot satisfy both the positivity and the linearity preservation properties simultaneously.

In the search for a scheme which satisfies all of the above properties it is initially advantageous to consider a linear, positive, upwind scheme, the N scheme. By the above definition of upwind, any triangle with only one downstream vertex will send the whole of its fluctuation to that node. Where it differs from other upwind schemes is in its treatment of cells with two inflow sides. Taking a single cell in isolation, with vertices i, j, k, of which i and j are downstream nodes, the N scheme can be written

$$\begin{aligned} S_i u_i^{n+1} &= S_i u_i^n - \Delta t k_i (u_i^n - u_k^n) \\ S_j u_j^{n+1} &= S_j u_j^n - \Delta t k_j (u_j^n - u_k^n) \\ S_k u_k^{n+1} &= S_k u_k^n \;, \end{aligned} \tag{30}$$

where $k_{i,j} = \frac{1}{2}\vec{\lambda}\cdot\vec{n}_{i,j}$ and contributions from other triangles are suppressed. By considering the complete nodal update (28), this scheme can be shown to be positive for a restriction on the time-step at a node i, given by

$$\Delta t \leq \frac{S_i}{\sum_{\cup\triangle_i} \max(0, k_i^j)} \;. \tag{31}$$

A linearity preserving scheme (which also retains the upwind and continuity properties) can be obtained from a positive upwind scheme [11] such as the N scheme by replacing the contributions, ϕ_i and ϕ_j, to the downstream nodes in the two-target case by 'limited' contributions,

$$\begin{aligned} \phi_i^* &= \phi_i - L(\phi_i, -\phi_j) \\ \phi_j^* &= \phi_j - L(\phi_j, -\phi_i) \;. \end{aligned} \tag{32}$$

where, in the case of the N scheme,

$$\phi_i = -k_i (u_i^n - u_k^n) \;, \qquad \phi_j = -k_j (u_j^n - u_k^n) \;. \tag{33}$$

$L(x, y)$ is any member of the family of symmetric limiter functions, although the minmod limiter,

$$L(x, y) = \frac{1}{2}(1 + \mathrm{sgn}(xy))\,\frac{1}{2}(\mathrm{sgn}(x) + \mathrm{sgn}(y))\,\min(|x|, |y|) \;, \tag{34}$$

is the only one for which the 'limited' scheme remains positive. The resulting scheme, which is equivalent to the Positive Streamwise Invariant (PSI) scheme [2], satisfies all of the desired properties and will be used here to distribute all fluctuations without any associated source term.

When the flow is subsonic, not all of the components of the decomposition yield homogeneous advection equations and these are dealt with differently. In this case the two characteristic equations making up the acoustic subsystem are still solved with a scalar scheme, but one different to the PSI scheme described above.

The new scheme is derived by analogy with the finite element method and from consideration of a weak form of the linear advection equation, given by

$$\int\!\!\int_{\Omega} \tilde{\omega}_i u_t \; dx\, dy = -\int\!\!\int_{\cup\triangle_i} \tilde{\omega}_i \vec{\lambda} \cdot \vec{\nabla} u \; dx\, dy \;, \tag{35}$$

where the approximation to u in terms of the linear basis functions ω_i is

$$u(x,y) = \sum_{i=1}^{N_n} u_i \, \omega_i(x,y) \;. \tag{36}$$

Test functions $\tilde{\omega}_i$ are chosen [2] which add both linear and nonlinear dissipation terms to the standard linear 'tent' function, used as the basis function. When combined with mass-lumping and forward Euler time-stepping on the left hand side of (35), the result is a fluctuation distribution scheme for which the coefficients are given by

$$\alpha_i^j = \frac{1}{3} + \tau_1 \frac{k_i}{S_j} + \tau_2 \frac{(k_i)_{\|}}{S_j} \;, \tag{37}$$

where

$$(k_i)_{\|} = \frac{1}{2} \vec{\lambda}_{\|} \cdot \vec{n}_i \;, \tag{38}$$

and

$$\tau_1 = C_1 \frac{h}{|\vec{\lambda}|} \;, \qquad \tau_2 = C_2 \frac{h}{|\vec{\lambda}_{\|}|} \;. \tag{39}$$

C_1 and C_2 are both constants set to 0.5 [12], h is some measure of the size of the discretisation, taken to be the length of the longest edge of the triangle, and $\vec{\lambda}_{\|}$ is the gradient dependent advection velocity, the projection of the advection velocity on to the solution gradient. This describes a nonlinear, mass-lumped, Streamline Upwind Petrov-Galerkin (SUPG) scheme, formulated in terms of fluctuation distributions. It is linearity preserving and continuous, but not generally positive nor, by the earlier definition, truly upwind.

