INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 16, 1079–1098 (1993)

A SECOND-ORDER UPWIND FINITE-VOLUME METHOD FOR THE EULER SOLUTION ON UNSTRUCTURED TRIANGULAR MESHES

DARTZI PAN* AND JEN-CHIEH CHENG†

Institute of Aeronautics and Astronautics, National Cheng Kung University, Tainan, Taiwan 70101, ROC

SUMMARY

A scheme for the numerical solution of the two-dimensional (2D) Euler equations on unstructured triangular meshes has been developed. The basic first-order scheme is a cell-centred upwind finite-volume scheme utilizing Roe's approximate Riemann solver. To obtain second-order accuracy, a new gradient based on the weighted average of Barth and Jespersen's three-point support gradient model is used to reconstruct the cell interface values. Characteristic variables in the direction of local pressure gradient are used in the limiter to minimize the numerical oscillation around solution discontinuities. An Approximate LU (ALU) factorization scheme originally developed for structured grid methods is adopted for implicit time integration and shows good convergence characteristics in the test. To eliminate the data dependency which prohibits vectorization in the inversion process, a black–gray–white colouring and numbering technique on unstructured triangular meshes is developed for the ALU factorization scheme. This results in a high degree of vectorization of the final code. Numerical experiments on transonic Ringleb flow, transonic channel flow with circular bump, supersonic shock reflection flow and subsonic flow over multielement aerofoils are calculated to validate the methodology.

KEY WORDS Unstructured triangular mesh Upwind finite-volume method Euler Approximate LU factorization

1. INTRODUCTION

Algorithms for solving the Euler equations on structured grids are widespread in the literature. One of the difficulties encountered by the structured grid method is the generation of a suitable structured mesh over complex geometries. The multizone methodology which divides the domain into separate and simple zones is effective for two-dimensional (2D) problems, however, still requires an enormous grid generation effort for 3D complex geometries. Recently attention has shifted to the solution of the Euler equations on unstructured grids which, in principle, has no difficulty in treating the geometrical singularities and irregularities. This shift is largely due to impressive results reported by Jameson and Mavriplis,[1–3] who constructed cell-centred and cell-vertexed schemes based on continuous piecewise linear elements with added artificial viscosity. Desideri and Dervieux[4] and Rostand and Stoufflet[5] have also shown results of upwind vertex approximations using Osher's flux formulation. Barth and Jespersen[6] have developed a discontinuous piecewise linear reconstruction and, later, Barth and Frederickson[7] have developed

*Associate Professor.
†Graduate student.

0271–2091/93/121079–20$15.00

Received 1 June 1992
Revised 8 January 1993

a high-order (k-exact) reconstruction with high-order Gaussian quadratures for the flux evaluations. However, the k-exact reconstruction model is rather complicated due to the need of satisfying the conservation of the mean, and, furthermore, the rule of selecting the optimal set of cells to support the k-exact reconstruction is still unresolved. As for the time integration, Whitaker *et al.*[8] have compared many explicit and implicit time integration methods, including a direct LU decomposition method with the reverse Cuthill–Mcgee renumbering technique to reduce the bandwidth of the implicit operator. In general, implicit methods on unstructured grids are usually difficult to vectorize and inefficient on vector computers due to the high degree of data dependency. An efficient method for the inversion of implicit operator is still to be sought.

In the present paper, efforts have been made to improve both the accuracy and efficiency of the Euler solution algorithm on unstructured triangular grids. The basic first-order scheme is a cell-centred upwind scheme on triangular mesh utilizing Roe's approximate Riemann solver. To obtain second-order accuracy, the weighted average of Barth and Jespersen's three-point support linear gradient is used to reconstruct the Riemann states at the cell interface. Characteristic variables in the direction of local pressure gradient are introduced to minimize the numerical oscillation around solution discontinuities. An implicit Approximate LU (ALU) factorization scheme, originally developed on structured meshes, is adopted for triangular meshes, which, in principle, has similar convergence characteristics as that on structured meshes. To facilitate an effective vectorization of the ALU scheme, a colored black–gray–white (BGW) numbering technique is developed and implemented. This BGW numbering sorts the mesh cells into independent sets and effectively eliminates the data dependency in the inversion process. This results in a high degree of vectorization of the final code. Numerical examples of transonic Ringleb flow, transonic channel flow with circular bump, supersonic shock reflection flow and subsonic flows over multielement aerofoils are calculated to validate the methodology.

2. MATHEMATICAL MODEL

The 2D Euler equations without body force and heat source can be written in integral form as

$$\frac{\partial}{\partial t}\int\int_{\Omega} Q\,\mathrm{d}\Omega + \oint_{S} \mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S = 0, \tag{1}$$

with

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ e\end{pmatrix}, \qquad \mathbf{F}=f\hat{\mathbf{i}}+g\hat{\mathbf{j}}$$

and

$$f=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (e+p)u\end{pmatrix}, \qquad g=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (e+p)v\end{pmatrix},$$

where Ω is the domain of interest and S is the boundary surrounding Ω, $\hat{\mathbf{n}}$ is the unit normal of S in outward direction, Q represents the vector of conserved variables and $\mathbf{F}$ the vector of flux

functions. The pressure is obtained by the equation of state:

$$p=(\gamma-1)\left[e-\rho\frac{(u^2+v^2)}{2}\right], \tag{2}$$

where γ is the ratio of specific heats. In the above equations, the density is non-dimensionalized by the free-stream density ρ_∞, velocities u and v by free-stream sound speed a_∞, and energy per unit volume and pressure by $\rho_\infty a_\infty^2$.

2.1. *First-order scheme*

The flow field is discretized using triangular cells to form an unstructured mesh. The average of conserved variables is stored at the centre of each triangular cell. The edges of each cell define the faces of the triangular control volume. For each triangular control volume, equation (1) can be rewritten as

$$\frac{\partial Q_i V_i}{\partial t}=-\oint_{\partial C_i}\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S, \tag{3}$$

where Q_i now represents the averaged conserved variables of cell i, ∂C_i and V_i denote the boundary cell face and the volume of cell i, respectively. To evaluate the right-hand side of equation (3), the flux vectors over each face of ∂C_i are summed, resulting in the discrete form of flux integral as

$$\oint_{\partial C_i}\mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}S=\sum_{j=k(i)}F_{i,j}\Delta S_j, \tag{4}$$

where $F_{i,j}$ is the numerical flux at each cell face, ΔS_j is the length of each cell face, and $k(i)$ is a list of neighbouring cells of cell i. For a cell-centred scheme, the number of cell faces of each triangular cell is always three. To evaluate the numerical flux $F_{i,j}$, Roe's[9] flux difference scheme is applied locally at each cell face, assuming 1D Riemann problem in the normal direction. Two discrete Riemann states separated by the cell interface are first interpolated using the cell-centre values. For a first-order scheme, a piecewise constant distribution is assumed:

$$Q_{i,j}^-=Q_i,\quad Q_{i,j}^+=Q_j, \tag{5}$$

where + and − superscripts indicate right and left Riemann states in the counterclockwise sense. The numerical flux $F_{i,j}$ is obtained by

$$F_{i,j}=\tfrac{1}{2}[F(Q_{i,j}^+)+F(Q_{i,j}^-)-|A_{\mathrm{Roe}}|(Q_{i,j}^+-Q_{i,j}^-)], \tag{6}$$

where A_{Roe} is the flux Jacobian evaluated using Roe's average fluid states. The absolute value symbols indicate that the absolute value of the eigenvalues were used to evaluate A_{Roe}.

