INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 5, 483–500 (1985)

COMPUTATION OF THREE-DIMENSIONAL FLOW USING THE EULER EQUATIONS AND A MULTIPLE-GRID SCHEME

CHARLES KOECK

MATRA, 37, avenue Louis Breguet, 78146 Velizy, Villacoublay, France

SUMMARY

The unsteady Euler equations are numerically solved using the finite volume one-step scheme recently developed by Ron-Ho Ni. The multiple-grid procedure of Ni is also implemented. The flows are assumed to be homo-enthalpic; the energy equation is eliminated and the static pressure is determined by the steady Bernoulli equation; a local time-step technique is used. Inflow and outflow boundaries are treated with the compatibility relations method of ONERA. The efficiency of the multiple-grid scheme is demonstrated by a two-dimensional calculation (transonic flow past the NACA 12 aerofoil) and also by a three-dimensional one (transonic lifting flow past the M6 wing). The third application presented shows the ability of the method to compute the vortical flow around a delta wing with leading-edge separation. No condition is applied at the leading-edge; the vortex sheets are captured in the same sense as shock waves. Results indicate that the Euler equations method is well suited for the prediction of flows with shock waves and contact discontinuities, the multiple-grid procedure allowing a substantial reduction of the computational time.

KEY WORDS Three-dimensional Vortex flow Pseudo-unsteady Euler Equations Finite Volume Multiple Grid

INTRODUCTION

Missile aerodynamics at high angles of attack is concerned mostly with non-linear phenomena. Such non-linearities include mainly shock waves and vortex sheets. The Euler equations method is well suited for predicting those phenomena. It yields solutions with rotational flows, and gives the exact jump relations at discontinuities. The principal difficulty is that the Euler equations cannot model flow separating from a smooth surface; in particular, the formation of vortices above slender bodies at angle of attack cannot be predicted by the Euler equations without auxiliary conditions. When the flow separation occurs at a sharp edge, the starting position of the vortex sheet is known and the Euler method can yield realistic solutions, due to its ability to capture contact discontinuities in the same sense as shock waves. However, it is more difficult to have an accurate capture of a vortex sheet than a shock wave. The artificial dissipation introduced by the numerical scheme provokes a vorticity diffusion which may be greater than the diffusion related to the physical viscosity. The result is that the numerical solutions give a global modelization, rather than a precise description, of the vortex sheets.

Owing to their simplicity in numerical implementation, the Euler equations are often solved using time-marching explicit procedures. Both formulations with finite differences or finite volumes yield accurate solutions, the second one being more suitable for three-dimensional calculations with mesh singularities. But, because of the stability criterion inherent in explicit schemes, a great number of time steps is required to reach a steady state, with, by consequence,

0271–2091/85/050483–18$01.80

Received 3 January 1984
Revised 19 December 1984

a very large computational time. Substantial improvements in convergence acceleration have been achieved by Jameson *et al.*,[1–3] using a multi-stage finite-volume scheme, with Runge–Kutta time stepping. Rizzi and Eriksson[4–6] have also presented computations with a three-stage finite-volume scheme. Another way to reduce computational time is to implement implicit schemes, as done for example by Lerat, Sides and Daru.[7]

If only the steady state is of interest, a very efficient means is the multiple-grid technique. The basic idea behind multiple-grid methods is to model the conservation laws on a fine grid, in order to ensure the accuracy of the steady solution, while the propagation of the unsteady waves is modelled on coaser grids, allowing large time increments, and reduced computational work.

Ron-Ho Ni[8] has presented an explicit scheme with multiple-grid convergence acceleration for solving the time-dependent Euler equations. The basic scheme is a one-step Lax–Wendroff scheme, with finite-volume formulation; it is second order accurate in time and space. This scheme, like the coarse-grid procedure, is based on distribution formulae. Computations using Ni's coarse-grid scheme have already been presented by Johnson,[9,10] for the Euler and Navier–Stockes equations. An alternative multigrid formulation has also been developed by Jameson.[77]

METHOD

Governing equations

The Euler equations are written in a pseudo-unsteady conservative form with constant total enthalpy:

$$\mathbf{f}_t + \operatorname{div} \mathbf{F}(\mathbf{f}) = 0,$$

$$\mathbf{f} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix}, \quad \mathbf{F}(\mathbf{f}) = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho v u & \rho w u \\ \rho u v & \rho v^2 + p & \rho w v \\ \rho u w & \rho v w & \rho w^2 + p \end{bmatrix}. \tag{1}$$

Here, ρ is the density, u, v, w are the velocity components and p is the static pressure given by the Bernoulli relation:

$$p = \frac{\gamma - 1}{\gamma} \rho \left(Hi - \frac{u^2 + v^2 + w^2}{2} \right). \tag{2}$$

Hi denotes the total enthalpy and γ the ratio of specific heats. Uniformity of the total enthalpy being true only for steady states, the transient solution has no physical meaning.

Mathematical properties of this pseudo-unsteady system have been studied at ONERA by Viviand and Veuillot[12] and Brochet,[13] and will be used to implement the boundary conditions and also to compute the time increments.

Basic finite-volume scheme

At each point of the grid, the corrections $\delta \mathbf{f}_{ijk} = \mathbf{f}_{ijk}^{n+1} - \mathbf{f}_{ijk}^{n}$ are evaluated by adding the eight contributions coming from the nearest-neighbouring cells having the node in common:

$$\delta \mathbf{f}_{ijk} = \sum_{c=1}^{8} (\delta \mathbf{f}_{ijk})_c \tag{3}$$

where subscript c denote values related to a cell.

Each contribution $(\delta \mathbf{f}_{ijk})_c$ is calculated by means of distribution formulae which involve quantities depending only on the considered cell. The establishment of these distribution formulae is now presented.

The corrections $\delta \mathbf{f}$ are given by the second order approximation

$$\delta \mathbf{f} = \Delta t \mathbf{f}_t + \frac{\Delta t^2}{2} \mathbf{f}_{tt}. \tag{4}$$

Time derivatives are expressed as space derivatives:

$$\mathbf{f}_t = -\operatorname{div} \mathbf{F} \tag{5}$$

$$\mathbf{f}_{tt} = -\operatorname{div} \mathbf{F}_t, \quad \text{with } \mathbf{F}_t = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{f}}\right) \cdot \mathbf{f}_t. \tag{6}$$

Equations (5) and (6) may be written in integral form for any volume V, bounded by ∂V; using the mean value theorem results in

$$\mathbf{f}_t = -\frac{1}{V} \int_{\partial V} \mathbf{F} \cdot \mathbf{n} \, \mathrm{d}s \tag{7}$$

$$\mathbf{f}_{tt} = -\frac{1}{V} \int_{\partial V} \mathbf{F}_t \cdot \mathbf{n} \, \mathrm{d}s \tag{8}$$

where $\mathbf{n}$ is the outward unit normal to ∂V. Here $\mathbf{f}_t$ and $\mathbf{f}_{tt}$ are associated with some point interior to the volume (taken to be the volume centre for practical purposes).

Equation (7) is discretized by a finite-volume approximation applied to the control volume defined by one mesh cell. The result is a value of $\mathbf{f}_t$ at the centre of the cell:

$$(\mathbf{f}_t)_c = \frac{-1}{V_c} \sum_{l=i,j,k} \mathbf{F}_l^+ \cdot \mathbf{S}_l^+ + \mathbf{F}_l^- \cdot \mathbf{S}_l^-. \tag{9}$$

Here V_c is the volume of the cell; $\mathbf{S}_l^+$ and $\mathbf{S}_l^-$ are the outward area vectors of opposite faces of index l (Figure 1).

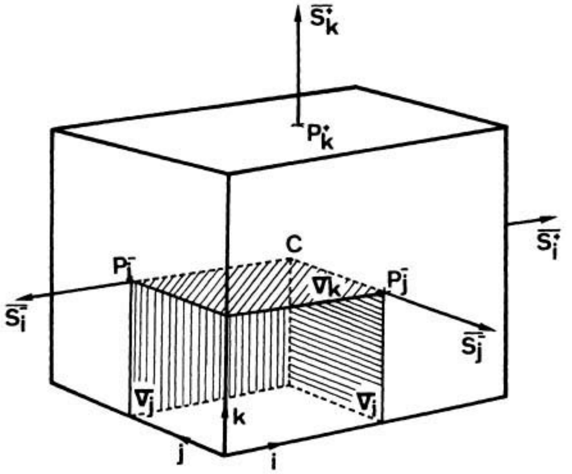


Figure 1. Nomenclature of the computational mesh

The values of $\mathbf{f}_t$ at mesh nodes are then determined by the relation:

$$(\mathbf{f}_t)_{ijk} = \sum_{c=1}^{8} V_c(\mathbf{f}_t)_c / \sum_{c=1}^{8} V_c. \tag{10}$$

The sums involve the eight cells sharing the node ijk.

