AN EFFICIENT UPWIND/RELAXATION ALGORITHM FOR THE EULER AND NAVIER–STOKES EQUATIONS

GE-CHENG ZHA, DAO-ZHI LIU AND TIE-YOU MA

Department of Jet Propulsion, Beijing University of Aeronautics and Astronautics, Beijing, P.R. China

SUMMARY

An efficient Euler and full Navier–Stokes solver based on a flux splitting scheme is presented. The original Van Leer flux vector splitting form is extended to arbitrary body-fitted co-ordinates in the physical domain so that it can be used with a finite volume scheme. The block matrix is inverted by Gauss–Seidel iteration. It is verified that the often used reflection boundary condition will produce incorrect flux crossing the wall and cause too large numerical dissipation if flux vector splitting is used. To remove such errors, an appropriate treatment of wall boundary conditions is suggested. Inviscid and viscous steady transonic internal flows are analysed, including the case of shock-induced boundary layer separation.

KEY WORDS Flux splitting Implicit scheme Transonic flow

INTRODUCTION

The purpose of the present paper is to develop an efficient algorithm for obtaining steady state solutions to the compressible Euler and Navier–Stokes equations. The explicit procedure is easy to implement, but it is subject to the CFL condition which limits the size of the time step. The implicit approximate factorization (AF) procedure can have a larger time step than the explicit procedure, but the factorization introduces an error proportional to $(\Delta t)^3$ which restricts the time step from being too large. In three dimensions the stability restriction is even more severe.

The recently developed flux splitting scheme opens the door to a new class of efficient procedures. In 1979 Steger and Warming first proposed a flux vector splitting scheme for Euler equations. Van Leer suggested another form of flux splitting in 1982 which showed some improved properties.[1,2] With flux splitting, information in the flow field will propagate in the natural directions, and the implicit coefficient matrix will have more diagonally dominant properties which enable very efficient relaxation procedures to be used.[3] The implicit upwind (flux splitting) relaxation algorithm does not have the approximation error, so much larger time step can be used.[9–15]

The flux splitting scheme inherently possesses the dissipation needed to suppress numerical instabilities. The dissipation is related to the eigenvalue–eigenvector structure of the governing equations, but it causes too large numerical dissipation as a steady contact discontinuity (e.g. solid wall) is encountered.[4] The wall boundary treatment suggested in the present paper avoids this too large dissipation.

The integral finite volume method is used to solve the flow field in the physical domain, so some unnecessary linearization error caused by co-ordinate transformation, especially for non- orthogonal higly stretched and twisted meshes, will be reduced.

0271–2091/89/050517–13$06.50

Received 15 December 1987
Revised 10 July 1988

GOVERNING EQUATIONS

The governing equations are 2D time-dependent compressible Euler and full Navier–Stokes equations in non-dimensional conservation form:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\alpha\frac{1}{Re}\left(\frac{\partial \mathbf{R}}{\partial x}+\frac{\partial \mathbf{S}}{\partial y}\right). \tag{1}$$

Here

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ e\end{bmatrix},\quad \mathbf{F}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (e+p)u\end{bmatrix},\quad \mathbf{G}=\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ (e+p)v\end{bmatrix},$$

$$\mathbf{R}=\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ r_4\end{bmatrix},\quad \mathbf{S}=\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ s_4\end{bmatrix},$$

$$\sigma_{xx}=(\lambda+2\mu)u_x+\lambda v_y,\qquad \sigma_{yy}=(\lambda+2\mu)v_y+\lambda u_x,$$

$$\sigma_{xy}=\sigma_{yx}=\mu(u_y+v_x),\qquad r_4=u\sigma_{xx}+v\sigma_{xy}+\frac{1}{Pr}\frac{\partial T}{\partial x},$$

$$s_4=u\sigma_{xy}+v\sigma_{yy}+\frac{1}{Pr}\frac{\partial T}{\partial y},$$

$\alpha=0$ for inviscid flow, $\alpha=1$ for viscous flow, u and v are the velocity components in the x- and y-directions respectively and e is the total energy per unit volume. The molecular viscosity is determined by the Sutherland law, and Stokes' hypothesis is used for the bulk viscosity $\lambda=-2\mu/3$. The pressure is determined by the ideal gas law

$$p=(\gamma-1)[e-\rho(u^2+v^2)/2],$$

where ρ is the density and γ is the ratio of specific heats, taken as $\gamma=1{\cdot}4$.

Let

$$\mathbf{P}=\mathbf{F}'\mathbf{i}_x+\mathbf{G}'\mathbf{i}_y$$

and

$$\mathbf{F}'=\mathbf{F}-\frac{1}{Re}\mathbf{R},\qquad \mathbf{G}'=\mathbf{G}-\frac{1}{Re}\mathbf{S}.$$

The integral form of equation (1) can be written as

$$\frac{\partial}{\partial t}\int_Q \mathbf{U}\mathrm{d}Q+\int_S \mathbf{P}\cdot\mathbf{n}\mathrm{d}S=0, \tag{2}$$

where Q is the volume bounded by the surface S and $\mathbf{n}$ is the outward pointing unit vector normal to the surface. The area of S is $S=\sqrt{(S_x^2+S_y^2)}$. Let $S'_x=S_x/S$ and $S'_y=S_y/S$; so $\mathbf{n}=S'_x\mathbf{i}_x+S'_y\mathbf{i}_y$.