2.2. *High-order schemes*

To construct high-order schemes, a linear- or higher-order distribution of flow variables over each cell is assumed to replace the piecewise constant cell distribution. The left and right Riemann states of each cell face are reconstructed using the linear- or high-order assumption, from which the numerical fluxes are computed. A piecewise linear reconstruction of the cell A can be represented by

$$Q(x,y)=Q_A+\nabla Q_A\cdot\mathbf{r}, \tag{7}$$

where $\mathbf{r}$ is the vector from the cell centre A to any point (x, y) in the cell, Q_A is the cell-centre value of cell A, and ∇Q_A represents the gradient vector of cell A evaluated using the cell-centre values. It is the evaluation of ∇Q_A which distinguishes various schemes.

By treating equation (7) as the first-order Taylor expansion of $Q(x, y)$, it is evident that a first-order accurate ∇Q_A can be solved by three known point values of Q. However, the three known points must not be collinear, that is, they must form a triangle. This is equivalent to assuming that a constant ∇Q_A exists in the contour enclosed by the three known points, and Gauss theorem can be applied to evaluate the gradient as

$$\nabla Q_A = \frac{1}{a_\Omega} \oint_{\partial a} Q \hat{\mathbf{n}} \, dS + O(\Delta x, \Delta y), \tag{8}$$

where ∂a is the integration path connecting three known points and a_Ω is the area contained in the path. It is well known that Gauss theorem can be applied to arbitrary polygon contours, but the accuracy of the gradient estimation decreases as more points are brought in from further away. Only the result of triangular contour is equivalent to a Taylor expansion.

To obtain the gradient of cell A in Figure 1, it is reasonable to utilize the information at A itself and its neighbouring cells B, C and D. Among four possible integration path connecting three cell centres, Barth and Jespersen[6] suggest the integration path connecting the neighbouring cell centres B, C and D. Note that in extreme cases where highly stretched and skewed triangles are formed, this integration path may degenerate into a straight line. To avoid this singularity, Barth and Jespersen further suggested that the set of all cell centres neighbouring to and sharing a common vertex with cell A be connected to form the integration path. In this paper, the former path is taken under the assumption that proper cautions have been taken in the grid generation effort to detect and eliminate singular cells.

Among many possible candidates examined in Reference 10, a simple and accurate gradient estimation based on the weighted average of Barth and Jespersen's three-point model is used in this paper:

$$\nabla Q_A^{\text{mod}} = (\nabla Q_A + \omega_B \nabla Q_B + \omega_C \nabla Q_C + \omega_D \nabla Q_D)/2,$$

$$\omega_B = \frac{\mathbf{r}_{AC} \times \mathbf{r}_{CD}}{\mathbf{r}_{BC} \times \mathbf{r}_{CD}}, \qquad \omega_C = \frac{\mathbf{r}_{AD} \times \mathbf{r}_{DB}}{\mathbf{r}_{BC} \times \mathbf{r}_{CD}}, \qquad \omega_D = \frac{\mathbf{r}_{AB} \times \mathbf{r}_{BC}}{\mathbf{r}_{BC} \times \mathbf{r}_{CD}}, \tag{9}$$

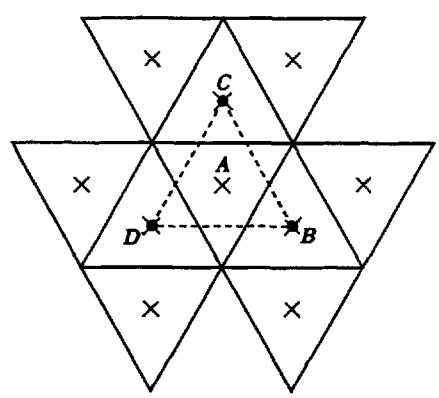

Figure 1. •: Stencil support for unmodified gradient ∇Q_A; ×: stencil support for modified gradient ∇Q_A^{mod}

where $\nabla Q_{B,C,D}$ are Barth and Jespersen's gradient of cell B, C and D, respectively, $\mathbf{r}_{AC}$ is the distance vector from cell centres A to C and other r's are defined similarly. Note that the sum of the area weighting factors $\omega_{B,C,D}$ is always one, regardless of whether centre A is inside the path BCD or not. This results a 10-point gradient model, as shown in Figure 1. There are two advantages of using this modified gradient. First, on a 'regular' triangular mesh made of equilateral triangles, it can be proven that the modified gradient is of the order of $O(\Delta x^2, \Delta y^2)$. Second, the process of area weighting usually reduces the unevenness in the computed gradients, and the stability of the overall scheme may be enhanced. This weighted averaging process of equation (9) can be easily extended to 3D tetrahedral meshes. Results of 3D tests will be reported in another paper.

3. CHARACTERISTIC LIMITER

The reconstruction mentioned above can be applied to conserved variables or primitive variables. Limiters of some sort are often applied to the interpolated quantities to minimize the numerical oscillation around solution discontinuities. It is known from the experience of structured grid methods[11] that when flow discontinuities are present, the use of characteristic variables in the limiting process can in general produce better results than using primitive or conserved variables. On structured grid methods, the reconstruction is generally quasi-one-dimensional along each co-ordinate direction. Thus, it is a standard practice to use cell-face normals to define the characteristic direction along which the characteristic variables are defined. For the proposed scheme on triangular meshes, however, the reconstruction is truly two-dimensional and difficulty arises in defining the characteristic directions for each cell.

Here, it is chosen to define a unified characteristic direction for the three cell faces of each cell. In order to produce the best limiting result, the chosen direction should be aligned with the direction of pertinent solution gradient in the flow field, or normal to the pertinent solution discontinuity. For example, the direction of pressure gradient is a natural indication of the shock. Similarly, the direction of density gradient can capture a contact surface, and the direction of tangential velocity gradient is proper for a slip line. One may try to use a single gradient, e.g. entropy gradient, to indicate various possible discontinuities in the flow field. But entropy jump is normally small compared with other jumps across shocks. In the present paper, local pressure gradient is chosen to demonstrate the use of characteristic limiting when shock is present in the flow field. In regions of small pressure gradient, the direction of maximum solution gradient of primitive variables is used instead.

The limited version of the reconstruction of cell A can be written as

$$Q(x, y) = Q_A + R_p \Phi_A (L_p \nabla Q_A^{\text{mod}} \cdot \mathbf{r}), \quad \Phi_A \in [0, 1], \tag{10}$$

where Φ_A is a chosen limiter for cell A, L_p and R_p are the left and right eigenvectors of the flux Jacobian evaluated at the cell centre of cell A using the direction of local pressure gradient. If individual cell-face normals were used instead of a unified characteristic direction, the L_p and R_p in equation (10) would be different for different cell faces. A popular limiter following Barth and Jespersen[6] can be written as

$$\Phi_{A_j} = \min(\Phi_{A_j}), \quad j = k(i),$$

where Φ_{A_j} is computed at each face j of cell A as

$$\begin{aligned} \Delta W_{Aj} &= L_p (Q_j - Q_A), \\ \Delta W_A^{\min} &= \min\{\Delta W_{AB}, \Delta W_{AC}, \Delta W_{AD}, 0\}, \\ \Delta W_A^{\max} &= \max\{\Delta W_{AB}, \Delta W_{AC}, \Delta W_{AD}, 0\}, \end{aligned} \tag{11}$$

$$\Phi_A(\Delta W_{Aj})=\begin{cases}\min(1, \Delta W_A^{\max}/\Delta W_{Aj}) & \text{if } \Delta W_{Aj}>0,\\ \min(1, \Delta W_A^{\min}/\Delta W_{Aj}) & \text{if } \Delta W_{Aj}<0,\\ 1 & \text{if } \Delta W_{Aj}=0,\end{cases}$$

where Q_j is the unlimited cell-face value of cell A obtained by $\nabla Q_A^{\text{mod}} \cdot \mathbf{r}$.