The control volume associated with the finite-volume approximation of equation (8) is a translated cell, whose vertices are the centres of the eight cells defined above. Inside one cell, the boundary of this volume is composed of three surfaces $\boldsymbol{\sigma}_l$, $l = i, j, k$, given by

$$\boldsymbol{\sigma}_l = \mathbf{S}_l/4, \quad \text{where} \quad \mathbf{S}_l = \varepsilon_l(\mathbf{S}_l^+ - \mathbf{S}_l^-).$$

The parameter ε_l is equal to $+1$ or -1, so that the surface $\boldsymbol{\sigma}_l$ is always facing outwards (Figure 1).

The finite-volume approximation of equation (8) reads then as follows:

$$(\mathbf{f}_{tt})_{ijk} = \left[\sum_{c=1}^{8} (\mathbf{F}_t)_c \cdot (\mathbf{S}_i + \mathbf{S}_j + \mathbf{S}_k)/4 \right] \bigg/ \left(\frac{1}{8} \sum_{c=1}^{8} V_c \right), \tag{11}$$

with

$$(\mathbf{F}_t)_c = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{f}} \right)_c \cdot (\mathbf{f}_t)_c.$$

Combining equations (10) and (11) in equation (4) gives the values of the corrections $\delta\mathbf{f}_{ijk}$ at the mesh nodes. The contribution coming from the cell (denoted by subscript c) of index ($i + 1/2$, $j + 1/2$, $k + 1/2$) can be easily extracted:

$$(\delta\mathbf{f}_{ijk})_c = \left[\Delta\mathbf{f}_c - \frac{\Delta t}{V_c}(\Delta\mathbf{F}_{ic} + \Delta\mathbf{F}_{jc} + \Delta\mathbf{F}_{kc}) \right] \left(V_c \bigg/ \sum_{c=1}^{8} V_c \right), \tag{12}$$

where

$$\begin{aligned} \Delta\mathbf{f}_c &= \Delta t(\mathbf{f}_t)_c \\ \Delta\mathbf{F}_{lc} &= \Delta\mathbf{F}_c \cdot \mathbf{S}_l \quad l = i, j, k \\ \Delta\mathbf{F}_c &= (\partial\mathbf{F}/\partial\mathbf{f})_c \cdot \Delta\mathbf{f}_c. \end{aligned}$$

Conversely, first order changes occurring in a cell are distributed to its vertices by means of eight distribution formulae:

$$(\delta\mathbf{f}_s)_c = \left[\Delta\mathbf{f}_c + \frac{\Delta t}{V_c}(\varepsilon_i\Delta\mathbf{F}_{ic} + \varepsilon_j\Delta\mathbf{F}_{jc} + \varepsilon_k\Delta\mathbf{F}_{kc}) \right] \left(V_c \bigg/ \sum_{c=1}^{8} V_c \right). \tag{13}$$

The subscript s denotes the considered vertex, and the parameters ε are defined by the following relations:

$\varepsilon_i = +1$ for the four vertices of index $i + 1$,
$\varepsilon_i = -1$ for the four vertices of index i,

and so on for ε_j and ε_k.

In practice, area vectors $\mathbf{S}_l^+$ are computed as half the cross product of the diagonal vectors; the volume of a cell is computed in the following way:

$$V_c = \sum_{l=i,j,k} (\overline{\mathrm{CP}_l^+} \cdot \mathbf{S}_l^+ + \overline{\mathrm{CP}_l^-} \cdot \mathbf{S}_l^-)/3, \tag{14}$$

where C is the centre of the cell, and P_l the centre of a face (Figure 1). The coordinates of these points are obtained by averaging the vertex coordinates. If the faces are not planar, the computations of $\mathbf{S}_l$ and V_c are not exact, but still compatible.

The flux vectors $\mathbf{F}_l$ are computed for each face of a cell by simple averaging of the four flux vectors given at the corners of the face. The time derivative of the flux vector, $\Delta\mathbf{F}$, is not calculated through the costly computation of the Jacobian matrix $(\partial\mathbf{F}/\partial\mathbf{f})$ but more simply by the following relations:

$$\Delta\mathbf{f} = \begin{bmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta\rho w \end{bmatrix}, \quad \Delta\bar{\mathbf{F}} = \begin{bmatrix} \Delta\rho u & \Delta\rho v & \Delta\rho w \\ \Delta\rho u^2 + \Delta p & \Delta\rho v u & \Delta\rho w u \\ \Delta\rho u v & \Delta\rho v^2 + \Delta p & \Delta\rho w v \\ \Delta\rho u w & \Delta\rho v w & \Delta\rho w^2 + \Delta p \end{bmatrix}, \tag{15}$$

where

$$\Delta\rho u^2 = 2u\Delta\rho u - u^2\Delta\rho,$$
$$\Delta\rho u v = u\Delta\rho v + v\Delta\rho u - uv\Delta\rho,$$
$$\Delta p = \frac{\gamma-1}{\gamma}\left(Hi\Delta\rho - \frac{\Delta\rho u^2 + \Delta\rho v^2 + \Delta\rho w^2}{2}\right).$$

The components of the velocity, (u, v, w), are calculated at the cell centre using averaged values for ρ and $\rho\mathbf{V}$ which are given at the cell vertices at iteration n.

We may notice that this scheme is rigorously conservative when the time increments Δt are associated with the mesh nodes. If we wish to use the local time-step technique, the distribution formulae lead us to use time steps associated with the mesh cells; in this case, the scheme is no longer strictly conservative. The weighting of the distribution formulae by the volumes may also be suppressed; the distribution formulae become then

$$(\delta\mathbf{f}_s)_c = \frac{1}{8}\left[\Delta\mathbf{f}_c + \frac{\Delta t}{V_c}(\varepsilon_i\Delta\mathbf{F}_{ic} + \varepsilon_j\Delta\mathbf{F}_{jc} + \varepsilon_k\Delta\mathbf{F}_{kc})\right]. \tag{16}$$

If $\Delta t/V_c$ is constant, the scheme is again strictly conservative. In this paper, all the computations have been performed using equation (16) and the local time-step technique.

Time increment calculation

The time increment Δt is limited in magnitude by the Courant–Friedrichs–Lewy stability criterion. At each mesh cell, the maximum allowable time step is given by

$$\Delta t = \min_{l=i,j,k}(\Delta t_l), \tag{17}$$

with

$$\Delta t_l = \frac{V_c}{\frac{\gamma+1}{2\gamma}\mathbf{V}\cdot\mathbf{S}_l + \sqrt{\left[\left(\frac{\gamma-1}{2\gamma}\mathbf{V}\cdot\mathbf{S}_l\right)^2 + \frac{a^2}{\gamma}\mathbf{S}_l^2\right]}}.$$

$\mathbf{V}$ is the velocity vector at the cell center, and a the speed of sound.

Smoothing

Artificial viscosity is required for the computation of flow with large gradients (shocks mainly). A linear model of second order is used; it is simply composed of a Laplace operator ∇^2 discretized with 27 points, as the basic scheme. Smoothing is applied when updating the unknowns:

$$\mathbf{f}^{n+1}_{\text{smooth}} = \mathbf{f}^{n+1} + q_0\nabla^2(\mathbf{f}^n). \tag{18}$$

For 2D calculations, a dissipation model very similar to that of Jameson[7] is used. It is composed of a non-linear second-order term, and a linear fourth-order term. The non-linear term is written as (for the index i only):

$$D_i(\mathbf{f}) = \varepsilon_{i+1/2}(\mathbf{f}_{i+1} - \mathbf{f}_i) - \varepsilon_{i-1/2}(\mathbf{f}_i - \mathbf{f}_{i-1}),$$
$$\varepsilon_{i+1/2} = q_1 \sup_{l=i,i+1} (|\rho_{l+1} - 2\rho_l + \rho_{l-1}|/\rho_l).$$

Usual values for $q_0$and q_1 are 0·01 and 0·1.

Boundary conditions

Mesh nodes belonging to a boundary receive information coming only from four, two or even one cell. Since no extrapolation is used, second order accuracy in space may be lost. Complementary information is given by the boundary conditions.

Inflow and outflow boundaries. The compatibility relations technique developed at ONERA[12–14] is used.