For Navier–Stokes equations there are viscous flux terms $\mathbf{R}$ and $\mathbf{S}$ with x- and y-derivatives. A local co-ordinate rotation is needed to align the directions of partial derivatives with the lines passing through the central points of adjacent cells.

FLUX SPLITTING

In this study upwind differencing is implemented by splitting the convective fluxes **F** and **G** into positive and negative contributions. Van Leer's flux vector splitting technique is chosen since it is continuously differentiable through sign changes of the eigenvalues and it leads to normal- shock profiles with only one or two transition zones.

The splitting of **F** is given by

$$\mathbf{F}(\mathbf{U}) = \mathbf{F}^+(\mathbf{U}) + \mathbf{F}^-(\mathbf{U}).$$

For subsonic $|M_x| < 1$,

$$\mathbf{F}^\pm = \begin{bmatrix} f_{\mathrm{m}}^\pm \\ f_{\mathrm{m}}^\pm[(\gamma-1)u \pm 2c]/\gamma \\ f_{\mathrm{m}}^\pm v \\ f_m^\pm\{[(\gamma-1)u \pm 2c]^2/[2(\gamma^2-1)] + v^2/2\} \end{bmatrix},$$

where $f_{\mathrm{m}}^\pm = \pm\rho c[\frac{1}{2}(M_x \pm 1)]^2$, and for supersonic $|M_x| \geqslant 1$,

$$\begin{aligned} &\mathbf{F}^+ = \mathbf{F}, \qquad && \mathbf{F}^- = 0, \qquad && M_x \geqslant 1, \\ &\mathbf{F}^- = \mathbf{F}, && \mathbf{F}^+ = \mathbf{0}, && M_x \leqslant -1, \end{aligned}$$

where c is the local sound speed and M_x is the local Mach number based on u. The splitting of **G** in terms of $M_y = v/c$ follows similarly. The Jacobian matrices of the positive split flux contributions have eigenvalues $\lambda \geqslant 0$ and those of the nagative ones have eigenvalues $\lambda \leqslant 0$. Since the time-dependent Euler equations remain hyperbolic in the subsonic and supersonic regions, the signs of the eigenvalues will determine the directions of information travel. From a one-dimensional characteristic analysis, the positive information goes downstream and the negative information goes upstream (see Figure 1).

The flux crossing an interface of two adjacent cells is the normal component of the vector **P** in equation (2). Let P' be the normal component passing through the unit interface; then

$$P' = \mathbf{P}\cdot\mathbf{n} = (\mathbf{F}\cdot\mathbf{S}'_x + \mathbf{G}\cdot\mathbf{S}'_y) - \frac{\alpha}{Re}(\mathbf{R}\cdot\mathbf{S}'_x + \mathbf{S}\cdot\mathbf{S}'_y).$$

The inviscid part of the flux $\mathbf{Z} = \mathbf{F}S'_x + \mathbf{G}S'_y$ should be split. So **Z** is first transformed into $\bar{\mathbf{Z}}$ by a local rotation matrix **T**:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & S'_x & S'_y & 0 \\ 0 & -S'_y & S'_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \bar{\mathbf{Z}} = \mathbf{T}\mathbf{Z} = \begin{bmatrix} \rho\bar{u} \\ \rho\bar{u}^2 + p \\ \rho\bar{u}\bar{v} \\ (e+p)\bar{u} \end{bmatrix}.$$

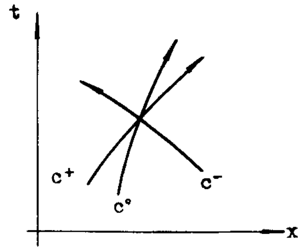

Figure 1. Pathline of information travel

The rotated velocity components $\bar{u}$ and $\bar{v}$ are normal and parallel to the interface respectively. The transformed flux $\bar{\mathbf{Z}}$ is of the same form as the Cartesian flux vector and thus can be split using Van Leer's splitting:

$$\bar{\mathbf{Z}} = \bar{\mathbf{Z}}^+ + \bar{\mathbf{Z}}^-.$$

Then

$$\mathbf{Z} = \mathbf{T}^{-1}\bar{\mathbf{Z}} = \mathbf{T}^{-1}(\bar{\mathbf{Z}}^+ + \bar{\mathbf{Z}}^-) = \mathbf{Z}^+ + \mathbf{Z}^-.$$

