AIAA JOURNAL
Vol. 30, No. 3, March 1992

Cartesian Euler Method for Arbitrary Aircraft Configurations

Boris Epstein,* Alexander L. Luntz,† and Aharon Nachshon‡
Israel Aircraft Industries, Ben Gurion International Airport, 70100 Israel

A three-dimensional Euler method is described, for steady flow calculations about arbitrary aircraft configurations. The method uses Multigrid calculations with equally meshed Cartesian (not necessarily rectangular) grids and local refinement. The use of local computational grids that are not aligned to the body surface, or even, possibly, not aligned to one another, removes the need for complicated grid generation, but introduces the need for sophisticated boundary condition implementation. Appropriate boundary condition implementation is an important part of the algorithm. Computation examples show the ability of the method to produce results in good agreement with experiment, for a wide range of flight conditions.

Nomenclature

C_D = drag coefficient
C_L = aircraft lift coefficient
C_p = pressure coefficient
c = speed of sound
D = dummy point
$\boldsymbol{D}_h$ = dissipation vector term, in the discretized Euler equation
$\boldsymbol{E},\boldsymbol{E}_h$ = Euler spatial vector operator, continuous and discretized
e = specific energy, per unit mass
$\boldsymbol{F}$ = $(\rho,\rho u,\rho v,\rho w,\rho e)$ vector of Euler variables
G,G^h = grid, grid box
h = mesh size
H = $e + p/\rho$ − enthalpy
$\boldsymbol{n}$ = vector normal to the surface (of the body or of the computational box)
M = freestream Mach number
p = pressure
$\boldsymbol{q}$ = stream velocity vector
s = entropy
u,v,w = components of $\boldsymbol{q}$ in global coordinates x,y,z
$vs0,vs2$ = dissipation parameters
x,y,z = global coordinates: x along the body axis, y up, z spanwise
α = angle of attack
ρ = density
γ = ratio of specific heats

Subscripts

e = extrapolated
∞ = freestream value

Introduction

THE need for computational transonic aerodynamic codes able to handle arbitrary configurations has been widely recognized. It is also recognized that it is unrealistic to design one universal method able to generate a grid about any given configuration, just by mapping the outside of the configuration onto a semispace.

Three main methods have been used to overcome these difficulties (for various approximations to the complete flow equations, including potential and Euler). One method creates smooth local grids, aligned to the body surface, and then tailors them into one universal grid. This approach is virtually unlimited in handling complicated configurations, but is far from being fully automated (it demands months usually, for an experienced engineer to complete a grid about a new, complicated configuration).

Another method,[2,3] using unstructured (but body-aligned) grids, automatically generated from a cloud of vertices. This cloud originates from a collection of separate grids, each one about a different element of the configuration. Given that all difficulties associated with this approach have been overcome, this procedure allows for dealing with the geometrical difficulty of really arbitrary three-dimensional configurations. The solution process itself still has all the drawbacks of fully nonregular grids.

The third method is the use of Cartesian grids. They allow second-order accuracy approximation to the governing equations in the free air. They are not body-surface aligned and, thus, eliminate almost all the problems of grid generation. At the same time, the use of Cartesian grids introduces the problem of body-boundary condition implementation. Several efforts have been made to use Cartesian grids as a means to enable the code to handle arbitrary three-dimensional configurations.[4] To the best of the author's knowledge, this goal has not been achieved as yet.

The present paper describes a three-dimensional Euler code for steady flows, which uses uniform Cartesian (not necessarily rectangular) grids. Local grid refinement is used, to achieve the needed resolution without overflooding the computer memory with grid points. Multigrid approach is used, not only to speed up the computation but mainly to hold all the local grids together in the field solution. A given aircraft configuration is treated as a Boolean sum of bodies. A satisfactory body-boundary condition discretization has been necessary for the success of the method as a whole.

The design of the previously described Multigrid Full Potential Code for Arbitrary Configurations (MGPOTAC)[5,6] has been based on similar principles. However, the implementation of these principles in the present Euler method is substantially different, for two reasons: 1) the difference in the mathematical model of the flow; and 2) the improvements made in those parts of the algorithm common to both codes.

Several application cases are presented to show the accuracy of the method (as compared with experiment) and its ability to handle complicated configurations in different flight conditions, including high angles of attack and supersonic flow.

Presented as Paper 89-1960 at the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY, June 13–15, 1989; received Sept. 17, 1990; revision received June 4, 1991; accepted for publication July 19, 1991.

*Head of CFD Group, Dept. 2473. Member AIAA.
†Senior Research Scientist, CFD Group, Dept. 2473.
‡Research Scientist, CFD Group, Dept. 2473.