At a supersonic inflow boundary, all the variables are set to the free-stream values; on the other hand, if the upstream flow is subsonic, three boundary conditions must be specified:

$$\begin{aligned}&\text{total pressure } p_i = p_{i\infty},\\ &\text{velocity direction } \mathbf{v}/v = (\mathbf{v}/v)_\infty.\end{aligned} \tag{19}$$

At a supersonic outflow boundary, no condition is required; and if the flow is subsonic, one condition is necessary; in 2D cases, we set the static pressure to its free-stream value, and in 3D cases, we use the following non-reflective condition:

$$p_t + \rho C^-(v_n)_t = 0$$

where v_n is the velocity component normal to the boundary, and C^- is given by:

$$C^- = \frac{\gamma-1}{2\gamma}v_n - \sqrt{\left[\left(\frac{\gamma-1}{2\gamma}v_n\right)^2 + \frac{a^2}{\gamma}\right]}. \tag{20}$$

Solid walls. Fluxes corresponding to a surface fitted with a solid wall are computed by taking into account only the pressure term. Then, at the end of the distribution formulae step, the velocity component normal to the wall is set to zero:

$$\mathbf{v}^{n+1} = \mathbf{v}^s - (\mathbf{v}^s\cdot\mathbf{N})\mathbf{N}. \tag{21}$$

$\mathbf{v}^s$ is the velocity given by the scheme, and $\mathbf{N}$ the unit normal to the wall.

Coordinate cuts. Treatment for these boundaries is very simple: contributions coming from cells located on either sides of the cut are summed to give the corrections at the nodes of the cut, so discretization continuity is recovered.

Symmetry planes. Symmetry planes may be considered as solid walls; an improved treatment has been used for the present computations, which consists of accounting for additional contributions defined by mirror conditions, thus, second order in space is maintained.

Multiple-grid procedure

The Euler equations are solved numerically on the coarse grid using an integration method similar to that of the fine grid. The main feature of the coarse-grid procedure is to approximate first order time derivatives by means of a complete time iteration of the basic scheme (instead of discretizing equation (5)). Second order time derivatives are then obtained by a coarse-grid discretization of equation (6), and, lastly, the corrections $\delta\mathbf{f}$ are calculated at the coarse-grid nodes using equation (4). Transfer of information between two grids is achieved by the restriction operator (fine to coarse) and the prolongation operator (coarse to fine).

General structure of the two-grid method. First, let us denote by h and H the typical mesh sizes of the fine and coarse grids; the basic integration method is represented by the operator $Q_h(Q_h$ includes the basic finite-volume scheme, artificial viscosity and the boundary conditions); the restriction and prolongation operators are noted by R_h^H and P_H^h.

The two-grid method reads as shown in Table I. Δt is the fine-grid time step; ω may be considered as an over-relaxation factor, which allows an improvement of the convergence. The stability of the complete algorithm has been studied by the author,[15] by means of a Fourier mode analysis. The resulting optimal values for ω and β are 1·7 and 1·4.

Numerical implementation of the multiple-grid procedure

Coarse grids

Each coarse grid is obtained from the previous one by deleting the mesh surfaces whose index is even, thus, we have $H = 2h$, and, in three-dimensional cases, the number of cells is divided by eight from a grid to the next coarser one. Specific mesh surfaces, such as boundaries or surfaces including leading or trailing edges, must be present in each grid.

Restriction operator R_h^{2h}

In order to have a coarse-grid procedure similar to the basic one, the restriction operator yields values for $\Delta\mathbf{f}_c^{2h}$ at the centres of the cells defined by the grid $2h$:

$$\Delta\mathbf{f}_c^{2h} = \Delta t R_h^{2h}\mathbf{r}_h \tag{22}$$

Best results have been obtained when using for R_h^{2h} the 'full weighting' operator, which

Table I

$\mathbf{f}^{(0)} = \mathbf{f}^n$	Solution at iteration n, basic grid
$\mathbf{f}^{(1)} = Q_h\mathbf{f}^{(0)}$	Basic integration method
$\mathbf{r}_h = (\mathbf{f}^{(1)} - \mathbf{f}^{(0)})/\Delta t$	Fine-grid residual
$\mathbf{f}_t^H = R_h^H\mathbf{r}_h$	Restriction $h \rightarrow H$
$\mathbf{f}_{tt}^H = -\nabla_H\left(\frac{\partial\mathbf{F}}{\partial\mathbf{f}}\cdot\mathbf{f}_t^H\right)$	∇_H = divergence operator, discretized on H
$\delta\mathbf{f}^H = \omega(\Delta t\mathbf{f}_t^H + \beta\Delta t^2\mathbf{f}_{tt}^H)$	Coarse-grid correction
$\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + P_H^h\delta\mathbf{f}^H$	Prolongation $H \rightarrow h$; update the solution
$\mathbf{f}^{n+1} = \mathbf{f}^{(2)}$	Solution at iteration $n+1$

Table II

$\mathbf{f}^{(0)} = \mathbf{f}^{n}$	Solution at iteration n, basic grid
$\mathbf{f}^{(1)} = \mathbf{f}^{(0)} + \delta \mathbf{f}_h$	Fine grid integration method
$\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + P_{2h}^{h} \delta \mathbf{f}_{2h}$	Grid $2h$
$\mathbf{f}^{(3)} = \mathbf{f}^{(2)} + P_{4h}^{h} \delta \mathbf{f}_{4h}$	Grid $4h$
$\mathbf{f}^{n+1} = \mathbf{f}^{(3)}$	Solution at iteration $n + 1$

involves all the residuals of the fine-grid points defining the coarse-grid cell. This operator is composed of two averaging operators, whose coefficients are constants independent of the mesh.

Coarse-grid corrections

The coarse-grid corrections are computed using the same procedure as for the basic scheme; that is to say, coarse-grid corrections are given by means of distribution formulae expressed for the grid $2h$ (with the notations of equation (16)):

$$(\delta \mathbf{f}_s)_c^{2h} = \frac{\omega}{8}\left[\Delta \mathbf{f}_c^{2h} + \beta \frac{\Delta t}{V_c^{2h}}(\varepsilon_i \Delta \mathbf{F}_{ic}^{2h} + \varepsilon_j \Delta \mathbf{F}_{jc}^{2h} + \varepsilon_k \Delta \mathbf{F}_{kc}^{2h})\right] \tag{23}$$

Prolongation operator P_{2h}^{h}

A linear interpolation operator is used; P_{2h}^{h} may be split into three elementary operators corresponding to the indices of the mesh. Each of these operators is linear, with constant coefficients, equal to 1/2. In the case of stretched meshes, it has been found that constant coefficients give better results than coefficients involving the distances between the nodes.

Three-grid (and more) procedure

The three-grid procedure runs as shown in Table II.

The basic integration method is applied only to the fine grid; then, for each level of coarse-grid, the procedure used is composed of:

(a) restriction of residuals from the previous grid
(b) distribution formulae
(c) prolongation of corrections to the fine grid
(d) update of the solution.

RESULTS

Two-dimensional case, NACA 12, $M_\infty = 0{\cdot}85$, $\alpha = 1°$

The effectiveness of the multiple-grid procedure has been tested in the case of a transonic flow past the NACA 12 aerofoil, for the conditions $M_\infty = 0{\cdot}85$, $\alpha = 1°$. The mesh is composed of 193×33 points, 145 points being located on the profile (Figure 2).

For all the computations, uniform flow with the far upstream properties is taken as initial condition; convergence is judged by plotting the evolution of the lift coefficient C_L, and also the maximum value of $\Delta \rho u / \Delta t$ against the number of work units (Figure 3(a)). Here the work unit is defined as the computational work required for one iteration of the basic method.