3. Grid adaptation

The adaptation algorithm presented in this paper is a very simple form of node movement. It takes the form of an iteration where, at each step, nodes are moved to a weighted average of the positions of the centroids of the neighbouring triangles [13, 14]. The new nodal position can thus be written in terms of the old positions as

$$\vec{x}_i^{\,n+1} = \frac{\sum_{\cup\triangle_i} w_j \vec{x}_j^{\,n}}{\sum_{\cup\triangle_i} w_j} \;, \tag{40}$$

where the $\vec{x}_j$ are the positions of the centroids, w_j are the cell weights and the sums are over the cells adjacent to node i. The iteration (40) with constant, non-negative weights can be shown to converge. Moreover, provided that the weights themselves converge, convergence of the iteration can be shown for variable weights.

In the solution of the Euler equations, the weights w have been chosen to depend on local approximations to the first and second derivatives of the density of the flow, in the form

$$w = \frac{S_\triangle}{S_O}\left(1 + \alpha|\vec{\nabla}\rho|^2 + \beta(\vec{\nabla}^2\rho)^2\right)^{\frac{1}{2}} \ , \tag{41}$$

where α and β are arbitrary parameters, $S_\triangle$ is the current area of the triangle and S_O is the original area of that triangle. For the linear advection equation ρ is replaced by u, the advected variable. The choice of $\alpha = 1$ and $\beta = 0$ in (41) gives a simple generalisation of the weights which lead to arc length equidistribution in one dimension (assuming that the initial grid is equispaced so that S_O is constant throughout the domain).

Although there is no corresponding genuinely two-dimensional equidistribution property, the algorithm will still tend to move nodes towards regions where the weights are high. In the above case this means regions of high first and/or second derivatives, such as those found at shocks, but the weights can be modified so that nodes are attracted towards any detectable feature of the flow. Also, since the weights depend on derivatives of the flow, the degree of attraction to these features can be varied by scaling the grid - the smaller the grid size, the stronger the effect of the adaptation. In the present application, where the weights depend on a flow which will ultimately be steady, the gradients, and hence the w_i move towards steady state values, and at the limit the grid can be interpreted as having a local equidistribution property in the direction of the normal to ρ. The algorithm can also be easily generalised to three dimensions.

In one dimension mesh tangling can be avoided by ensuring that the chosen weights are always positive. In higher dimensions, though, particularly on the highly distorted grids which become common once the mesh is allowed to move, tangling occurs quite readily. Even with positive weights in (40), it is possible for a node at the vertex of a triangle to be overtaken by the opposite edge of that triangle, thus causing the cell to 'flip' and acquire a negative area.

This can be avoided by artificially limiting the distance which a node can move. A simple but rather restrictive limit is

$$(\Delta x_i)_{\max} = \min_{\cup\triangle_i}\left(\frac{S_j}{\max_{k=1,3} l_{jk}}\right) \ , \tag{42}$$

where S_j is the area of cell j and l_{jk} is the length of edge k of cell j. This expression is equivalent to half the smallest height of the surrounding triangles. A second restriction is also imposed which places a lower limit on the radius of the inscribed circle of each cell. This avoids extremely distorted meshes and the possibility of a prohibitively small limit on the time-step.

Using this strategy, a displacement can be found for all nodes, including boundary nodes, although the latter must be projected back on to the nearest point on the boundary and ‘corner’ nodes forced to remain fixed.

Once all the displacements have been found, the nodal positions are updated in a block. The solution is then obtained on the new grid using linear interpolation of the solution on the previous grid.

4. Solution strategy

The method by which node movement is combined with multidimensional upwinding to obtain steady state solutions to the two-dimensional Euler equations can be expressed in three stages:

1) Run the time-stepping algorithm on an initial, fixed grid until the solution appears steady (but long before convergence is achieved).
2) Run the time-stepping interspersed with the grid movement until the grid has adapted to the steady solution. In this work, each time-step is alternated with a single node movement iteration.
3) Fix the grid and run the time-stepping algorithm to convergence using the solution from step 2) as initial conditions.