SPATIAL DISCRETIZATION

To discretize equation (2), upwind differencing is used for the convective and pressure terms and central differencing for the shear stress and heat flux terms. According to the direction of information travel (Figure 1), the positive contribution crossing surface $i + 1/2$ (Figure 2) should be determined by upstream points and the negative contribution by downstream points. MUSCL-type flux differencing[2] is used. We first extrapolate the centre-point values of $\mathbf{U}$ toward the interfaces and then obtain $\mathbf{Z}^+$ and $\mathbf{Z}^-$.

$$\mathbf{Z}_{i+\frac{1}{2},j} = \mathbf{Z}^+(U^-_{i+\frac{1}{2},j}) + \mathbf{Z}^-(\mathbf{U}^+_{i+\frac{1}{2},j}).$$

The conserved state variables on the upwind side of the interface are obtained by an upwind-biased interpolation.

$$\mathbf{U}^-_{i+\frac{1}{2},j} = \mathbf{U}_{i,j} + \tfrac{1}{4}\phi[(1-k)\nabla + (1+k)\Delta]\mathbf{U}_{i,j},$$

$$\mathbf{U}^+_{i+\frac{1}{2},j} = \mathbf{U}_{i+1,j} - \tfrac{1}{4}\phi[(1+k)\nabla + (1-k)\Delta]\mathbf{U}_{i+1,j}.$$

The symbols Δ and ∇ denote the forward and backward difference operators:

$$\Delta\mathbf{U}_{i,j} = \mathbf{U}_{i+1,j} - \mathbf{U}_{i,j}, \qquad \nabla U_{i,j} = \mathbf{U}_{i,j} - \mathbf{U}_{i-1,j}.$$

The switch ϕ is zero for first-order differencing and unity for higher-order differencing; $k = -1$ corresponds to the fully upwind second-order scheme and $k = 1/3$ to the upwind-biased third-order scheme.

Application of the implicit scheme in delta form and time linearization to the conservation laws given by equation (2) yields

$$[\mathbf{I}/\Delta t + \mathrm{d}\mathbf{R}/\mathrm{d}\mathbf{U}]\delta\mathbf{U}^{n+1} = \mathbf{R}^n, \tag{3}$$

where $\delta\mathbf{U}^{n+1} = \mathbf{U}^{n+1} - \mathbf{U}^n$, $\Delta t^n = t^{n+1} - t^n$ and n is the iteration index. $\mathbf{R}^n$ is the steady state

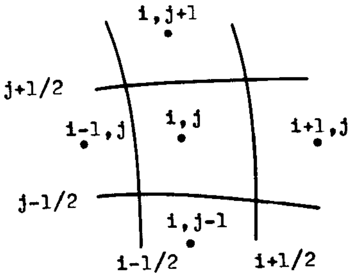

Figure 2. Notation for cell i, j. Cell boundaried at $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$; State variables at i, j

residual evaluated at time level n for the Euler or full Navier–Stokes equations.

$$\mathbf{R} = -\int_S \left[\left(\mathbf{F} - \frac{\alpha}{Re}\mathbf{R} \right) S'_x + \left(\mathbf{G} - \frac{\alpha}{Re}\mathbf{S} \right) S'_y \right] \mathbf{d}S.$$

In matrix notation equation (3) can be expressed as

$$\mathbf{M}\,\delta\mathbf{U} = \mathbf{R} \tag{4}$$

where $\mathbf{M}$ is a large, banded, block coefficient matrix with block size of four. A line Gaus–Seidel iterative procedure is used for inversion. To reduce the computational work, first-order upwind differencing, which only leads to a block pentadiagonal matrix, is used on the left-hand side and third-order upwind-biased differencing is used on the right-hand side. The accuracy of the converged solution is controlled by the right-hand side. Then equation (4) can be rewritten as

$$\mathbf{B}\delta\mathbf{U}^{n+1}_{i,j+1} + \mathbf{A}\delta\mathbf{U}^{n+1}_{i,j} + \mathbf{C}\delta\mathbf{U}^{n+1}_{i,j-1} + \mathbf{E}\delta\mathbf{U}^{n+1}_{i-1,j} + \mathbf{D}\delta\mathbf{U}^{n+1}_{i+1,j} = \mathbf{R}^{n}_{i,j}, \tag{5}$$

where the coefficients $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{E}$ are 4×4 block matrices. The coefficient matrix on the left-hand side is diagonally dominant and two iterations are sufficient at each time step. The relaxation algorithm is implemented by alternately sweeping in the backward and forward streamwise directions. Alternating the direction of the implicit line inversion has been proven to be beneficial to the full Navier–Stokes equations with highly stretched grids. It is also consistent with the nature of transonic and viscous separated flow.

WALL BOUNDARY CONDITIONS

From a truncation error analysis, the numerical dissipation of the third-order upwind-biased differencing is small at the interior points of a flow field.[5] But large numerical dissipation occurs with Van Leer's flux vector splitting when a steady contact discontinuity such as a solid wall is encountered. The following analysis indicates that upwind dissipation is related to the treatment of wall boundary conditions and is not unavoidable.