Note that the gradient vector ∇Q_A^{mod} can be evaluated using either primitive variables or conserved variables. The corresponding L_p is then applied to obtain characteristic variables in difference form before limiting. The limiter Φ_A is applied to characteristic variables after which the corresponding R_p is used to recover the primitive or conserved variables. It has been our experience that whether the primitive or conserved variables are being reconstructed has very minor effect on the final accuracy of the solution. Since the primitive variables have a much simpler eigenvector than that of the conserved variables, it is chosen to use primitive variables in the reconstruction process.

4. IMPLICIT TIME INTEGRATION

The semidiscrete form of equation (3) can be written as

$$\frac{\partial Q_i V_i}{\partial t}=-\sum_{j=k(i)} F_{i,j}\Delta S_j=-R(Q). \tag{12}$$

For implicit time integration, the Euler implicit method is chosen:

$$V\frac{\Delta Q^n}{\Delta t}=-R(Q^{n+1}), \tag{13}$$

where $\Delta Q^n=Q^{n+1}-Q^n$ and Δt is the time increment. This equation can be linearized about a known state Q^n to obtain

$$\left[\frac{VI}{\Delta t}+\frac{\partial}{\partial Q}R(Q)\right]\Delta Q^s=-\left[\frac{(Q^s-Q^n)}{\Delta t}V+R(Q^s)\right]=\text{RHS}, \tag{14}$$

where $\Delta Q^s=Q^{s+1}-Q^s$ and I is the identity matrix. Here 's' is a subiteration index which can be subiterated to obtain a better approximation Q^{s+1} to Q^{n+1}.

To simplify the inversion process of equation (14), only a first-order spatial accurate scheme is used on the left-hand side, while the full high-order accuracy is kept on the right-hand side. This simplification will not alter the time and spatial accuracy of the solution when the subiteration in 's' converges. The fully discretized equation can be written in matrix form as

$$(D+L+U)\Delta Q^s=\text{RHS}, \tag{15}$$

where D is a block diagonal matrix, L is a block lower triangular matrix with zero diagonal elements, and U is a block upper triangular matrix with zero diagonal elements. For the 2D Euler equations, the elements of L, D and U are all 4×4 matrices.

The direct inversion of equation (15) is still prohibited due to the need for huge computer resources. This is particularly true for unstructured grid methods since the construction of L and U is unstructured and sparse. The choice of relaxation method for solving equation (15) is also limited on unstructured meshes, because the advantage of a structured ordering of mesh cells is lost. For example, the popular ADI scheme (approximate factorization) or line Gauss–Seidel scheme is not directly applicable on unstructured meshes. Thus, it is desirable to develop

a relaxation method using unstructured random ordering, yet still maintaining a similar convergence characteristic as that on structured meshes.

In the present paper, an ALU scheme originally developed for structured grid method[18] is adopted to factorize the left-hand side of equation (15) as

$$(D+L)D^{-1}(D+U)\Delta Q^s = \mathrm{RHS}. \tag{16}$$

The factorization error of equation (16) is $LD^{-1}U\Delta Q^s$. The three sweeps of the inversion process of equation (16) are:

$$\begin{aligned} &\text{lower sweep:} && (D+L)\Delta Q^* = \mathrm{RHS},\\ &\text{diagonal sweep:} && D^{-1}\Delta Q^{**} = \Delta Q^*,\\ &\text{upper sweep:} && (D+U)\Delta Q^s = \Delta Q^{**}. \end{aligned}$$

Since only first-order upwind method is used on the left-hand side, the lower and upper triangular matrices of the above steps are diagonally dominant. The inversion processes are simple because they require only matrix multiplications and additions and 4×4 matrix inversions. Note that this ALU factorization scheme is applicable to both structured and unstructured meshes, and to 2D as well as 3D problems. This is because the operator split of the left-hand side into L, D and U requires no prior knowledge of dimensionality or mesh structure. At least in principle, the convergence of the ALU scheme depends only on the sequence of information propagation of boundary conditions. Hence, with a proper handling of boundary points, it is expected that the convergence rate of the ALU scheme should not deteriorate when applied to unstructured schemes.

To make the ALU scheme fully vectorizable, a black–gray–white (BGW) colouring and numbering technique similar to the checkboard colouring on structured grid is developed and implemented. The computational cells are divided into three sets of different colours, namely, black, gray and white. The rule of colouring is that no two adjacent cells have the same colour. Note that this rule can be followed for most part of the computational domain by using only two colours, say black and white. Only for a small portion of the computational domain, gray cells are needed to separate the black and white cells. On a 'regular' triangular mesh, no gray cell is needed at all. The fully discretized equations with BGW colouring can be written as

$$\begin{pmatrix} D_{\mathrm{B}} & U_B^{\mathrm{G}} & U_B^{\mathrm{W}} \\ L_{\mathrm{G}}^{\mathrm{B}} & D_{\mathrm{G}} & U_{\mathrm{G}}^{\mathrm{W}} \\ L_{\mathrm{W}}^{\mathrm{B}} & L_{\mathrm{W}}^{\mathrm{G}} & D_{\mathrm{W}} \end{pmatrix} \begin{pmatrix} \Delta Q_{\mathrm{B}}^s \\ \Delta Q_{\mathrm{G}}^s \\ \Delta Q_{\mathrm{W}}^s \end{pmatrix} = \begin{pmatrix} \mathrm{RHS}_{\mathrm{B}} \\ \mathrm{RHS}_{\mathrm{G}} \\ \mathrm{RHS}_{\mathrm{W}} \end{pmatrix}, \tag{17}$$

where subscripts B, G and W indicate the cell colour under consideration and superscripts B, G and W indicate the neighbouring cell colours. The three sweeps of the ALU scheme are now as follows.

Lower sweep:

for B: $\Delta Q_{\mathrm{B}}^* = D_{\mathrm{B}}^{-1}\,\mathrm{RHS}_{\mathrm{B}}$,

for G: $\Delta Q_{\mathrm{G}}^* = D_{\mathrm{G}}^{-1}[\mathrm{RHS}_{\mathrm{G}} - L_{\mathrm{G}}^{\mathrm{B}}(\Delta Q_{\mathrm{B}}^*)]$,

for W: $\Delta Q_{\mathrm{W}}^* = D_{\mathrm{W}}^{-1}[\mathrm{RHS}_{\mathrm{W}} - L_{\mathrm{W}}^{\mathrm{B}}(\Delta Q_{\mathrm{B}}^*) - L_{\mathrm{W}}^{\mathrm{G}}(\Delta Q_{\mathrm{G}}^*)]$.

Diagonal sweep:

for B, G and W: $\Delta Q^{**} = D\Delta Q^*$.

Upper sweep:

for W: $\Delta Q^s_W = D_W^{-1} \Delta Q^{**}_B$,

for G: $\Delta Q^s_G = D_G^{-1}[\Delta Q^{**}_G - U^W_G(\Delta Q^s_W)]$,

for B: $\Delta Q^s_B = D_B^{-1}[\Delta Q^{**}_B - U^W_B(\Delta Q^s_W) - U^G_B(\Delta Q^s_G)]$.

Note that since only a first-order spatial accurate scheme is used on the left-hand side, this BGW colouring has completely eliminated the data dependency and allows a full vectorization of the inversion process. The cost of this high degree of vectorization may be some extra memory for the colour management, and more seriously, a change in the convergence characteristics because of the change in the sequence of information propagation of boundary conditions. In the following numerical experiments, the actual benefit of this BGW colouring on vector computers will be investigated.