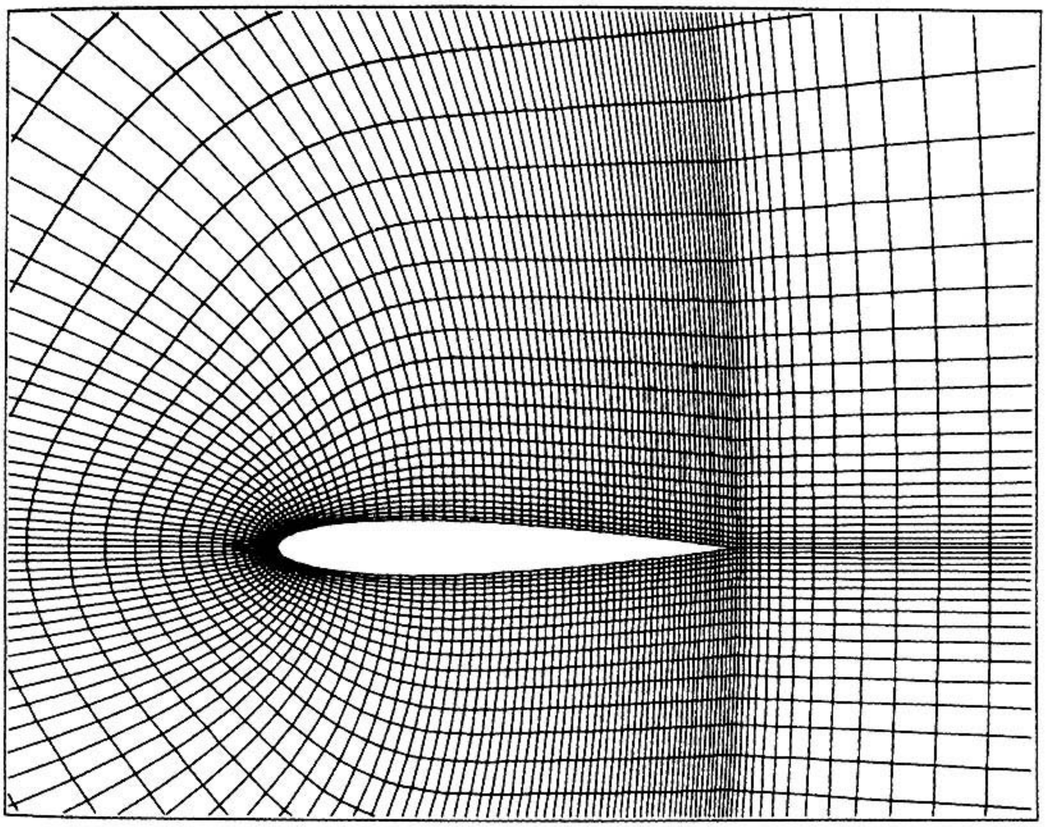


Figure 2. NACA 12 computational mesh (193 × 33)

Table III. NACA 12 aerofoil; $M = 0{\cdot}85$, $\alpha = 1°$

Calculation	1	2	3
Number of iterations	10,000	2000	2000
Number of grids	1 (basic scheme)	3 H–$2H$–$4H$	250 iterations, 4 grids 1750 iterations, 3 grids
Lift coefficient	0·3322	0·3323	0·3323
Drag coefficient	0·0493	0·0493	0·0493
$\lvert\Delta\rho u/\Delta t\rvert_{\max}$	$2{\cdot}0 \times 10^{-5}$	$5{\cdot}8 \times 10^{-7}$	$8{\cdot}4 \times 10^{-8}$
$\lvert\Delta\rho u/\Delta t\rvert_{\mathrm{rms}}$	$1{\cdot}5 \times 10^{-6}$	$4{\cdot}7 \times 10^{-8}$	$6{\cdot}7 \times 10^{-9}$
Work units for one iteration	1	1·30	1·33

Three calculations have been made, the main results are given Table III.

In this case, the computational work is divided by 6 when using the multiple-grid strategy of calculation number 3, the number of iterations required for having the same residuals as obtained at the end of calculation number 1 being equal to 1300. Figure 3(c) presents the isomach contours; total pressure loss and Mach number distribution on the profile are given in Figures 3(b) and 3(d).

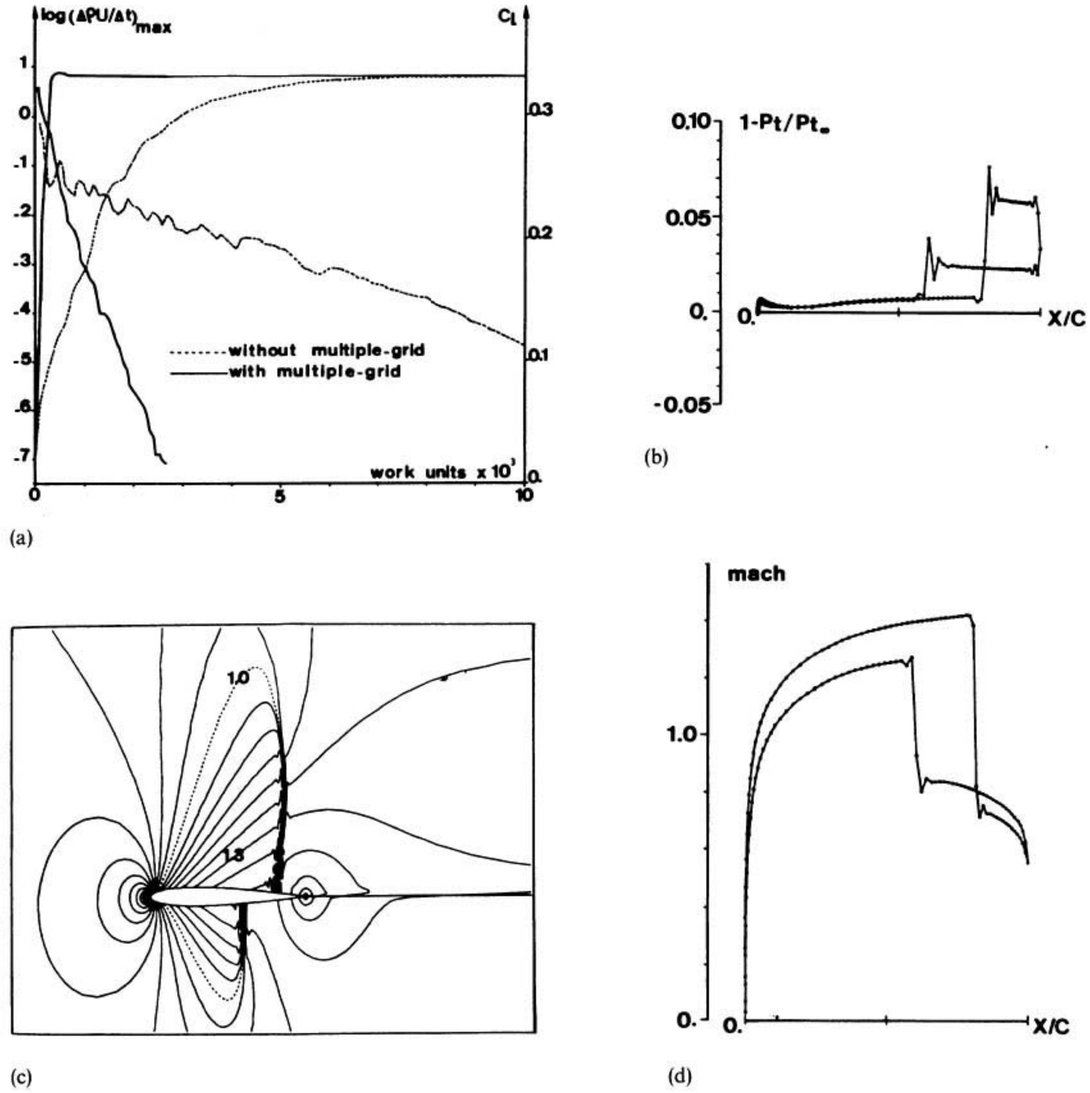


Figure 3. Transonic flow past the NACA 12 aerofoil; $M = 0{\cdot}85$, $\alpha = 1°$: (a) convergence history; (b) total pressure loss; (d) isomach lines (interval $= 0{\cdot}05$); (d) Mach number distribution

Three-dimensional case, ONERA M6 wing, $M_\infty = 0{\cdot}84$, $\alpha = 3{\cdot}06°$

Transonic flow past the M6 wing is computed, first using three grids (H–$2H$–$4H$), and then without multiple-grid capability (–H–). A mesh system of C–O type has been developed, which discretizes finely the rounded tip of the wing (Figure 4). The mesh is composed of $53 \times 49 \times 17$ nodes, with 37×49 points of the wing. Two singular lines are present in the mesh, where hexahedra degenerate into prisms; thanks to the finite-volume formulation, no special treatment is applied to those cells, even during the multiple-grid procedure. Only the linear second order dissipation is used; we must note that this dissipation model is too rudimentary and a more sophisticated model (as used in 2D) will be necessary. For these two calculations, uniform flow is taken as an initial