For finite volume schemes, the reflection condition has often been used to treat the flow on a wall boundary. A row of dummy cells located just across the wall boundary is constructed which is of the same size and shape as the row of cells just inside the boundary, as shown in Figure 3. For no-slip wall boundary conditions,

$$u_1 = -u_2, \qquad v_1 = -v_2, \qquad p_1 = p_2, \qquad T_1 = T_2.$$

Obviously the correct flux crossing the wall will be obtained by using a central differencing scheme. If the wall flux is determined by flux vector splitting with reflection conditions,

$$\mathbf{G}_w = \mathbf{G}_1^+ + \mathbf{G}_2^-,$$

the flux obtained will be

$$\mathbf{G}_w = [0,\ 2g_1^+ u_1,\ dp_1,\ 0]^{\mathrm{T}},$$

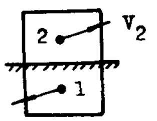

Figure 3. Reflection condition for no-slip wall boundary

where

$$d = (1 + M_y)^2[1 + M_y(\gamma - 1)/2] \not\equiv 1, \qquad 2g_1^+ u_1 \not\equiv 0.$$

Thus it cannot be consistent with the correct value

$$\mathbf{G} = [0, 0, p_w, 0]^T.$$

Similarly, for a tangent condition on the wall the flux obtained will be

$$\mathbf{G} = [0, 0, dp_1, 0]^T$$

which is still not correct. It is the error that causes the too large numerical dissipation.

To avoid the above errors, an appropriate wall boundary condition treatment has been suggested herein. For inviscid flow $\mathbf{P}_w$ is first-order extrapolated from the inner point. On the implicit left-hand side

$$\delta \mathbf{G}_w^{n+1} = (\partial \mathbf{G}_w/\partial \mathbf{U})\partial \mathbf{U}^{n+1},$$

where $\partial \mathbf{G}_w/\partial \mathbf{U}$ is the Jacobian matrix. The boundary conditions are incorporated into the matrix equation (5) which leads to increased rates of convergence. For viscous flow the solution points of the cells just inside the wall boundary are put at the wall surface instead of the cell centre, as shown in Figure 4. The slip or no-slip conditions can be specified at point 1. In the present computation no-slip conditions are set as

$$u_1 = v_1 = 0, \qquad p_1 = p_2 \text{ as } \partial p/\partial n = 0, \quad T_1 = T_2$$

for the adiabatic wall. Even though the wall boundary conditions for viscous flow are treated explicitly, the high convergence rate can still be reached because of the accurately imposed no-slip conditions.

The wall boundary treatment given above can be used not only for the case of flux splitting but also for general finite volume schemes.

GRID GENERATION AND TIME STEPPING

The computational grid is constructed algebraically and has variable spacing in both x- and y-directions. Grid points are clustered near the wall to resolve large gradients in the boundary layer.

The stretching function in the y-direction is

$$y(j) = \eta(j)(y_u - y_l) + y_l, \quad j = 0, 1, 2, \ldots, M,$$
$$\eta(j) = (j/M)^a,$$

where y_u and y_l are the y-co-ordinates of the upper and lower boundary respectively, M is the number of grid points in the transverse direction and a is a constant chosen to control the stretch of the mesh. This is a simple approach to get a high-quality mesh.

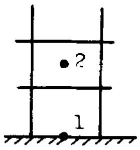

Figure 4. Wall boundary condition for viscous flow

For inviscid flow the time step is determined by

$$\Delta t = d/(|u| + |v| + 2a),$$

where d is a constant and a is the local sound speed. Δt here is independent of the mesh point spacing Δx or Δy and is relatively uniform in the entire flow field. This is very efficient for inviscid flow computation.

For viscous flow the time step is determined by the usual formulation

$$\Delta t = \mathrm{CFL} \cdot \Delta y/(|v| + a).$$

This is very efficient for the highly stretched grid of viscous flow. Different sizes of time steps varying with the local grid spacing can be obtained.

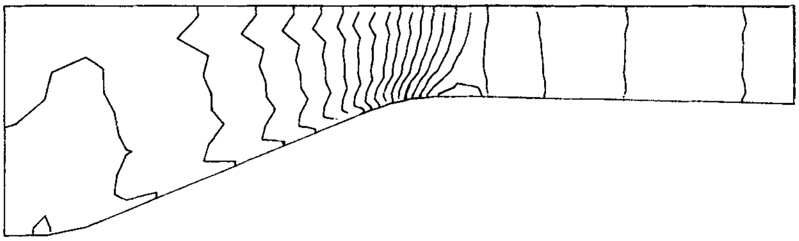

Figure 5. Pressure contours of the calculated inviscid flow field for the converging–diverging nozzle; 25 × 11 H-mesh