For comparison, the explicit four-stage Runge–Kutta method (RK4), and the implicit point Gauss–Seidel method (PGS) used in Reference 8 will also be described briefly. The RK4 method can be written without subscripts i and j as

$$Q^{(1)} = Q^n - \alpha_1 \frac{\Delta t}{V} R(Q^n), \qquad Q^{(2)} = Q^n - \alpha_2 \frac{\Delta t}{V} R(Q^{(1)}),$$

$$Q^{(3)} = Q^n - \alpha_3 \frac{\Delta t}{V} R(Q^{(2)}), \qquad Q^{n+1} = Q^n - \alpha_4 \frac{\Delta t}{V} R(Q^{(3)}), \tag{18}$$

where n is the index in time, Δt is the time increment and the coefficients are

$$\alpha_1 = 0{\cdot}15, \qquad \alpha_2 = 0{\cdot}3275, \qquad \alpha_3 = 0{\cdot}57, \qquad \alpha_4 = 1{\cdot}0$$

These coefficients have been experimentally determined to accelerate convergence to steady state in upwind structured code.[12] In addition, the convergence of RK4 method to the steady state is also accelerated by the use of implicit residual smoothing and local time stepping.[13] To solve equation (15) approximately, the implicit PGS method neglects the U part of the implicit side to obtain

$$(D + L)\Delta Q^s = \text{RHS}. \tag{19}$$

The inversion of this equation is simple and straightforward.

5. NUMERICAL TESTS

Various numerical tests have been performed to validate and evaluate the schemes outlined above. These include tests for (1) the accuracy of the solution algorithms, (2) the effects of the limiter and (3) the efficiency of the time integration methods. Finally, the flow over a four-element aerofoil is computed to demonstrate the capability of the scheme to handle complex geometries. For most of the tests, characteristic boundary conditions[14] are used for the far field, and tangency conditions are used for the solid bodies. Exact boundary conditions are used whenever applicable.

5.1. The accuracy of the solution algorithm

The numerical solution of the Ringleb flow is computed and compared with analytical solution.[15] The two Riemann states at the cell interface are computed using various reconstruction models without limiting. Results of three types of reconstruction models are compared:

(1) Type 1 which uses the constant distribution of equation (5), (2) Type 2 which is Barth and Jespersen's three-point linear model of equation (8), and (3) Type 3 which is the proposed 10-point linear model of equation (9). For the sake of comparison, both first-order and second-order Gauss quadrature integrals are used in the flux evaluation. A sequence of three meshes are used for grid refinement: coarse (374 cells), medium (711 cells) and fine (1020 cells). The coarse grid and the density contours of the exact solution are shown in Figures 2 and 3.

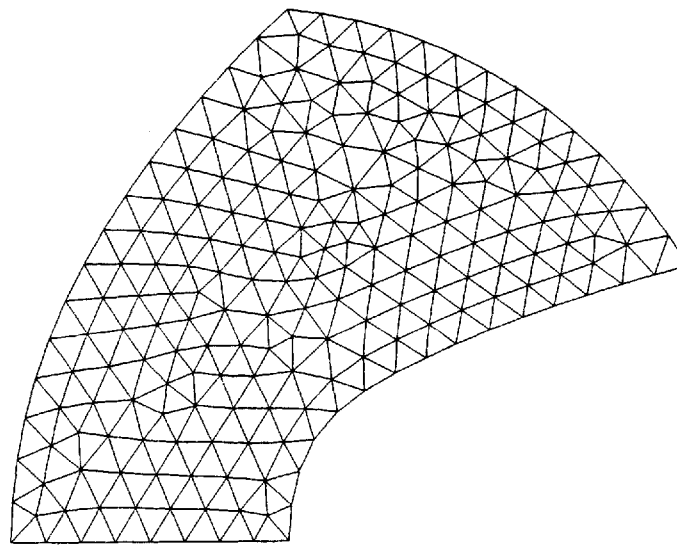

Figure 2. Coarse grid for Ringleb flow problem, 374 cells

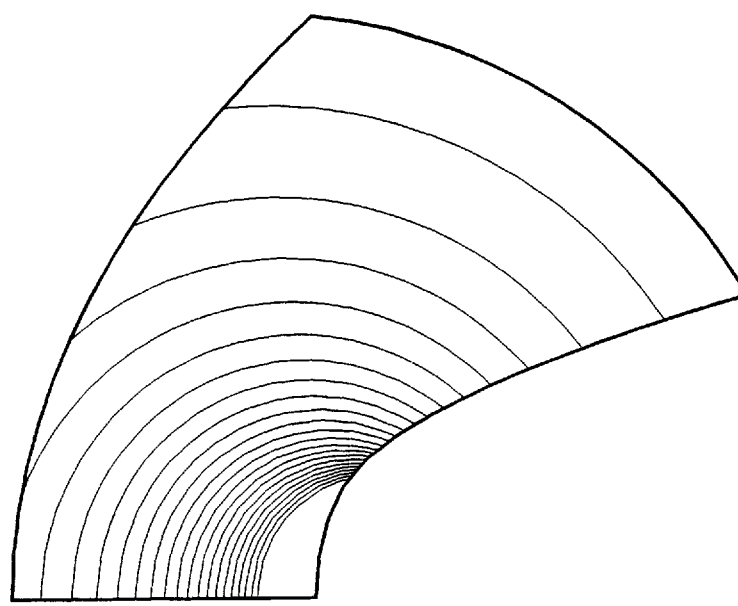

Figure 3. Exact density contours of Ringleb flow problem

The $L2$ norm of the solution error is defined as

$$L2=\left\{\left[\sum_{i=1}^{N}(Q_e-Q_n)^2\right]\Big/N\right\}^{1/2},$$

where Q_e is the exact solution, Q_n is the numerical solution and N is the number of cells used. Since a curved boundary will introduce a truncation error from modelling the boundary by a series of straight-line segments, all cells adjacent to boundary surfaces are not included in the $L2$ calculation. This exclusion of boundary cells will make the order estimation more conservative in general. Figure 4 and Table I show the $L2$ norm of the solution errors with grid refinement. The horizontal axis is the reciprocal of square root of the total cell number, which is equivalent to the grid spacing Δx and Δy on structured meshes. Another parameter NG is the order of Gauss quadrature integral used in the flux evaluation. Table I(a) lists the numerical data of Figure 4. Table I(b) shows the data fitting of Figure 4 by power curves $y=ax^b$, where b is equivalent to the order of the solution in Δx and Δy. It is seen that Types 2 and 3 are all second-order schemes with decreasing $L2$ norms. The second-order Gauss quadrature is effective for Type 3, but not for Type 2. This indicates that the accuracy of Type-2 gradient has limited the accuracy of the solution algorithm.

A further test is done on a mesh with poor quality as shown in Figure 5. The error norms computed on this poor mesh are recorded in Table II. It is evident that the Type-3 scheme is more accurate than the Type-2 scheme on this poor mesh. Again, the second-order Gauss quadrature is effective for Type 3, but not for Type 2. It can be concluded that the modified linear gradient of equation (9) is effective in increasing the accuracy of the original Barth and Jespersen's scheme.