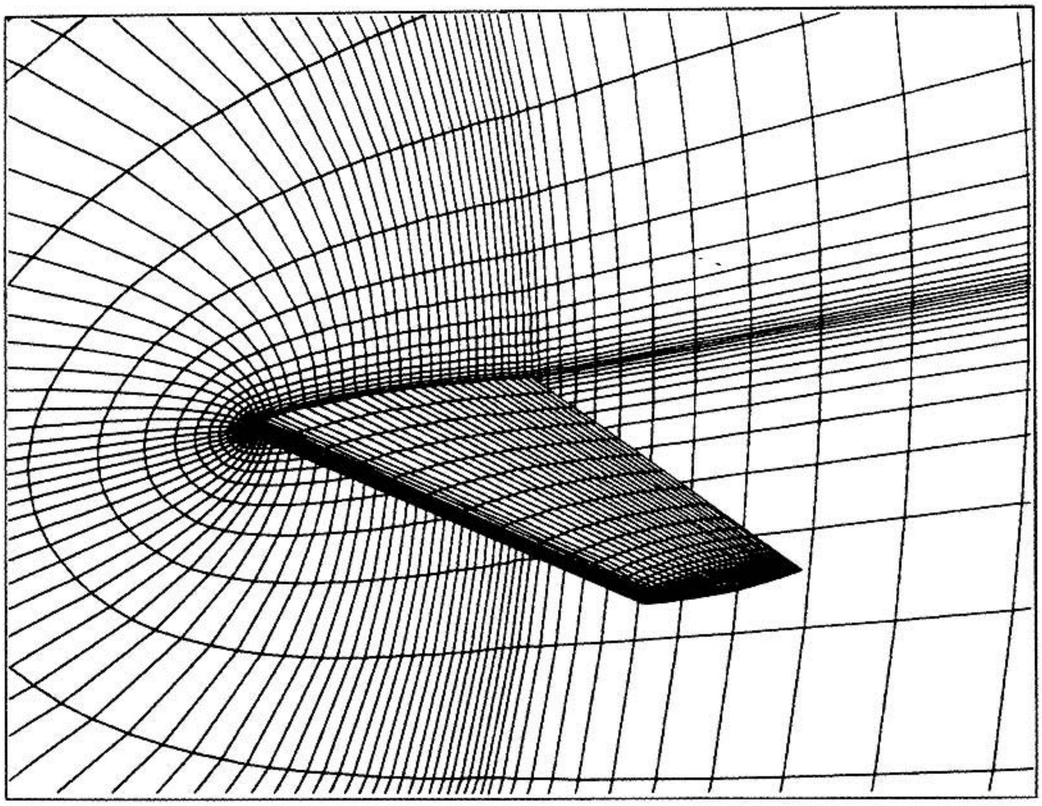


Figure 4. M6 wing—oblique view of the C–O mesh

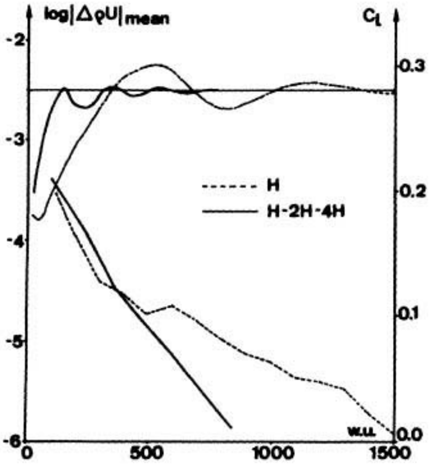


Figure 5. M6 wing, $M = 0{\cdot}84$, $\alpha = 3{\cdot}06°$; convergence history

Table IV. M6 wing, $M = 0{\cdot}84$, $\alpha = 3{\cdot}06°$

Calculation	1	2	3
Number of iterations	700	1500	400
Number of grids	3	1	3
Dissipation coefficient	0·013	0·013	0·009
Lift	0·280	0·278	0·285
Drag	0·0122	0·0122	0·0120
Averaged error	$1{\cdot}4 \times 10^{-6}$	$1{\cdot}3 \times 10^{-6}$	$1{\cdot}3 \times 10^{-6}$
Maximum error	$2{\cdot}8 \times 10^{-5}$	$3{\cdot}6 \times 10^{-5}$	$2{\cdot}9 \times 10^{-5}$
Work units for one iteration	1·20	1	1·20

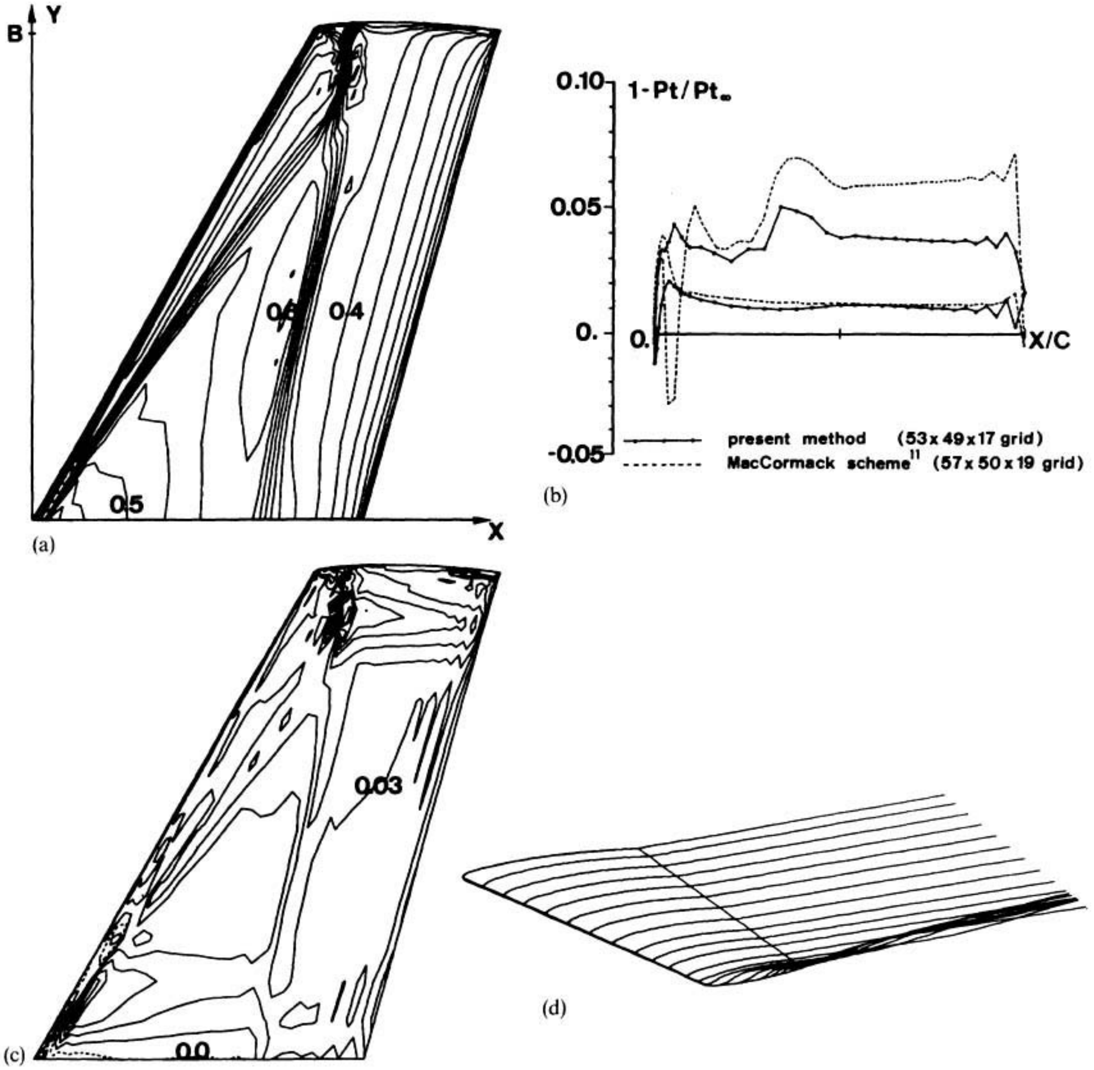


Figure 6. Transonic flow past the M6 wing; $M = 0{\cdot}84$, $\alpha = 3{\cdot}06°$: (a) isobars of normalized pressure $1 - p/Pt$; (b) total pressure loss at $Y/B = 0{\cdot}80$; (c) total pressure loss contours (interval = 0·01); (d) computed streamlines

condition; multiple-grid convergence speeding-up is shown in Figure 5 where the lift coefficient and the averaged error are plotted against the number of work units. Averaged error is defined by:

$$\mathrm{Er} = \frac{1}{N}\sum_{N} |\rho u^{n+1} - \rho u^{n}|.$$

For this test case, the computational time is divided approximately by 2 when using the multiple-grid procedure; we do not have the same effectiveness as for the two-dimensional case; this may be due to the fact that the basic mesh used for this 3D case is rather coarse.