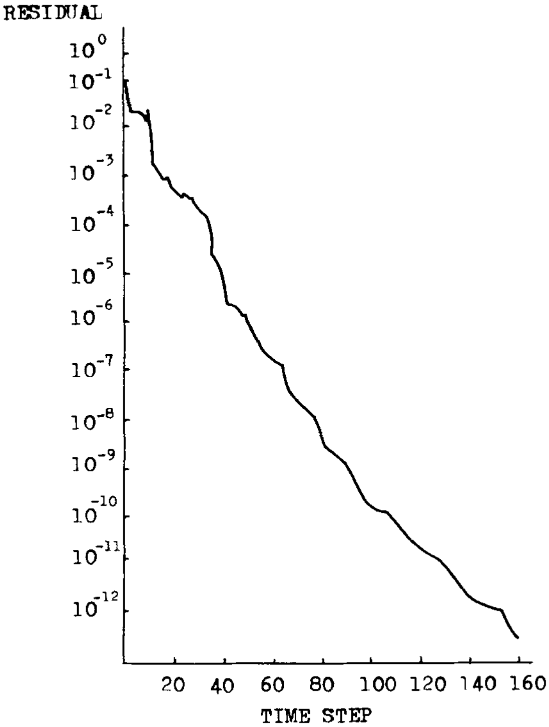

Figure 6. Convergence history of the inviscid computation for the converging–diverging nozzle

RESULTS AND DISCUSSION

The numerical procedures described were applied to a series of compressible inviscid and viscous flow problems.

Inviscid computational results

The first case is a symmetric converging–diverging nozzle. Figure 5 shows the pressure contours of the computed flow field. Initially the flow was at rest with pressure and temperature set everywhere to their total values at the nozzle entrance. The CFL number on a stretched non-uniform 25 × 11 mesh reached 90. Figure 6 shows the rapid convergence rate. The L_2-norm of the residual was reduced by twelve orders of magnitude in less than 160 iterations. Figure 7 is the computed pressure distribution which compares well with the experimental results.[6]

The second case is an inlet-diffuser which involves a normal shock in the flow field. Figure 8 shows the Mach number contours of the calculated flow field. The lower wall pressure distribu-

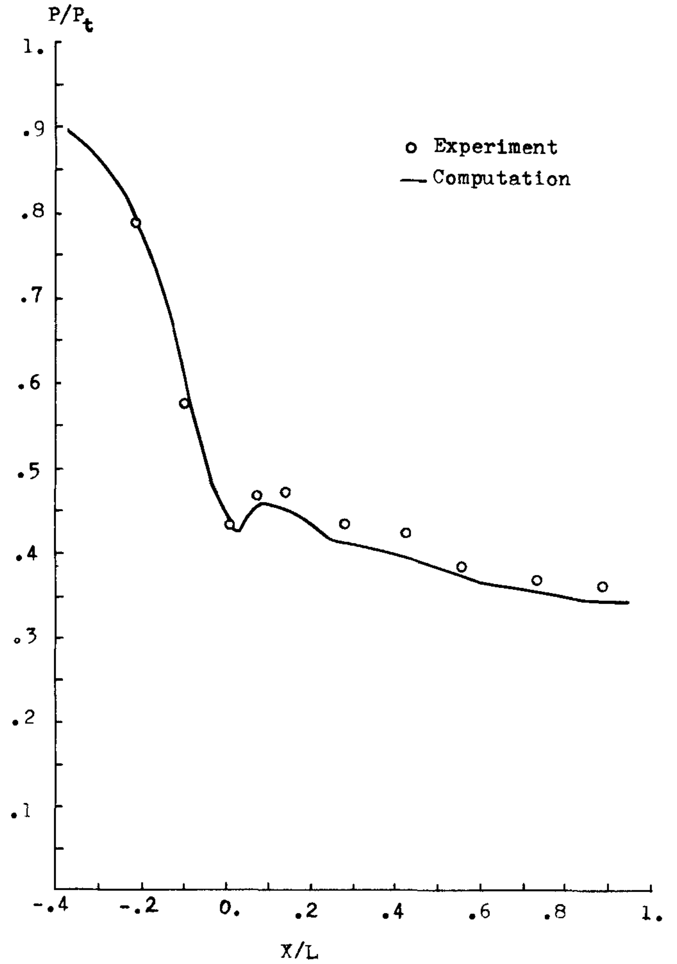

Figure 7. Surface pressure distribution of the converging–diverging nozzle

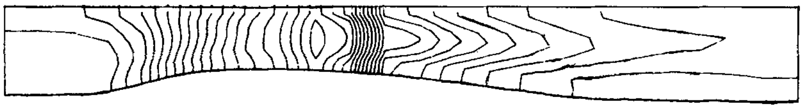

Figure 8. Mach number contours of the calculated inviscid flow field for the inlet-diffuser; 25 × 11 H-mesh