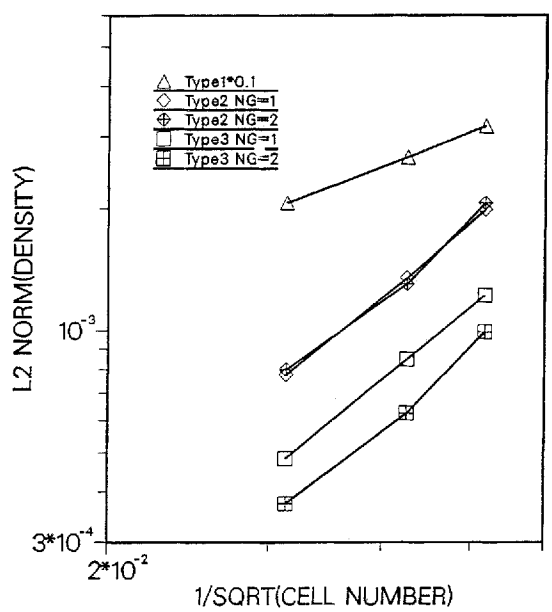

Figure 4. $L2$ error of density versus grid refinement for Ringleb flow problem

Table I(a). $L2$ norm of Ringleb flow on Euler solver

Mesh	Method type	$L2$ norm	
		$NG=1$	$NG=2$
Coarse grid	1	$3{\cdot}19\times10^{-2}$	
374 cells	2	$1{\cdot}99\times10^{-3}$	$2{\cdot}03\times10^{-3}$
	3	$1{\cdot}26\times10^{-3}$	$9{\cdot}95\times10^{-4}$
Medium grid	1	$2{\cdot}68\times10^{-2}$	
553 cells	2	$1{\cdot}36\times10^{-3}$	$1{\cdot}31\times10^{-3}$
	3	$8{\cdot}50\times10^{-4}$	$6{\cdot}25\times10^{-4}$
Fine grid	1	$2{\cdot}07\times10^{-2}$	
1020 cells	2	$7{\cdot}78\times10^{-4}$	$8{\cdot}02\times10^{-4}$
	3	$4{\cdot}85\times10^{-4}$	$3{\cdot}75\times10^{-4}$

Table I(b). Curve fitting of Figure 4 by power curve $y=ax^b$

Method type	NG	a	b
1	1	0·044	0·86
2	1	0·488	1·86
	2	0·493	1·86
3	1	0·291	1·85
	2	0·285	1·92

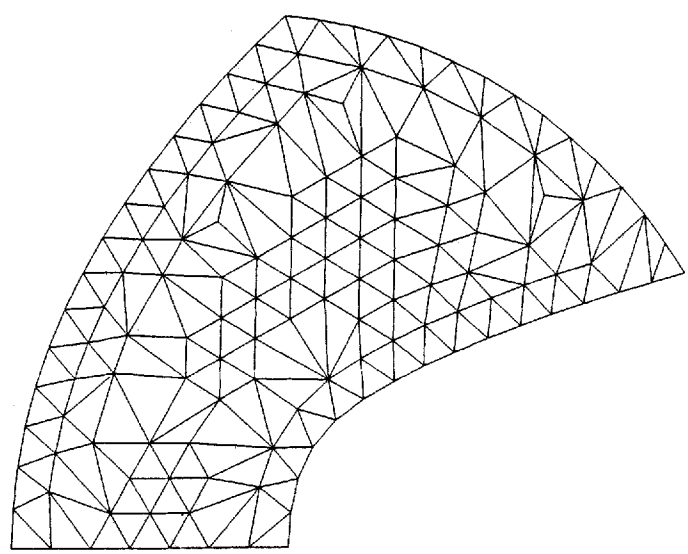

Figure 5. Poor grid for Ringleb flow problem; 221 cells

Table II. $L2$ norm of Ringleb flow on Euler solver applied to poor grid in Figure 5

Mesh	Method	$L2$ norm	
		$NG=1$	$NG=2$
Poor grid	1	$4{\cdot}09\times10^{-2}$	
221 cells	2	$8{\cdot}54\times10^{-3}$	$9{\cdot}54\times10^{-3}$
	3	$3{\cdot}25\times10^{-3}$	$2{\cdot}49\times10^{-3}$

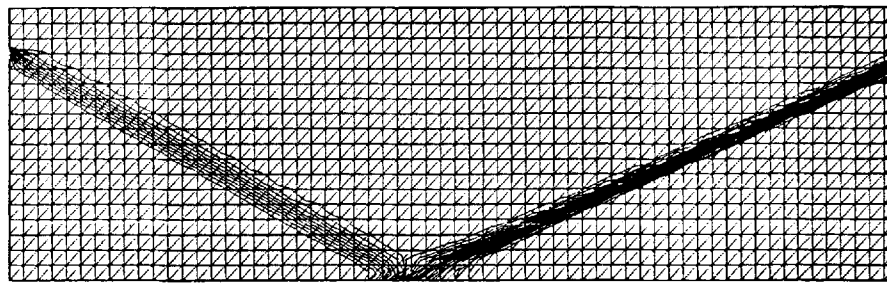

Figure 6. Mesh and pressure contours for shock reflection problem: 2232 cells, $M_\infty=2{\cdot}9$, $\beta=-29^\circ$

5.2. *The effects of limiter*

To examine the effects of the characteristic limiter, an $M_\infty=2{\cdot}9$ shock reflection problem is computed with the limiter turned on. The shock angle is -29° to the horizontal. A mesh of 2232 cells on a rectangular region of length 4 and height 1·2 and the computed pressure contours using Type-3 reconstruction are shown together in Figure 6. Note that the captured shock width is thicker before the reflection on the wall. This is due to the fact that the mesh cells after reflection align better with the reflected shock.

As mentioned in Section 3, the limiter can be applied to conserved variables, primitive variables, or characteristic variables. Figures 7 and 8 show the convergence history of density of Type-2 and Type-3 schemes with the limiter applied to different variables. It is clear that Type-3 scheme converges better and faster than Type-2 scheme in general, and the characteristic limiter results in the best convergence for both schemes. The computed pressure profiles along the wall using Type-3 scheme are shown in Figure 9. The first-order result is also shown for comparison. It is seen that the characteristic limiter yields the least numerical oscillation. Results using Type-2 scheme are similar and not shown here.

5.3. *The efficiency of time integration methods*

The steady state of an $M_\infty=0{\cdot}675$ transonic flow in a 2D channel with a 10% circular bump is computed using various time integration methods, namely, RK1, RK4, PGS, ALU and ALU with BGW colouring. A mesh of 1071 cells used in the computation is shown in Figure 10. The solution algorithms investigated are the first-order scheme using Type-1 reconstruction and the second-order scheme using Type-3 reconstruction with characteristic limiter. The computed surface pressures are compared with the result obtained by a structured upwind TVD method of

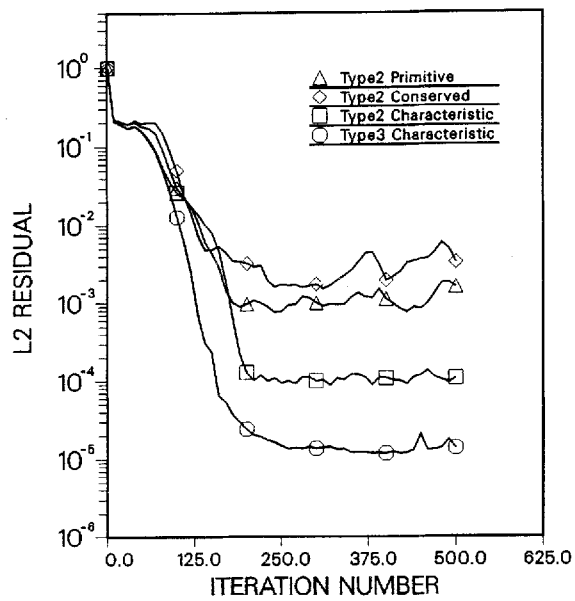

Figure 7. $L2$ residual of density versus iteration count for shock reflection problem; Type 2: 2232 cells, $M_\infty = 2{\cdot}9$, $\beta = -29^\circ$