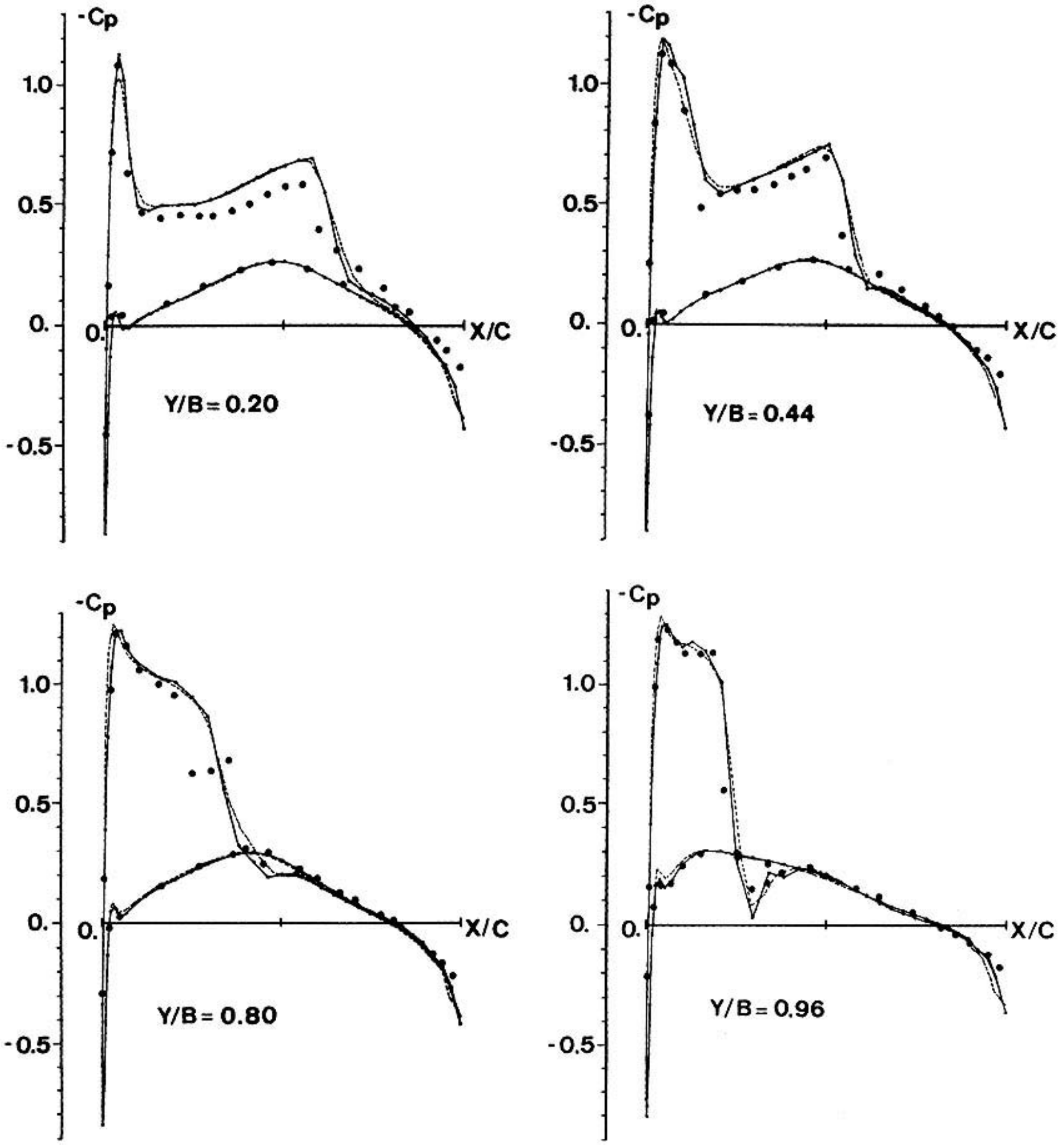
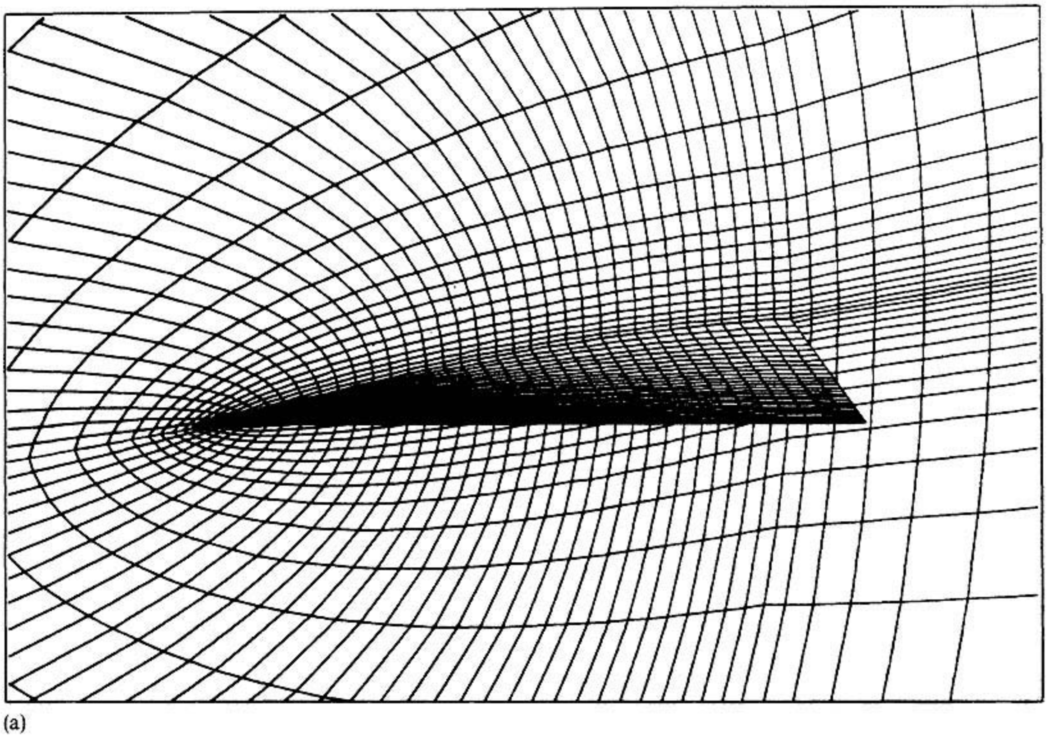


Figure 7. C_p distribution on the M6 wing; $M = 0{\cdot}84$, $\alpha = 3{\cdot}06°$: ———— present method (53 × 49 × 17 grid); -------- MacCormack scheme (57 × 50 × 19 grid);.... experiment[18]



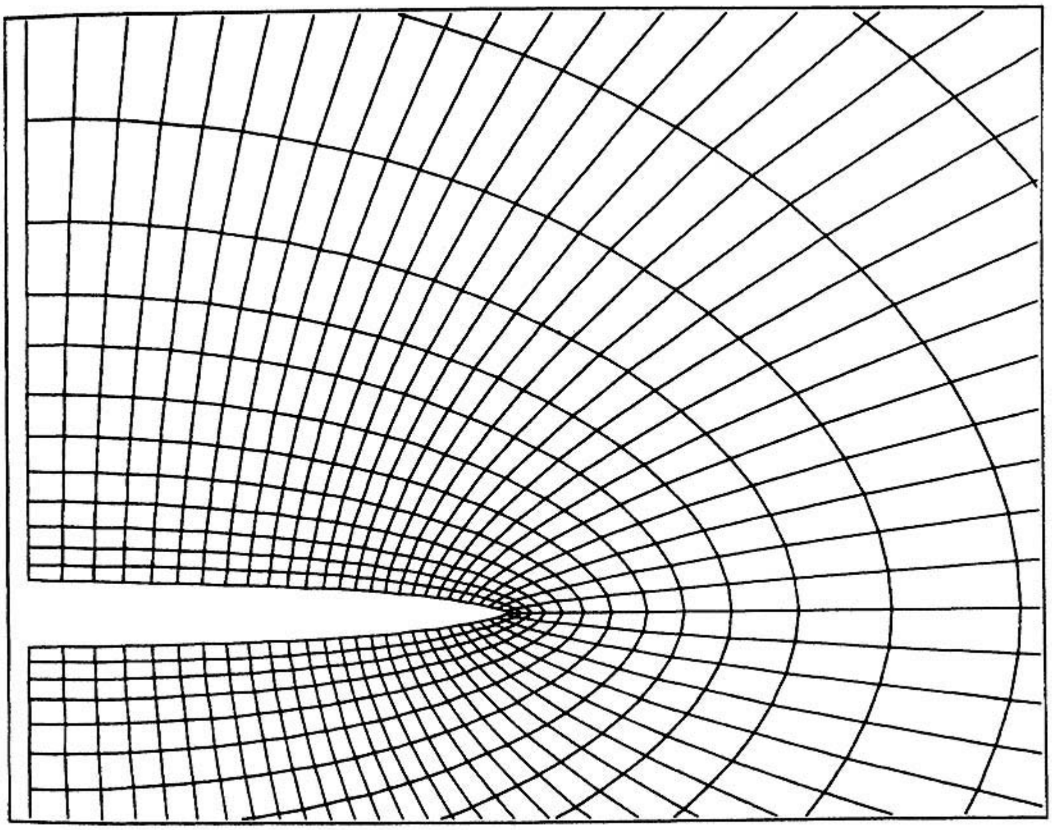


Figure 8. Dillner wing; computation mesh: (a) oblique view of the C—O mesh; (b) planar view of the mesh surface $i = 30$ $(X/C = 0{\cdot}8)$

A third calculation is made with a smaller dissipation coefficient; the solution at the end of the calculation number 1 is taken as the initial condition. The main results concerning these three computations are summarized in Table IV.