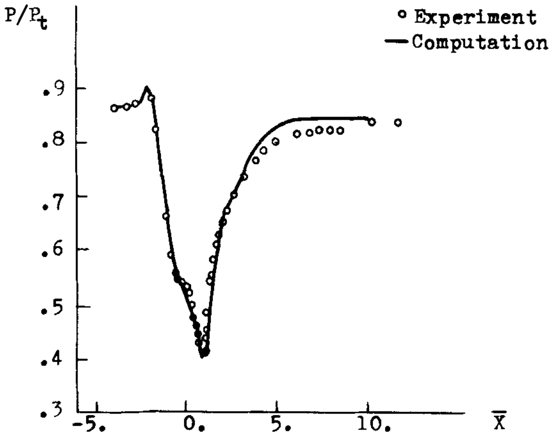

Figure 9. Lower wall pressure distribution of the inlet-diffuser

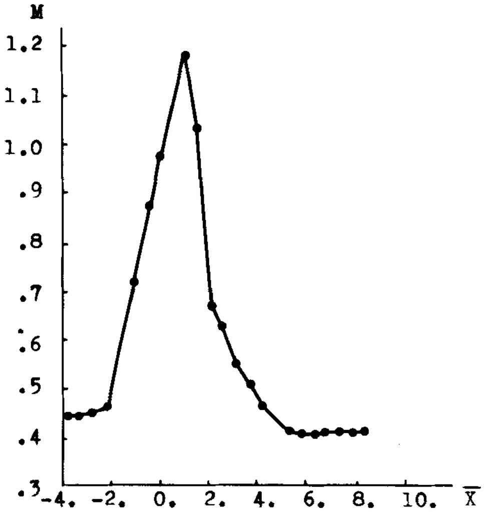

Figure 10. Mach number distribution along the lower wall of the inlet-diffuser

tion is shown in Figure 9. It compares favourably with the experiment given in Reference 7. Good shock location is obtained. The residual was reduced by four orders of magnitude in 61 iterations. It took only 120 s CPU time on an IBM 4381 computer. Figure 10 indicates that there are only two points within the shock transition area.

The third case is a supersonic internal flow which involves an oblique shock and expanion waves with a wedge angle of 5° and entrance Mach number of 3. The reflecting shock intersects with the expansion waves. Figure 11 shows the pressure contours. For the fully supersonic flow, the coefficient matrix **D** in equation (5) is zero everywhere. The information travels only in the forward direction. In each time step just one forward sweep is needed for the global iteration. Figure 12 presents the pressure distribution of the lower surface. It compares well with the results in Reference 8. The cross symbols show that both the shock and the expansion waves have just one transition zone.

Viscous computational results

The first case of viscous flow is again the two-dimensional symmetric converging–diverging nozzle. Figure 13 presents the 53 × 16 H-mesh which is extremely stretched in the transverse direction and non-uniform in the streamwise direction. In the computation the CFL number in the y-direction was kept constant at 210 and the maximum CFL number in the x-direction reached $7{\cdot}2 \times 10^4$. The residual was reduced by four orders of magnitude in 81 iterations. Figure 14 shows the pressure distribution which compares well with the measured data. In the region just downstream of the throat ($x = 0$) the nozzle diverges rapidly and the flow shows a tendency to separate which is caused by the local pressure rising. The disturbance travels upstream along the boundary layer and causes a small undulation. Figure 15 shows the Mach number contours of the viscous flow field.

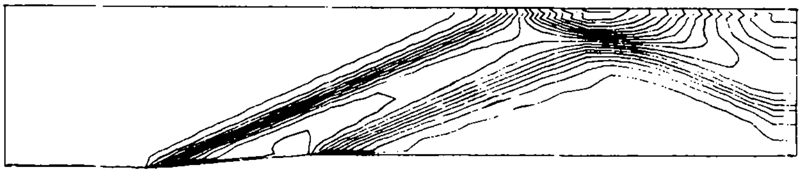

Figure 11. Pressure contours of the calculated inviscid flow field for the supersonic wedge duct; 36 × 27 H-mesh