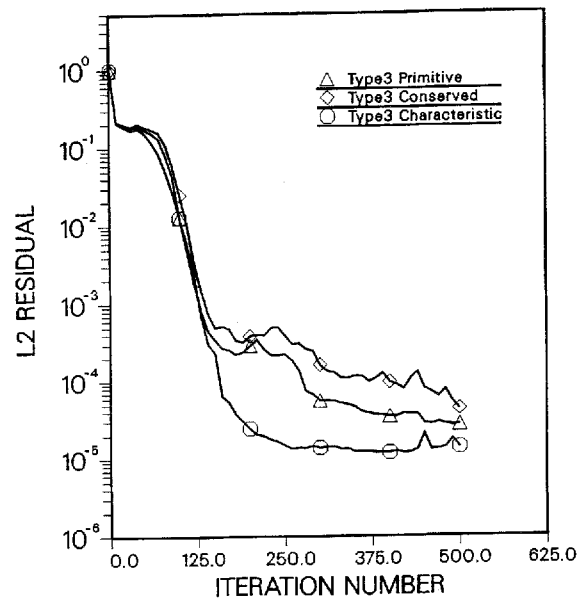

Figure 8. $L2$ residual of density versus iteration count for shock reflection problem; Type 3: 2232 cells, $M_\infty = 2{\cdot}9$, $\beta = -29^\circ$

Chakravarthy[16] on a 64×22 mesh in Figure 11. The second-order results agree well, except for some slight differences behind the shock. The convergence of the $L2$ residual of density versus iteration number and CPU time are plotted in Figures 12 and 13 for the first-order scheme, and in Figures 14 and 15 for the second-order scheme.

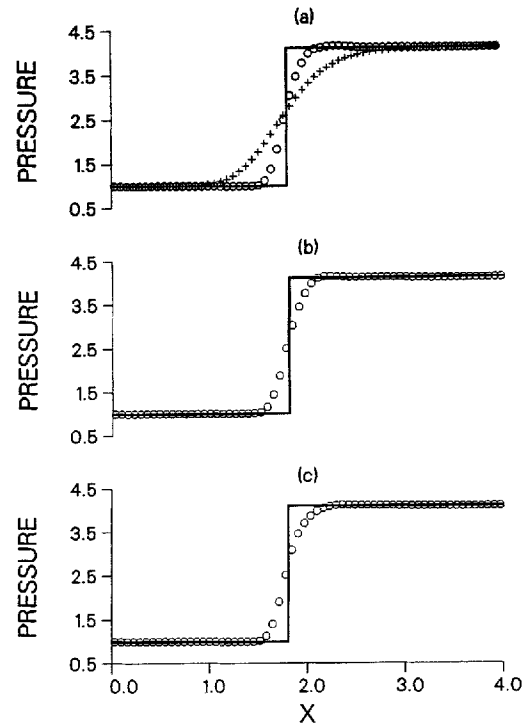

Figure 9. Pressure distribution on the lower wall for shock reflection problem: 2232 cells, $M_\infty = 2{\cdot}9$, $\beta = -29°$. Lines: exact solution, +: Type-1 solution without limiting, ○: Type-3 solution with limiting on (a) primitive variables, (b) conserved variables and (c) characteristic variables

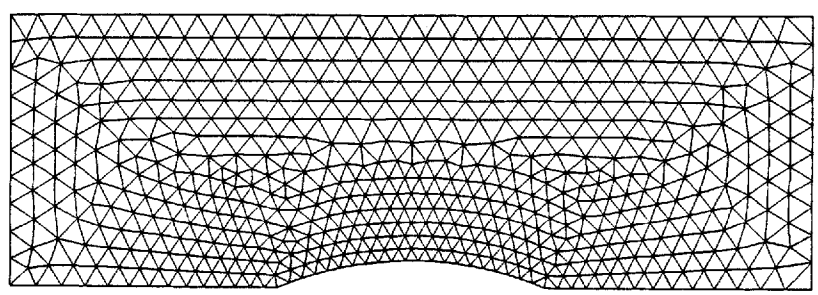

Figure 10. Mesh for transonic channel flow with 10% circular bump: 1071 cells, $M_\infty = 0{\cdot}675$

For the first-order results, the ALU method has the least iteration number, while the ALU + BGW method has the least CPU time on Alliant FX/80 single vector processor. The CPU time per iteration per cell is $1{\cdot}13 \times 10^{-3}$ s for the ALU method and $0{\cdot}63 \times 10^{-3}$ s for the ALU + BGW method. As mentioned before, the BGW colouring changes the sequence of information propagation, and hence slows down the convergence of the method in general. On the other hand, the BGW colouring allows the full vectorization of the inversion process, which in this case has saved as much as 45% of the CPU time per iteration. Of course, the actual saving

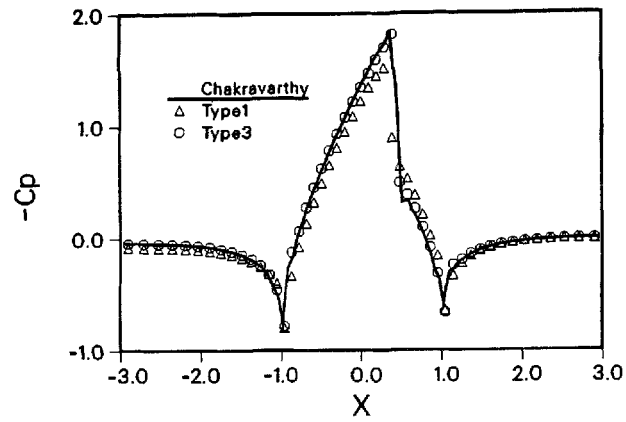

Figure 11. Surface Cp distribution for transonic channel flow with 10% circular bump: 1071 cells, $M_\infty = 0{\cdot}675$. Line: Chakravarthy's scheme, symbol: computed

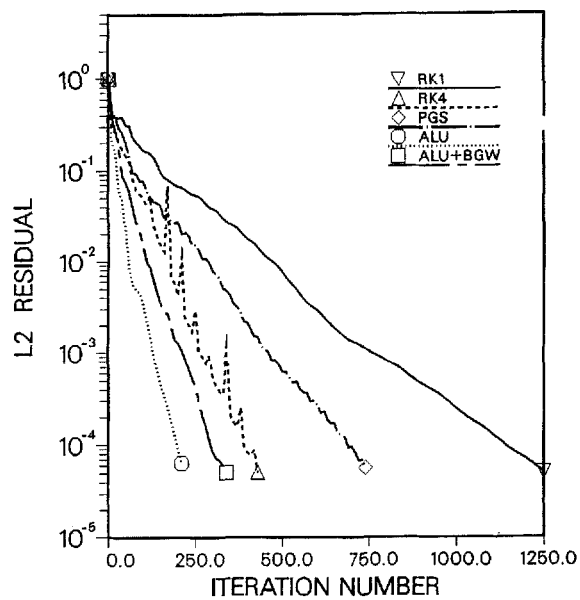

Figure 12. $L2$ residual of density versus iteration count for transonic channel flow with 10% circular bump; Type 1: 1071 cells, $M_\infty = 0{\cdot}675$

may depend on the programming technique as well as the computer hardware. The RK1 method performed well in terms of CPU time, since it is a single-stage method and fully vectorizable. The RK4 method is the most unstable scheme probably because of a CFL number too large. The PGS method is the most expensive one in terms of CPU time.