The grid movement in step 2) can be initiated when the RMS of the residual over the grid drops below a certain level (typically a drop of 2 or 3 orders of magnitude from the initial residual), in effect when the flow has stopped changing.

It may well be possible that the combination of time-stepping and grid movement in stage 2) would lead to a converged solution if allowed to run indefinitely. However, it would be impractical to attempt this because the convergence of the overall scheme, depending as it does on two separate iterations, would be prohibitively slow. Also, this stage of the method is not, as it stands, conservative due to the interpolation step of the grid movement. However, since steady state solutions are sought, the grid can be frozen after a fixed number of time-steps (typically 500 for the Euler equations) after which the solution strategy returns solely to the conservative time-stepping scheme. Local time-stepping has been used throughout to accelerate convergence, particularly on the more distorted meshes.

5. Results

Adapted and unadapted steady state solutions will be presented here of both the scalar advection equation and the Euler equations. The first test case used is that of clockwise circular advection, $\vec{\lambda} = (y, -x)^T$ of a square wave profile within the domain $(x, y) \in [-1, 1] \times [0, 1]$. The mesh used is an isotropic triangulation alternating the direction of diagonals inserted into a regular quadrilateral grid with 65×33 nodes and of the form shown in Figure 1.

Figure 1. A section of the initial isotropic grid.

The boundary conditions for this test case are

$$\begin{aligned}
u(x,0) &= 1 \quad \text{for} \quad -0.65 \le x \le -0.35 \\
u(x,0) &= 0 \quad \text{for} \quad -1.0 \le x < -0.65, \quad -0.35 < x \le 0.0 \\
u(x,1) &= 0 \quad \text{for} \quad 0.0 \le x \le 1.0 \\
u(0,y) &= 0 \quad \text{for} \quad 0.0 \le y \le 1.0\ .
\end{aligned}$$

On the rest of the boundary, where the flow is out of the domain, the solution is given the value predicted by the updates from the interior and initially the solution is set to zero in the whole of the domain's interior.

The steady state solution of this problem obtained using the PSI scheme (CFL = 0.8) is shown at the top of Figure 2. There are no spurious oscillations since the scheme is positive, and the linearity preservation property ensures that the discontinuities are captured reasonably sharply although a certain amount of numerical diffusion is unavoidable. The rest of Figure 2 shows both the adapted solution and the grid on which it has been obtained. As suggested in the previous section, the algorithm was run initially on the unadapted grid (for 200 iterations), then the grid movement was interleaved with the time-stepping for a further 200 iterations, after which the grid is fixed again and a converged solution obtained on the new grid. The CFL number can be kept at 0.8 throughout this procedure and the convergence of the algorithm on the adapted grid is not significantly

Figure 2. Circular advection of a square profile: solution on initial grid (top), solution on adapted grid with $\alpha = 1.0$, $\beta = 0.01$ (middle); adapted grid (bottom).

Figure 3. Circular advection of a triangular profile: solution on initial grid (top), solution on adapted grid with $\alpha = 0.1$, $\beta = 1.0$ (middle); adapted grid (bottom).

Figure 4. Solution on the boundary $y = 0$ for the advection of the square wave (left) and the triangular wave (right). Solid lines indicate the adapted solution, dotted lines the unadapted solution.

slower than on the initial, regular grid. However, the solution has improved enormously in quality; the discontinuities are captured extremely sharply and their positions are mirrored by the clustering of nodes in the grid. The grid movement parameters in (41) were chosen to be $\alpha = 1.0$ and $\beta = 0.01$, so emphasis was placed on adapting to the first derivative.

Figure 3 shows the steady state solution on both unadapted and adapted grids for a second circular advection test case, this time involving a triangular profile. The initial and boundary conditions are the same as before except that

$$u(x, 0) = 1.0 - \frac{|x + 0.5|}{0.15} \qquad \text{for} \qquad -0.65 \leq x \leq -0.35\ . \tag{43}$$

This time $\alpha = 0.1$ and $\beta = 1.0$ were chosen to give a greater significance to the second derivative, but the improvement in the quality of the solution is again remarkable.

Figure 4 shows the profiles of the adapted (solid line) and unadapted (dotted line) solutions on the boundary $y = 0$ for both test cases. When $x \leq 0$ this is an inflow boundary so the solution is exact, or as near as the grid will allow. The solution at outflow ($x > 0$) shows the improvement obtained by adapting the grid, and in both cases the shape and height of the solution has been maintained through its rotation.