The results presented now correspond to the third calculation. Figure 6(a) shows the isobar contours on the upper surface of the wing; the well known lambda shock pattern pertinent to the M6 wing can be seen. Comparison of the C_p distributions with another Euler method using the explicit scheme of MacCormack[14] does not point out notable differences (Figure 7). On the other hand, the total pressure loss (Figures 6(b) and 6(c)) seems to depend rather strongly on the method; in addition to the fact that the two schemes do not have the same properties, the artificial dissipation models are also quite different (the non-linear second-order and linear fourth-order dissipation is used with MacCormack's scheme). In Figure 6(d), the computed streamlines show a small vortex emanating from the wing tip.

Three-dimensional case, Dillner wing $M_\infty = 0{\cdot}7$, $\alpha = 15°$

The Dillner wing is a Delta wing, with a sharp leading edge which is swept 70°; chordwise sections are 6 per cent thick circular arcs. A mesh system of C–O type is used (Figure 8); it is composed of 45 × 57 × 17 nodes, with 37 × 57 points on the wing. The linear dissipation model is used; this model was found to have too large an influence on the vorticity diffusion in the vortex, so we have set its coefficient to a very low value, except for some points very close to the leading edge, where the flow separation induces spurious oscillations. The calculation is carried out with three grids (H–$2H$–$4H$); the initial condition corresponds to the undisturbed uniform flow. The computed flow separates at the leading edge (no condition is applied there), and an important vortex is formed above the upper surface of the wing. A qualitative description of the vortical structure of the flow is given in Figure 9. The leading-edge vortex is clearly revealed Figure 9(a) where the (v, w) components of the velocity vector are plotted for $X/C = 0{\cdot}80$. Total pressure loss contours for the same section show that this loss is maximum in the vortex core (25 per cent of the upstream total pressure). Non-physical phenomena, such as entropy production at the leading edge and vorticity diffusion due to the artificial dissipation are suspected to be responsible for these total pressure distributions, which look like those observed experimentally by Hummel.[16] Figure 9(b) gives indications on the solution in a cross-flow plane located downstream of the wing ($X/C = 1{\cdot}15$). A second vortex appears, corresponding to the trailing-edge wake rolling up; it is interesting to note that the maximum total pressure loss in the leading-edge vortex has not changed. Velocity components normal to the free-stream direction are also plotted in Figure 9(b), and show the double-vortex structure behind the wing.

Quantitative results concerning the interaction between the vortex and the upper surface of the wing are presented Figure 10. Isomach contours (Figure 10(a) and (10(b)) indicate that the flow accelerates through supersonic speeds; the maximum depression is encountered in the vortex core, at the same point as the maximum total pressure loss; Figure 10(e) shows that the total pressure loss on the wing remains low, except in the neighbourhood on the separation point, where very large oscillations can be seen. The C_p distribution on the wing, at $X/C = 0{\cdot}8$, shows the suction peak due to the vortex (Figure 10(f)). Comparison with experimental values points out two principal differences: the computed maximum value of C_p is greater (by ten per cent) than the experimental value; the computed position of the peak is closer to the leading-edge. Analogous results have been obtained by Hoeijmakers,[17] who uses a panel method. Experiments have shown that viscous effects are important in vortical flows; in particular, they are responsible for the formation of secondary vortices which, of course, cannot be predicted by an inviscid gas model (and which may explain why the computed primary vortex lies near the leading edge).

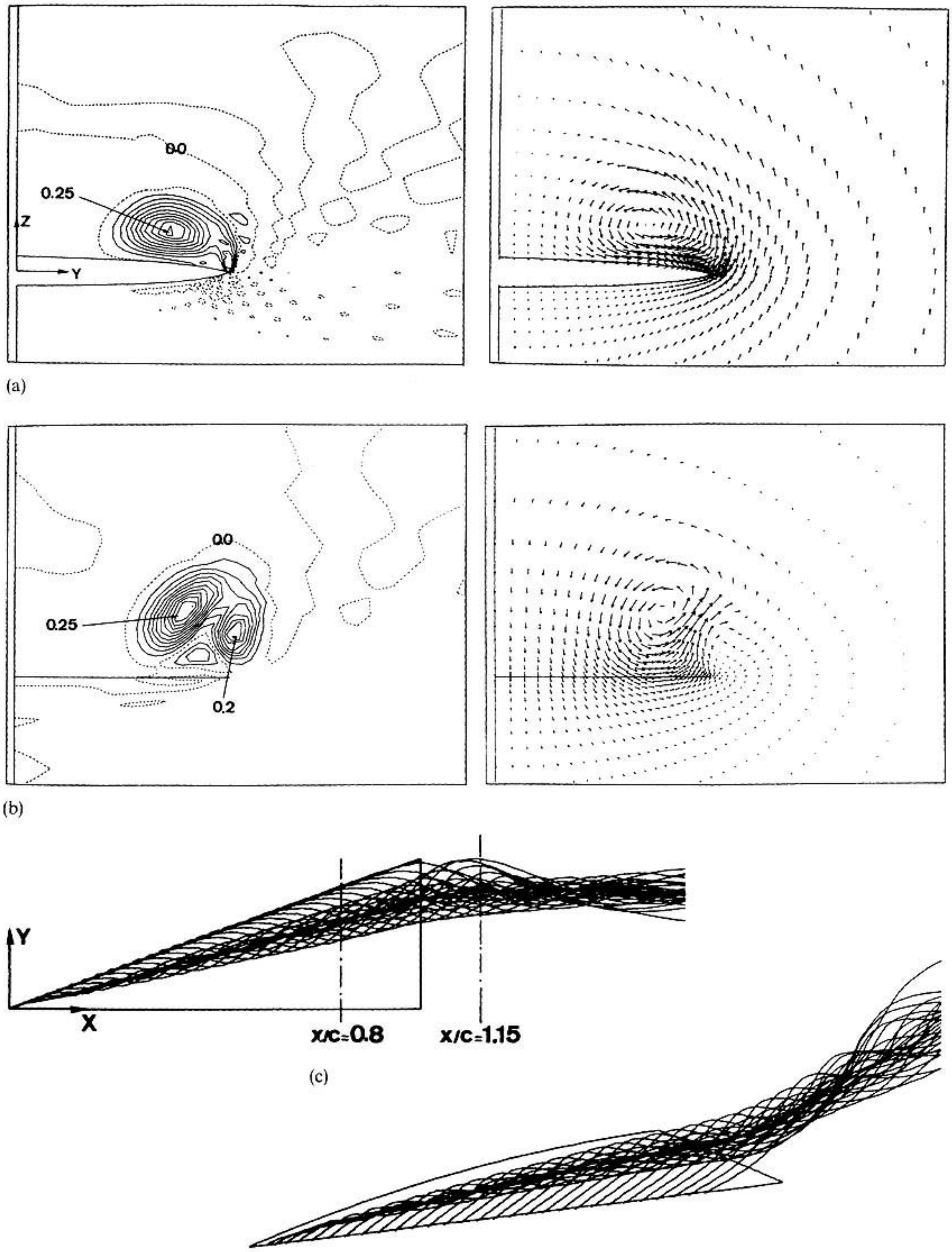


Figure 9. Vortex flow around the Dillner wing; $M = 0{\cdot}7$, $\alpha = 15^\circ$: (a) total pressure loss and cross-flow velocities (wing co-ordinates) at $X/C = 0{\cdot}80$; (b) total pressure loss and cross-flow velocities (free-stream co-ordinates) at $X/C = 1{\cdot}15$; (c) computed streamlines