As for the second-order results, RK1 fails to converge at all and PGS is slow to converge. The ALU method has the best convergence in terms of both iteration count and CPU time. The CPU time per iteration per cell is $1{\cdot}37 \times 10^{-3}$ s for the ALU method and $0{\cdot}83 \times 10^{-3}$ s for the

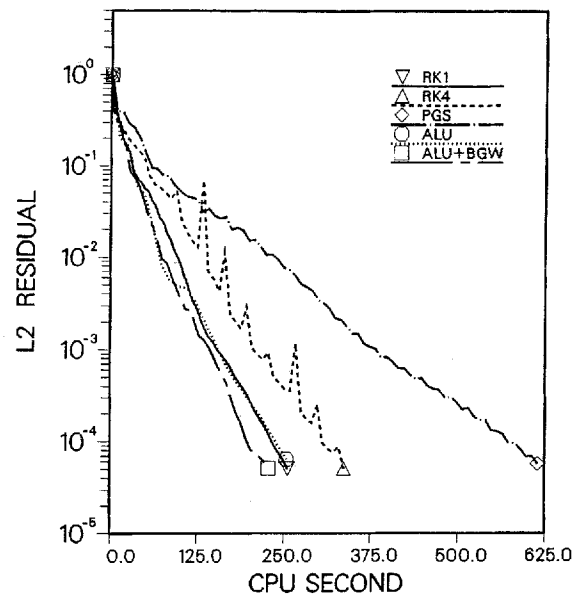

Figure 13. $L2$ residual of density versus CPU time(s) for transonic channel flow with 10% circular bump; Type 1: 1071 cells, $M_\infty = 0{\cdot}675$

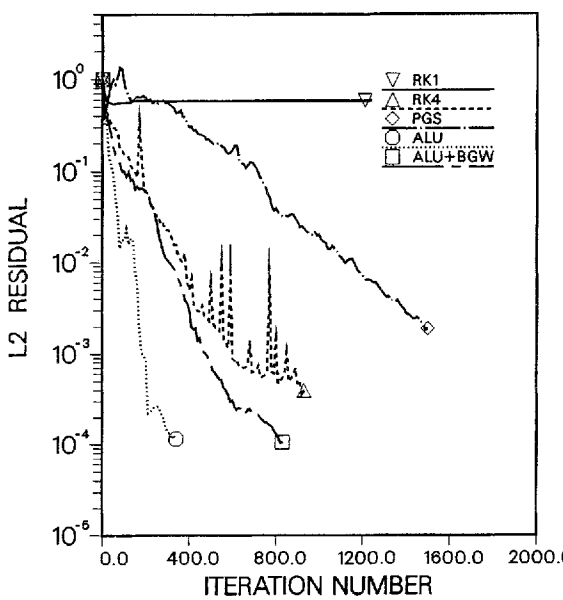

Figure 14. $L2$ residual of density versus iteration count for transonic channel flow with 10% circular bump; Type 3: 1071 cells, $M_\infty = 0{\cdot}675$

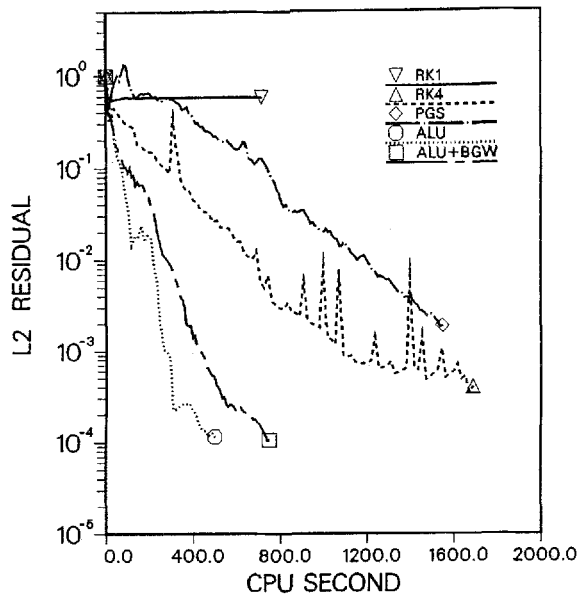

Figure 15. $L2$ residual of density versus CPU time (s) for transonic channel flow with 10% circular bump; Type 3: 1071 cells, $M_\infty = 0{\cdot}675$

ALU + BGW method. However, the decrease in the convergence rate due to BGW colouring is much more significant than that in the first-order case. Hence, although it is fully vectorized by using BGW colouring, the ALU + BGW method is still slower than the ALU method in terms of CPU time. On a computer of higher vector performance than Alliant FX/80, the actual saving due to BGW colouring may be increased. Nevertheless, it is highly desirable to develop a method that could adopt the BGW colouring technique without slowing down the convergence rate.

5.4. Numerical validation involving complex geometry

The numerical test involving complex geometries reported here is the subsonic flow over a four-element aerofoil defined in Suddhoo.[17] A mesh with 7160 cells used in the test is shown in Figure 16. The surface pressure coefficient of an $M_\infty = 0{\cdot}3$ solution using Type-3 reconstruction without limiter is compared with the incompressible solution in Figure 17. The comparison is reasonably accurate except for some local area of upper surfaces where flow expansion occurs. The differences may be attributed to the compressibility effect. The convergence of the $L2$ residual of density is shown in Figure 18. For the scheme using Type-3 reconstruction, the ALU method converges at a satisfactory rate, i.e. about 140 steps per order of magnitude drop in $L2$ residual. The BGW colouring has slowed down the convergence of the ALU method as expected. On Alliant FX/80, the CPU time per iteration per cell is $1{\cdot}21 \times 10^{-3}$ s for the ALU method and $0{\cdot}80 \times 10^{-3}$ s for the ALU + BGW method. As for the Type-2 scheme, the convergence is slow even using the ALU method under the same conditions as for the Type-3 scheme, i.e. the same CFL number (about 10), initial and boundary conditions, etc. There are noticeable and frequent oscillations in the convergence of the Type-2 scheme, in contrast to that of the Type-3 scheme. This indicates that the averaging process of equation (9) is effective in stabilizing the overall

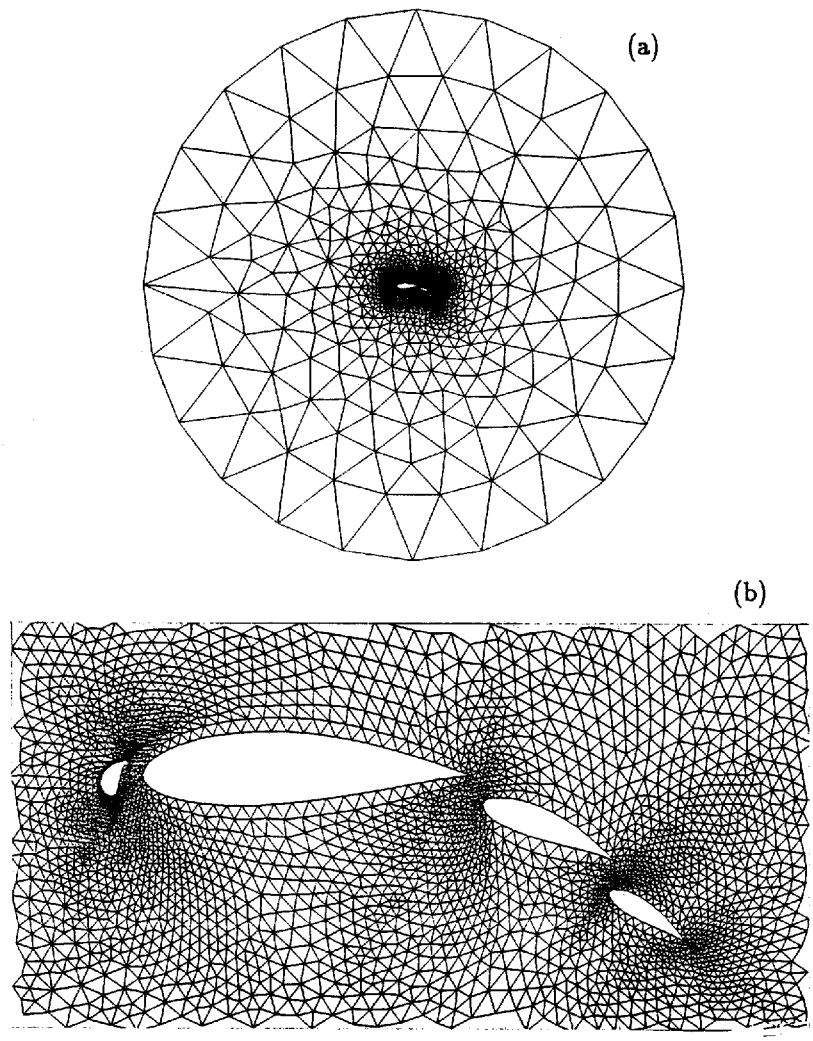

Figure 16. Mesh for subsonic flow over four-element aerofoils; 7160 cells: (a) full computational domain, (b) aerofoil region