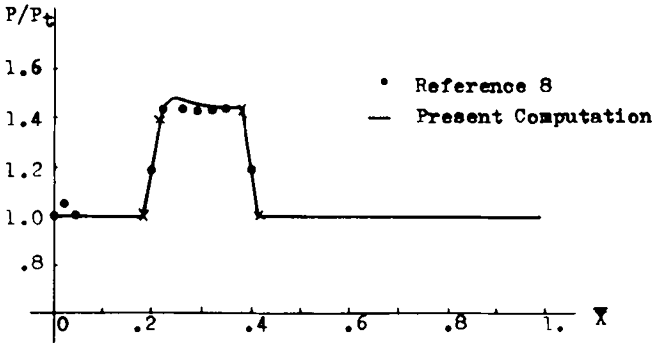

Figure 12. Lower wall pressure distribution of the wedge duct

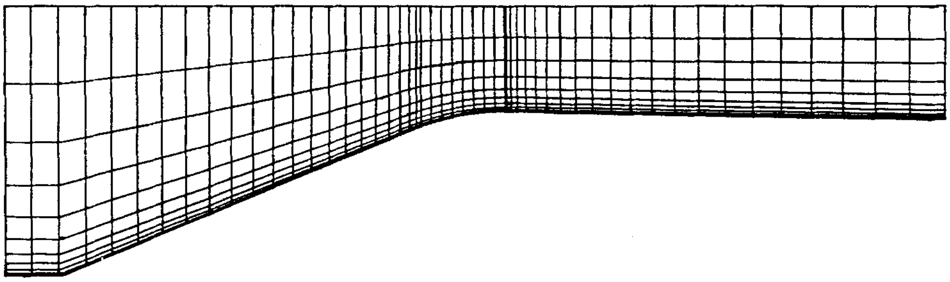

Figure 13. Computational grid for the converging–diverging nozzle in the viscous computation; 53 × 16 H-mesh

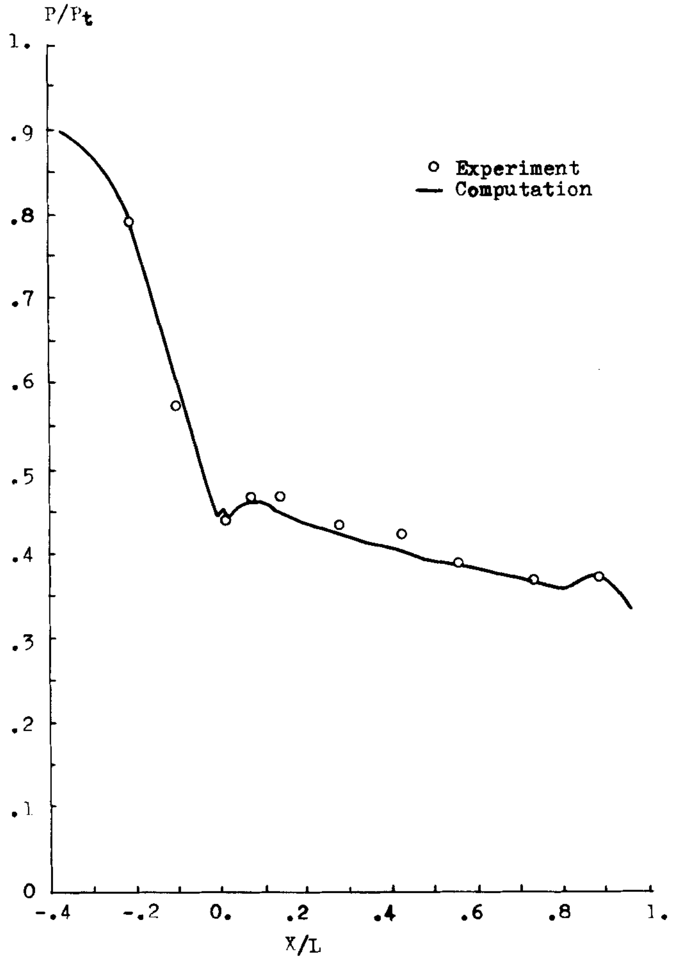

Figure 14. Lower wall pressure distribution of the converging–diverging nozzle calculated by the Navier–Stokes equations

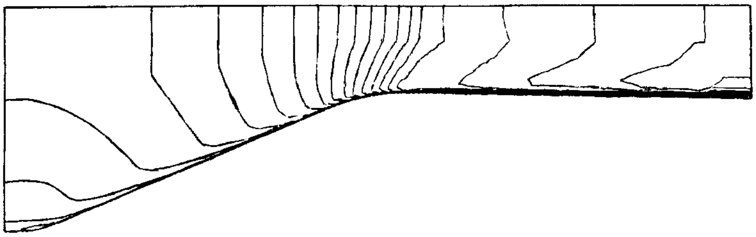

Figure 15. Match number contours of the calculated viscous flow field for the converging–diverging nozzle

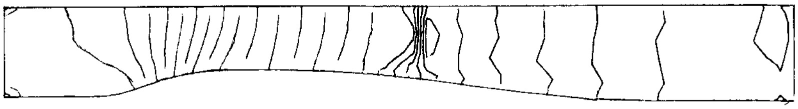

Figure 16. Pressure contours of the calculated viscous flow field for the inlet-diffuser; 60 × 17 H-mesh

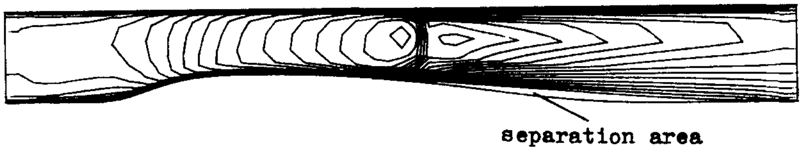

Figure 17. Mach number contours of the calculated viscous flow field for the inlet-diffuser