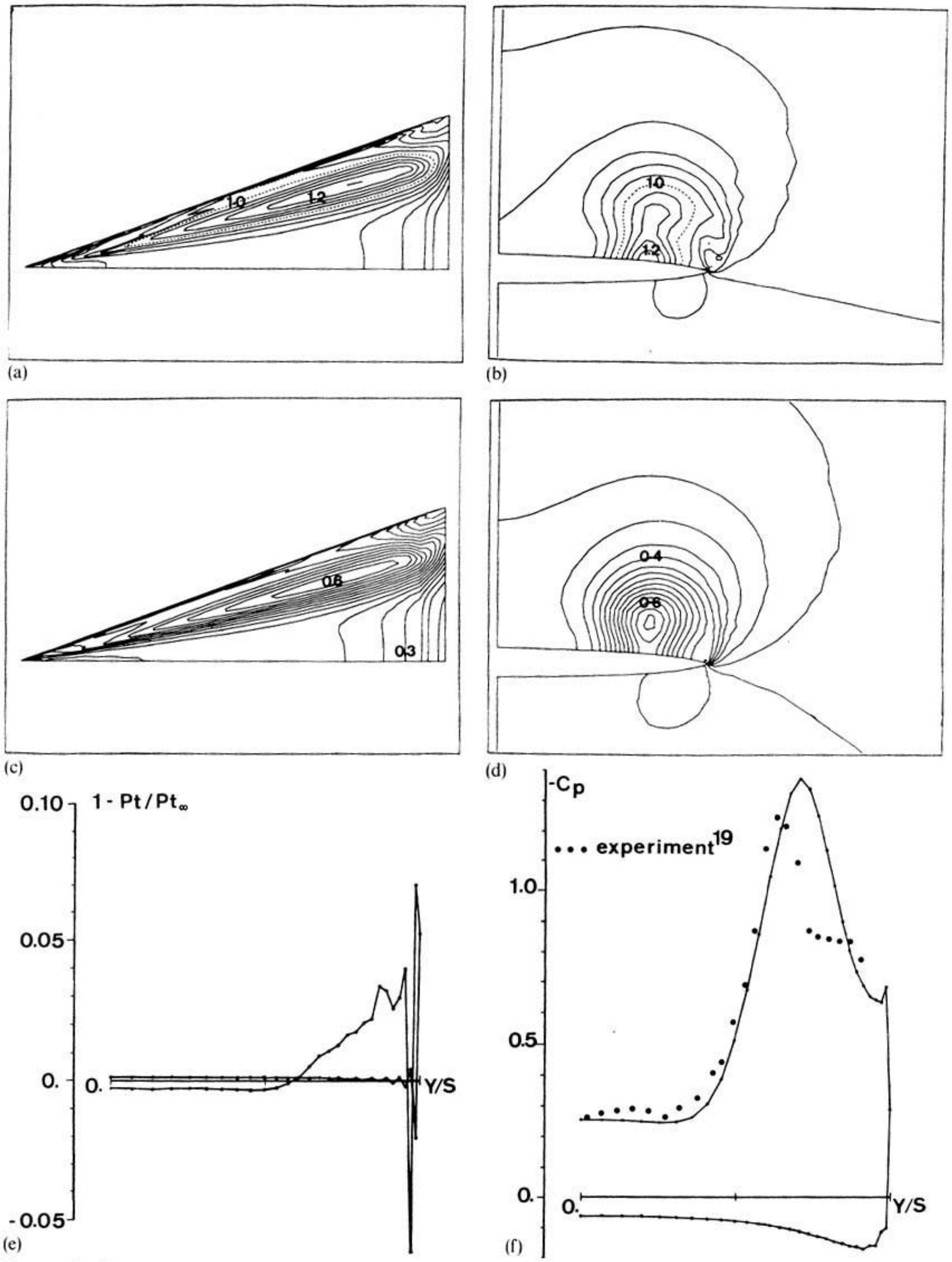


Figure 10 Dillner wing; $M = 0{\cdot}7$, $\alpha = 15°$: (a) isomach lines on the upper surface; (b) isomach lines at $X/C = 0{\cdot}8$; (c) isobars of normalized pressure on the upper surface, $C_D = 0{\cdot}171$, $C_L = 0{\cdot}678$; (d) isobars at $X/C = 0{\cdot}8$; (e) total pressure loss on the wing at $X/C = 0{\cdot}8$; (f) C_p distribution on the wing at $X/C = 0{\cdot}8$

CONCLUSION

The multiple-grid scheme of Ni has been used for the computation of three-dimensional flows, and it has given good results. Using the multiple-grid procedure leads to reductions of the computational time by a factor 2 or 3 in the three-dimensional cases; this factor can reach values up to 6 for two-dimensional computations with fine basic meshes; in addition, the numerical implementation of this method was found to be easy. The computational time per grid point and per iteration for the basic finite-volume scheme is approximatively the same as for the two-step explicit scheme of MacCormack; the additional cost due to the multiple-grid procedure is about 20 per cent for the three-dimensional code.

The ability for the Euler method to compute vortical flows past wings at large angles of attack is shown by the results concerning the Dillner wing; however, an accurate capturing of the vortex sheets remains a difficulty, and a great number of mesh nodes seems to be necessary. In any case, non-physical phenomena are present in the computed solutions; they are related to the numerical discretization of the Euler equations and to the artificial dissipation model; when those effects are small enough, the Euler equations method can give satisfactory global description of vortical flows.

ACKNOWLEDGEMENT

This work was performed with the partial financial support of DRET.

REFERENCES

1. W. Schmidt and A. Jameson, *Recent Developments in Finite-Volume Time-Dependent Techniques for Two- and Three-Dimensional Transonic Flow*, VKI Lecture Series 1982–04, Brussels, March 1982.
2. A. Jameson, 'Transonic aerofoil calculations using the Euler equations', *IMA Conf. on Numerical Methods in Aeronautical Fluid Dynamics*, Reading, March 1981.
3. A. Jameson, W. Schmidt and E. Turkel, 'Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes', *AIAA-Paper 81–1259*, 1981.
4. A. Rizzi and H. Viviand (eds), *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves*, Vieweg Verlag, 1981.
5. L. E. Eriksson, and A. Rizzi 'Computation of vortex flow around a Canard-delta combination', *Proc. Fifth GAMM Conference on Numerical Methods in Fluid Mechanics*, Rome, 1983, 5–7 October Vieweg Verlag.
6. L. E. Eriksson and A. Rizzi, 'Computation of Vortex flow around wings using the Euler equations', in H. Viviand (ed.), *Proc. 4th GAMM Conference on Numerical Methods in Fluid Mechanics*, Vieweg Verlag, Paris, 1981.
7. A. Lerat, J. Sides and V. Daru, 'An implicit finite-volume method for solving the Euler equations', *Lecture Notes in Physics*, **170**, 286–295 (1982).
8. R. H. Ni, 'A multiple-grid scheme for solving the Euler equations', *AIAA Paper 81–1025*, June 1981.
9. G. M. Johnson, 'Multiple-grid convergence acceleration of viscous and inviscid flow Computations', *Applied Mathematics and Computation*, **13**, (3–4), (1983).
10. G. M. Johnson, 'Flux-based acceleration of the Euler equations', *NASA TM-83453*, Presented at the *AIAA SIXTH Computational Fluid Dynamics Conf.*, Danvers, Mass., 13–15 July 1983.
11. A. Jameson, 'Solution of the Euler equations for two-dimensional transonic flow by a multigrid method', *International Multigrid Conference*, Copper Mountain, April 1983; *Applied Math. and Computation*, **13**, pp. 327–355 (1983).
12. H. Viviand and J. P. Veuillot, 'Méthodes psedo-instationnaires pour le calcul d'écoulements transsoniques'. ONERA, *Publication no. 1978–4*, 1978.
13. J. Brochet, 'Calcul numérique d'écoulements internes tridimensionnels transsoniques', *La Recherche Aérospatiale*, pp. 301–315, no. 1980–5. English translation ES A-TT 673.
14. Ch. Koeck, and M. Neron, 'Computation of three-dimensional inviscid flow on a wing by pseudo-unsteady resolution of the Euler equations', *Proc Fifth GAMM Conference on Numerical Methods in Fluid Mechanics*, Rome, 5–7 October 1983, Vieweg Verlag.
15. Ch. Koeck and J. J. Chattot, 'Computation of three dimensional flow using the Euler equations and a multiple-grid scheme', 9th ICNMFD, June 1984. Saclay, France.
16. D. Hummel, 'On the vortex formation over a slender wing at large incidence', *AGARD CP-247*, 1979.
17. H. W. M. Hoeijmakers, 'Computational vortex flow Aerodynamics', *AGARD CP-342*, 1983.
18. V. Schmidt and F. Charpin, 'Pressure distributions on the ONERA-M6 wing at transonic Mach number', *AGARD AR-138*, 1979.
19. P. O. Larson, 'Investigation of the flow around delta wings with transonic speeds' *FFA report, No. AU-295*, September 1955.