scheme, and hence in increasing the convergence rate. It can be concluded in this test that the Type-3 reconstruction together with the ALU time integration constitute an accurate and stable scheme which can effectively handle the flow problem with complex geometries.

6. CONCLUSIONS

An accurate, stable and efficient upwind finite-volume scheme for the Euler solution on 2D unstructured triangular meshes has been developed and tested. By taking the weighted average of Barth and Jespersen's three-point support gradient, a new linear gradient vector is introduced for the reconstruction model. The new gradient has a 10-point stencil support and is second-order accurate on a 'regular' triangular mesh. This gives an accurate and stable second-order Euler

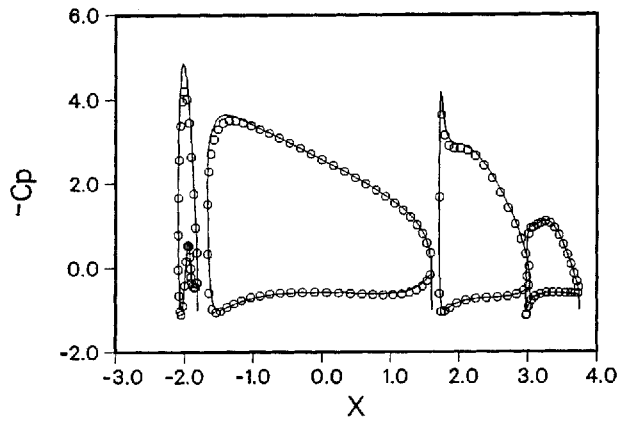

Figure 17. Surface Cp distribution for subsonic flow over four-element airfoils; $M_\infty = 0{\cdot}3$. line: incompressible solution; symbols: computed

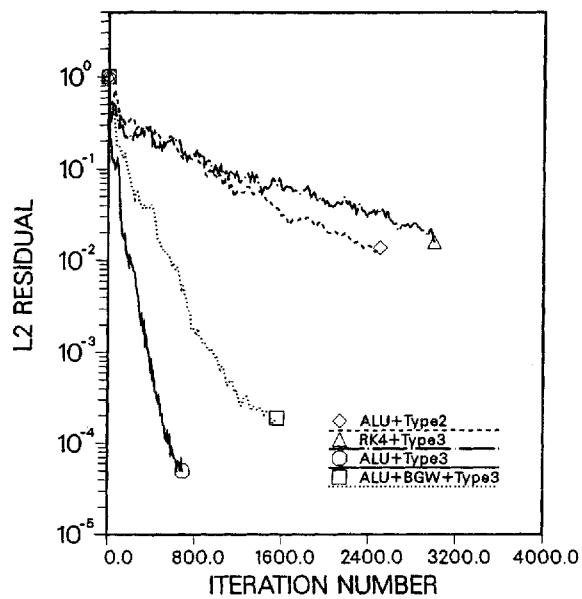

Figure 18. $L2$ residual of density versus iteration count for subsonic flow over four-element aerofoils; $M_\infty = 0{\cdot}3$

scheme in general. Characteristic variables in the direction of local pressure gradient are defined and used in the limiter, which is shown to be effective in eliminating the numerical oscillation around solution discontinuities. An ALU is developed for the implicit time integration and shows good convergence characteristics in all the numerical tests performed. To vectorize the scheme fully a BGW colouring technique is developed and implemented for the vector computer. However, this BGW colouring changes the sequence of information propagation, and hence slows down the convergence of the time integration. The actual saving of CPU time due to this BGW colouring is problem and computer dependent. Finally, the overall scheme is applied successfully to calculate the low subsonic flow over a complex four-element aerofoil.

ACKNOWLEDGEMENTS

This study is partially supported by FDID ROC, under contract CS79-0210-D006-15.

REFERENCES

1. A. Jameson and D. Mavriplis, 'Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh', *AIAA Paper 85-0435*, 1985.
2. D. Mavriplis, 'Multigrid solution of the Euler equations on unstructured and adaptive meshes', *ICASE Report No. 87-53*, 1987.
3. D. Mavriplis and A. Jameson, 'Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes', *AIAA Paper 87-0353*, 1987.
4. J. A. Desideri and A. Dervieux, 'Compressible flow solvers using unstructured grids', *VKI Lecture Series 1988-05*, 7–11 March 1988, pp. 1–115.
5. P. Rostand and B. Stoufflet, 'A numerical scheme for computing hypersonic viscous flows on unstructured meshes', *Proc. 2nd Inf. Conf. on Hypersonic Problems*, Aachen, W. Germany, 1988.
6. T. J. Barth and D. C. Jespersen, 'The design and application of upwind schemes on unstructured meshes', *AIAA Paper 89-0366*, 1989.
7. T. J. Barth and P. O. Frederickson, 'Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction', *AIAA Paper 90-0013*, 1990.
8. D. L. Whitaker, D. C. Slack and R. W. Walters, 'Solution algorithms for two-dimensional Euler equations on unstructured meshes', *AIAA Paper 90-0697*, 1990.
9. P. L. Roe, 'Approximate Riemann solvers, parameter vectors and difference schemes', *J. Comput. Phys.*, **43**, 357–372 (1981).
10. D. Pan and J.-C. Cheng, 'Reconstruction schemes for second-order Euler solution algorithms on two-dimensional unstructured triangular meshes', submitted to *J. Comput. Phys.*, currently under revision.
11. A. Dadone and B. Grossman, 'A rotated upwind scheme for the Euler equations', *AIAA Paper 91-0635*, 1991.
12. E. Turkel and B. Van Leer, 'Flux-vector splitting and Runge–Kutta methods for the Euler equations', *ICASE Report No. 84-27*, NASA CR 172415, June 1984.
13. A. Jameson, T. J. Baker and N. P. Weatherill, 'Calculation of inviscid transonic flow over a complete aircraft', *AIAA Paper 86-0103*, 1986.
14. D. Pan and S. R. Chakravarthy, 'Unified formulation for incompressible flows', *AIAA Paper 89-0122*, 1989.
15. G. Chiocchia, 'Exact solutions to transonic and supersonic flows', *AGARD Advisory Report AR-211*, 1985.
16. S. R. Chakravarthy, 'The versatility and reliability of Euler solvers based on high-accuracy TVD formulations', *AIAA Paper 86-0243*, 1986.
17. A. Suddhoo and I. M. Hall, 'Test cases for the plane potential flow past multi-element aerofoils', *Aeronaut. J.*, **89**, 403–414 (1985).
18. D. Pan and H. Lomax, 'A new approximate LU factorization scheme for the Reynolds-averaged Navier–Stokes equations', *AIAA J.*, **26**, 163–171 (1988).