However, it is necessary to be careful when choosing α and β. If $\beta = 0.0$, *i.e.* no second derivative contribution, the triangular profile becomes highly distorted as it rotates around the origin, Figure 5, and has almost become a

Figure 5. Circular advection of a triangular profile: solution on adapted grid with $\alpha = 1.0$, $\beta = 0.0$ (top); adapted grid (bottom).

square wave at outflow. This is because the flow varies linearly in the regions of high first derivative and can be modelled accurately using a coarse mesh, but changes rapidly at the base and peak of the triangle where the second derivative is high but the gradient is not. In fact, any solution in which capturing maxima or minima is important requires some contribution to the weights from the second derivative. Experience has shown that taking $\alpha = 1.0$ and $\beta = 0.1$ is a reasonably safe first choice for the weight parameters.

The first test case used for the Euler equations is that of flow through a walled channel of unit height with a circular arc bump on the lower surface which is 4% of the height of the channel. The freestream Mach number is $M_\infty = 1.4$. The time-stepping algorithm used decomposes the equations using the 'preconditioned' model described earlier and the PSI scheme on

Figure 6. Inviscid flow through a channel of unit width with a 4% circular arc bump of unit length, $M_\infty = 1.4$: local Mach number contours of unadapted solution (top), adapted solution (middle); adapted grid (bottom).

Figure 7. Initial grid for NACA0012 aerofoil.

the scalar components in all situations except for the acoustic subsystem in subsonic flow which is treated as two scalar equations and solved using the SUPG scheme. As a result, the CFL number used for these calculations is only 0.2. The boundaries are treated using a very simple characteristic-type boundary condition.

Figure 6 shows the local Mach number contours of the steady state solution obtained on a fixed isotropic grid, similar to that used for the scalar advection test cases, with 2145 nodes and 4096 cells. The adaptive algorithm is then used, with 5000 iterations being completed on this initial grid, 500 more being interleaved with the grid movement ($\alpha = 1.0$, $\beta = 0.1$) and then running to convergence on the adapted grid. The new grid and the solution obtained on it are both shown at the bottom of Figure 6 and again the improvement is marked, particularly in the capturing of the shocks. More surprisingly, perhaps, the oscillations which occur behind the shock reflection on the upper wall due to the non-positivity of the scheme in this small region of subsonic flow, are smoothed out by the adaptation. It is only where the shocks interact in a more complicated manner and nodes 'lock' that the improvement is less significant. Node locking can occur for two reasons, either because they are constrained to remain close to fixed objects such as boundaries or because they have an equal desire to move in more than one direction, *e.g.* within the triangle of shocks just behind the bump, and in either case some form of mesh refinement is needed if the solution is to be further improved.

Finally, two sets of results are given for inviscid flow around a NACA0012 aerofoil for which the initial grid is shown in Figure 7. No vorticity correction is included in the treatment of the far-field boundary so it has been

Figure 8. Inviscid flow over a NACA0012 aerofoil, $M_\infty = 0.85$, $\alpha = 1.0°$: local Mach number contours of unadapted solution, $c_l = 0.367$, $c_d = 0.050$ (top left), adapted solution, $c_l = 0.360$, $c_d = 0.049$ (top right); adapted grid (bottom right); comparison of c_p on aerofoil surface, unadapted - dotted line, adapted - solid line (bottom left).

placed at 30 chords distance from the aerofoil to reduce its effect on the solution.

The two standard test cases presented are

$$\begin{aligned} i)\quad & M_\infty = 0.85, \alpha = 1.0°. \\ ii)\quad & M_\infty = 0.8, \alpha = 1.25°. \end{aligned}$$

Figures 8 and 9 show the local Mach number contours of the solutions obtained on the fixed and moved grids, together with the adapted grid and a graph comparing the pressure coefficient c_p on the surface of the aerofoil

Figure 9. Inviscid flow over a NACA0012 aerofoil, $M_\infty = 0.8$, $\alpha = 1.25°$: local Mach number contours of unadapted solution, $c_l = 0.339$, $c_d = 0.015$ (top left), adapted solution, $c_l = 0.346$, $c_d = 0.016$ (top right); adapted grid (bottom right); comparison of c_p on aerofoil surface, unadapted - dotted line, adapted - solid line (bottom left).

