Multidimensional upwinding and grid adaptation

M.E.Hubbard & M.J.Baines

ICFD, The University of Reading

1 Introduction

Over the past ten years, multidimensional upwinding techniques have been developed with the intention of superceding 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 components: a wave decomposition model, a conservative linearisation and a scalar fluctuation distribution scheme. A detailed description of these can be found in [1, 2] while some more recent developments are presented in these proceedings [3].

The quality of the solution of a system of differential equations can be improved by means of grid adaptation. For example, multidimensional upwind schemes can capture shocks within two or three cells when they are aligned with the grid and adaptation can be used to take advantage of this. On unstructured grids this can be accomplished by refinement, which increases the density of the nodes, 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 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 obtained using multidimensional upwinding methods on unstructured triangular grids.

2 Multidimensional upwinding

2.1 Wave decomposition models

A wave decomposition model is a device used to split a system of equations into scalar components (or waves). The two-dimensional Euler equations in conservative form are written

$$\underline{\mathbf{u}}_t + \underline{\mathbf{F}}_x + \underline{\mathbf{G}}_y = \underline{\mathbf{0}}, \tag{2.1}$$

where $\underline{\mathbf{u}}$ is the vector of conservative variables and $\underline{\mathbf{F}}$, $\underline{\mathbf{G}}$ are the corresponding flux vectors. The wave model dictates how the flux balance within each triangle

of the grid, namely

$$\Phi = -\int\int_{\triangle} (\underline{\mathbf{F}}_x + \underline{\mathbf{G}}_y)\, dx\, dy, \tag{2.2}$$

is divided into simpler components. Each 'wave' is the solution of a scalar advection equation, so the flux balance can be written as a sum of scalar components,

$$\Phi = \sum_{k=1}^{N} \phi^k \underline{\mathbf{r}}^k, \tag{2.3}$$

where ϕ^k is the fluctuation (or strength) of the k^{th} wave and $\underline{\mathbf{r}}^k$ is the vector which maps the contribution of this wave back to the conservative variables, while N is the number of active waves in the decomposition.

Unlike in one dimension where a unique decomposition is available, many different wave models have been proposed for the two-dimensional Euler equations [2]. These can be divided into groups: some decompose the flux balance into contributions corresponding to simple wave solutions of the Euler equations, while others use a transformation of the system of equations into 'characteristic' variables. More recently, wave models have been produced which are based on a preconditioned form of the Euler equations [3]. However, the results presented in this paper have been produced using Rudgyard's Mach angle splitting [4]. This is not ideal since it consists of more than four active waves. Therefore the property of linearity preservation is lost and the method is not truly second order accurate, but it has proved to be very robust on distorted grids.

2.2 Conservative linearisation

All of the above wave models are based on analysis of a quasilinear form of the Euler equations,

$$\underline{\mathbf{u}}_t + A(\underline{\mathbf{u}})\underline{\mathbf{u}}_x + B(\underline{\mathbf{u}})\underline{\mathbf{u}}_y = \underline{\mathbf{0}}, \tag{2.4}$$

where $A(\underline{\mathbf{u}})$ and $B(\underline{\mathbf{u}})$ are flux Jacobian matrices. In order to guarantee that any discretisation involving such a decomposition will give rise to a conservative scheme, a conservative linearisation is necessary.

An appropriate and very neat linearisation may be obtained [5] by assuming that Roe's parameter vector variables,

$$\underline{\mathbf{z}} = \rho^{1/2}(1, u, v, H)^T, \tag{2.5}$$

vary linearly within each cell.

2.3 Scalar fluctuation distribution schemes

The wave model decomposes the system of equations into scalar components, equation (2.3), the behaviour of which can be modelled by the scalar advection equation,

$$u_t + \vec{\nabla} \cdot \vec{f} = 0 \quad \textit{or} \quad u_t + \vec{\lambda} \cdot \vec{\nabla} u = 0, \tag{2.6}$$

where $\vec{\lambda} = \vec{f}_u$ defines the velocity of the advected variable u.

Fluctuation distribution schemes can be constructed for the solution of this equation. The fluctuation

$$\phi = -\int\int_{\triangle} \vec{\lambda} \cdot \vec{\nabla} u \, dx \, dy = -V_{\triangle} \overline{\vec{\lambda}} \cdot \vec{\nabla} u \tag{2.7}$$

is calculated within each cell and then distributed 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 a factor of $V_{\triangle}$, the area of the triangle, and a cell-averaged wave speed, $\overline{\vec{\lambda}}$. For simplicity and compactness, a cell is allowed to contribute its fluctuation only to its own vertices. Conservation is assured as long as the whole of the fluctuation is distributed.