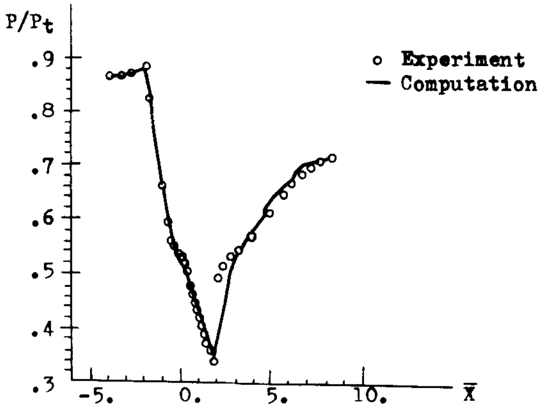

Figure 18. Lower wall pressure distribution of the inlet-diffuser

The second case of viscous computation is the inlet-diffuser. There is a normal shock which interacts with the laminar boundary layer. Figure 16 shows the pressure contours which feature the λ-shock wave produced by the shock wave/boundary layer interaction. The flow separation is shown clearly in Figure 17. Figure 18 shows that the pressure distribution and the shock location obtained agree quite well with the experimental results.[7]

For the two viscous cases above, the maximum aspect ratio of the grid on the wall reached $5{\cdot}79 \times 10^4$. This far exceeds the limit of other algorithms whose aspect ratio of the grid is usually less than 10^3. This is attributed to the high accuracy and good conservation of the scheme.

CONCLUSIONS

An efficient algorithm based on flux splitting for steady state solutions to the compressible Euler and Navier–Stokes equations has been described. The original Van Leer flux vector splitting has been extended to body-fitted co-ordinates in the physical domain so that the flow field computation can be implemented with no transformation. Different spatial discretizations on the implicit and explicit sides are used to reduce the computational work and to obtain solutions with higher-order accuracy. A line Gauss–Seidel relaxation approach is used to obtain very large time steps. It has been verified that the reflection wall boundary conditions cannot be used with flux vector splitting. The treatment of wall boundary conditions presented in the paper avoids the too large numerical dissipation and leads to increased rates of convergence. The algorithm is suitable for highly stretched and twisted grids. Computed solutions agree favourably with experiments for both inviscid and viscous flows, including the case of shock-induced boundary layer separation.

REFERENCES

1. B. Van Leer, 'Flux-vector splitting for the Euler equations', *Lecture Note in Physics, Vol. 170*, 1982.
2. W. K. Anderson, J. L. Thomas and B. Van Leer, 'A comparison of finite volume flux vector splitting for the Euler equations', *AIAA 85-0122*, 1985.
3. S.R. Chakravarthy, 'Relaxation method for unfactored implicit scheme', *AIAA 84-0165*, 1984.
4. B. Van Leer, J. L. Thomas and P. L. Roe, 'A comparison of numerical flux formulas for the Euler and Navier–Stokes equations', *AIAA 87-1104*, 1987.
5. J. L. Thomas and R. W. Walters, 'Upwind relaxation algorithm for the Navier–Stokes equations', *AIAA 85-1501*, 1985.
6. M. L. Mason, L. E. Putnam and R. J. Re, 'The effect of throat contouring on two-dimensional converging– diverging nozzles at static conditions', *NASA TP-1704*, 1980.
7. T. J. Bogar, M. Sajben and J. C. Krontil, 'Characteristic frequency and length scales in transonic diffuser flow oscillations', *AIAA 81-1291*, 1981.
8. R. W. Walters and H. A. Hassan, 'A strongly implicit procedure for the compressible Navier–Stokes equations', *AIAA 84-0424*, 1984.
9. J. L. Thomas, B. Van Leer and R. W. Walters, 'Implicit flux-split schemes for the Euler equations', *AIAA 85-1680*, 1985.
10. W. A. Mulder and B. Van Leer, 'Experiments with implicit upwind methods for the Euler equations', *J. Comput. Phys.*, **59**, 232–246 (1985).
11. J. L. Thomas, R. W. Walters, B. Van Leer and C. L. Rumsey, 'An implicit flux-split algorithm for the compressible Navier–Stokes equations', *ICNMFD paper*, Beijing, July 1986.
12. R. W. Walters, J. L. Thomas and B. Van Leer, 'An implicit flux-split algorithms for the compressible Euler and Navier–Stokes equations', *ICNMFD paper*, Beijing, July 1986.
13. D. Hanel, R. Schwane and G. Seider, 'On the accuracy of upwind scheme for the solutions of the Navier–Stokes equations', *AIAA 87-1105*, 1987.
14. R. W. Walters and R. L. Dwoyer, 'An efficient iteration strategy for the solution of the Euler equations', *AIAA 85-1529*, 1985.
15. W. MacCormack, 'Current status of numerical solutions of the Navier–Stokes equations', *AIAA 85-0032*, 1985.