on each grid. The solution algorithm was precisely that used to solve the channel flow, and the results again show how the capturing of shocks can be improved by the intelligent use of grid movement. It can also be seen that the oscillations behind the shocks have been reduced by moving the grid. In both cases the lift coefficient c_l is right in the middle of the range of values obtained from reference solutions using other codes [15], while the drag coefficients c_d are slightly lower. These are changed slightly by the adaptation but it is impossible to tell whether the new values are more accurate. Even so, the shocks have a significant effect on the lift and drag so the improvement in their capturing and the reduction in the oscillations behind them implies that c_l and c_d are likely to be more accurate on the adapted grids. In all of the above test cases, multidimensional has proved to be very robust even on highly distorted grids and, typically, the improved results obtained by using the adaptive algorithm take only 50% longer to produce than those on the fixed, regular grid.

6. Conclusions

Multidimensional upwinding techniques have been used to produce accurate steady state solutions to both the linear advection equation and the Euler equations in two dimensions in triangular grids, and a very simple and cheap node movement algorithm has been used to greatly improve the quality of these solutions. The strategy has proved to be robust, and steady state solutions have been obtained on all but the most highly distorted grids for a very small increase in the overall cost.

Even so, for many flows node movement is not enough to accurately model every feature and some form of grid refinement is necessary to obtain highly accurate solutions. This is particularly true where node locking prevents nodes from moving in the desired direction, a problem which can also be alleviated somewhat by the use of edge swapping. The best choices for the values of α and β, and the actual variable used to monitor the adaptation, are also unclear, and may vary depending on the flow. Finally, no measure of grid quality has been used here and indeed it is not clear what should be used since the improved solutions are obtained on grids which are much poorer in quality by any of the conventional criteria.

7. Acknowledgements

The second author would like to acknowledge the financial support of DRA Farnborough during the completion of this work.

References

1. H.Paillère, E.van der Weide and H.Deconinck. Multidimensional upwind methods for inviscid and viscous compressible flows. In *VKI LS 1995-02, Computational Fluid Dynamics*, 1995.
2. H.Deconinck, R.Struijs, G.Bourgois and P.L.Roe. High resolution shock capturing cell vertex advection schemes for unstructured grids. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.
3. H.Paillère. Multidimensional upwind residual distribution schemes for the Euler and Navier-Stokes equations on unstructured grids. PhD Thesis, Université Libre de Bruxelles, 1995.
4. P.L.Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. In *J. Comp. Phys.*, 63:458-476, 1986.

5. M.A.Rudgyard. Multidimensional wave decompositions for the Euler equations. In *VKI LS 1993-04, Computational Fluid Dynamics*, 1993.
6. H.Deconinck, C.Hirsch and J.Peuteman. Characteristic decomposition methods for the multidimensional Euler equations. In *Lecture Notes in Physics*, 264:216-221, 1986.
7. H.Deconinck and H.Paillère. Multidimensional upwinding: preparing the future of computational methods for compressible flows? In *Proceedings of the 5th ICFD Conference*, 1995.
8. L.M.Mesaros and P.L.Roe. Multidimensional fluctuation splitting schemes based on decomposition methods. AIAA Paper 95-1699, 1995.
9. B.van Leer, W.-T.Lee and P.L.Roe. Characteristic time-stepping or local preconditioning of the Euler equations. AIAA Paper 91-1552-CP, 1991.
10. H.Deconinck, P.L.Roe and R.Struijs. A multi-dimensional generalization of Roe's flux difference splitter for the Euler equations. *Journal of Computers and Fluids*, 22:215-222, 1993.
11. D.Sidilkover and P.L.Roe. Unification of some advection schemes in two dimensions. ICASE Report, submitted to Math. Comp., 1995.
12. C.Johnson. Finite elements for flow problems. In *Unstructured grid methods for advection dominated flows*, AGARD-R-787, 1992.
13. G.Erlebacher and P.R.Eiseman. Adaptive triangular mesh generation. *AIAA Journal*, 25:1356-1364, 1987.
14. M.E.Hubbard and M.J.Baines. Multidimensional upwinding and grid adaptation. In *Proceedings of the 5th ICFD Conference*, 1995.
15. H.Viviand. AGARD Report, AGARD-AR-211, 1985.