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

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{V_i} \sum_{j=1}^{N} \alpha_{ij} \phi_j, \tag{2.8}$$

where V_i is the area of the median dual cell for node i, α_{ij} is the distribution coefficient which indicates the proportion of the fluctuation ϕ_j to be sent from cell j to node i, and the sum is over the cells adjacent to node i. Since each fluctuation is a linear function of the data, the scheme is of the form

$$u_i^{n+1} = \sum_k c_{ik} u_k^n. \tag{2.9}$$

If the coefficients c_{ik} are now 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.
- 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 at the steady state on a regular mesh with a uniform choice of diagonals.
- Continuity - the contributions to the nodes depend continuously on the data, avoiding limit cycling as convergence is approached.

In this paper the PSI scheme [1], which satisfies all the above criteria, has been used throughout.

3 Grid adaptation

The adaptation algorithm used in this work 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 [6, 7]. The new nodal position can thus be written as

$$\vec{x}_n^{new} = \frac{\sum_{i=1}^{N} w_i \vec{x}_i}{\sum_{i=1}^{N} w_i} \tag{3.1}$$

where the $\vec{x}_i$ are the positions of the centroids, w_i are the cell weights and i runs over the cells adjacent to node n.

The weights here are chosen to depend on local approximations to the first and second derivatives of the density of the flow,

$$w = V_\triangle \left(1 + \alpha |\vec{\nabla}\rho|^2 + \beta (\vec{\nabla}^2 \rho)^2\right)^{1/2}, \tag{3.2}$$

where α and β are arbitrary parameters and $V_\triangle$ is the area of the triangle. The choice of $\alpha = 1$ and $\beta = 0$ gives a simple generalisation of the weights which lead to arc length equidistribution in one dimension. Although there is no corresponding equidistribution property in two dimensions, 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. 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 3.1, 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

$$(\Delta x)_{max} = \min_i \left(\frac{A_i}{\max_{j=1,3} l_{ij}} \right) \tag{3.3}$$

where i indexes the cells surrounding the node, A_i is the area of cell i and l_{ij} is the length of edge j of cell i. 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 but 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), effectively when the flow has stopped changing.

It is unlikely that the combination of time-stepping and grid movement will lead to a converged solution if allowed to run indefinitely, and it would be impractical to attempt this. 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 (500 in this case) 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

Results are shown for the steady state solution to the Euler equations in a walled channel of unit height and length three with a circular arc bump on the lower surface which is 4% of the height of the channel. The flow is mainly supersonic with $M_\infty = 1.4$. Figure 1 shows the local Mach number contours of the converged solution obtained on a fixed isotropic grid, created by dividing the domain into 64×32 evenly spaced quadrilaterals and then inserting diagonals which alternate in direction from one cell to the next.

The adaptive algorithm is then used, with this as the initial grid, and the resulting solution and grid are shown in Figures 2 and 3. This adapted converged solution took only 30% more cpu time to produce than that on the fixed grid. The weight parameters were chosen to be $\alpha = 1.0$ and $\beta = 0.01$ so the emphasis has been placed on adapting to the gradient of the density rather than the second derivative.

Figure 1. Local Mach number contours on the fixed isotropic grid, with $M_{min} = 0.9467$, $M_{max} = 1.6719$

Figure 2. Local Mach number contours on the adapted grid, with $M_{min} = 0.8413$, $M_{max} = 1.6681$

There is a marked improvement in the capturing of all the shocks, and the solution just behind the shock reflection on the upper wall where the minimum of the local Mach number occurs is much better. This was not the case when the second derivative was missing from the weights. It is only where the shocks interact in a more complicated manner, just behind the bump, and nodes 'lock' that the improvement is less significant. It is obvious that some form of mesh refinement is needed to improve the solution further.

6 Conclusions

A very simple and cheap node movement algorithm has been used in conjunction with multidimensional upwind schemes to improve the accuracy of steady state solutions to the two-dimensional Euler equations. The strategy has proved to be robust, and converged solutions have been obtained on all but the most highly distorted grids for a very small increase in the overall cost.

Even so, for some flows node movement is not enough 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. 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.

References

[1] 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.

[2] H.Paillère, J-C.Carette and H.Deconinck. Multidimensional upwind and SUPG methods for the solution of the compressible flow equations on unstructured grids. In *VKI LS 1994-05, Computational Fluid Dynamics*, 1994.

[3] H.Deconinck and H.Paillère. Multidimensional upwinding: preparing the future of computational methods for compressible flows? In *Proceedings of the 5th ICFD Conference*, 1995.

[4] M.A.Rudgyard. Multidimensional wave decompositions for the Euler equations. In *VKI LS 1993-04, Computational Fluid Dynamics*, 1993.

[5] 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.

[6] G.Erlebacher and P.R.Eiseman. Adaptive triangular mesh generation. *AIAA Journal*, 25:1356-1364, 1987.

[7] M.E.Hubbard. Grid adaptation and multidimensional upwinding. NA Report 8/94, University of Reading, 1994.