Equations and Numerical Scheme

The flow equation solved is the three-dimensional Euler equation:

$$\begin{aligned} \rho_t + \operatorname{div}(\rho \boldsymbol{q}) &= 0 \\ (\rho u)_t + \operatorname{div}(\rho u \boldsymbol{q}) + p_x &= 0 \\ (\rho v)_t + \operatorname{div}(\rho v \boldsymbol{q}) + p_y &= 0 \\ (\rho w)_t + \operatorname{div}(\rho w \boldsymbol{q}) + p_z &= 0 \\ (\rho e)_t + \operatorname{div}(\rho \mathrm{H} \boldsymbol{q}) &= 0 \end{aligned} \tag{1}$$

or

$$\boldsymbol{F}_t + \boldsymbol{E} = 0 \tag{2}$$

with the state equation

$$e = p/[\rho(\gamma - 1)] + \boldsymbol{q}^2/2 \tag{3}$$

The present method uses Cartesian grids (rectangular or swept) for the conservative spatial discretization of Eq. (1). Symmetrical schemes are used for the second order accurate approximation of the spatial derivatives. Dissipative terms[7] (second differences only) are introduced in the discretized equation for reasons of numerical stability (including elimination of the odd-even grid points decoupling of the solution) and consistent physical modeling (avoiding expansion shocks, etc). The discretized system is solved using the modified four-stage Runge-Kutta scheme.[8] After each Runge-Kutta stage, the calculated correction to the solution is smoothed, using factorized "residual smoothing."[9] The calculation process is described here in more detail for the special case of a rectangular grid with grid lines parallel to the global coordinate axes (with obvious modifications for other types of Cartesian grids used).

The definition of the dissipative vector term $\boldsymbol{D}_h$ is similar for each of the five equations in Eq. (1). At the grid point (i,j,k)

$$\begin{aligned} D_{i,j,k} = {} & d_{i+1/2,j,k} - d_{i-1/2,j,k} \\ & + d_{i,j+1/2,k} - d_{i,j-1/2,k} \\ & + d_{i,j,k+1/2} - d_{i,j,k-1/2} \end{aligned} \tag{4}$$

is calculated.

Here, for the x coordinate for example:

$$d_{i+1/2,j,k} = [\varepsilon_{i+1/2,j,k}\,(\eta_{i+1,j,k} - \eta_{i,j,k})]/\Delta\tau$$

where

$$\begin{aligned} \varepsilon_{i+1/2,j,k} &= \max(\nu_{i+1,j,k}, \nu_{i,j,k}) \\ \nu_{i,j,k} &= \max[\nu s0, |(p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k})/ \\ &\quad (p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k})|] \end{aligned} \tag{5}$$

η being one of the variables ρ, ρu, ρv, ρw, ρH.

The above defined dissipative vector $\boldsymbol{D}_h$ is calculated with the first of the four Runge-Kutta stages for the discretized equation:

$$\partial \boldsymbol{F}_h/\partial t + \boldsymbol{E}_h - \nu s2 \cdot \boldsymbol{D}_h = 0 \tag{6}$$

and frozen for the following three stages.

To start the four-stage calculation at a given iteration cycle, the three-dimensional analog to $\Delta\tau = h/[\max(|u| + c)]$ is calculated, the maximum being taken over all grid points of the grid G^h. The time step used is $\Delta t = \Delta\tau \cdot C_{CFL}$ for all grid points of the grid G^h.

The coefficient C_{CFL} is limited by the CFL condition depending on both terms $\boldsymbol{E}_h$ and $\boldsymbol{D}_h$ in Eq. (6) (apparently, mainly by $\boldsymbol{E}_h$). The additional limitation comes from the discretized boundary condition. In the actual computations, C_{CFL} seemed to be limited by 1 in the calculations not using the residual smoothing. With the residual smoothing, the actual limit is up to 8.

Boundary Conditions

At the faces of the coarsest grid box—the computational "infinity"—Riemann invariants are used.[8,10]

The normal (to the face) velocity component and the speed of sound are defined as

$$\begin{aligned} q_n &= \boldsymbol{q} \cdot \boldsymbol{n} = \cdot 5 \cdot (R_e + R_\infty) \\ c &= 0.25(\gamma - 1) \cdot (R_e - R_\infty) \end{aligned}$$

where

$$\begin{aligned} R_\infty &= \boldsymbol{q}_\infty \cdot \boldsymbol{n} - 2c_\infty/(\gamma - 1) \\ R_e &= \boldsymbol{q}_e \cdot \boldsymbol{n} + 2c_e/(\gamma - 1) \end{aligned}$$

At the outflow boundary, the velocity vector is defined as

$$\boldsymbol{q} = \boldsymbol{q}_e + (q_n - \boldsymbol{q}_e \cdot \boldsymbol{n})\,\boldsymbol{n}$$

while at the inflow boundary

$$\boldsymbol{q} = \boldsymbol{q}_\infty + (q_n - \boldsymbol{q}_\infty \cdot \boldsymbol{n})\,\boldsymbol{n}$$

With c and $\boldsymbol{q}$ known, the set is completed using entropy, by setting:

$$s = s_e \quad \text{or} \quad s = s_\infty$$

for outflow or inflow boundary, respectively.

At the local grid outer boundaries, Dirichlet conditions are used. These are reflecting boundary conditions and would lead to instability if used in a series of iterations for an isolated grid. However, only one (four-stage) iteration is used at a time, for a given grid, in the multigrid process. Between every two such iterations, there is a coarse grid correction or a fine grid smoothing. This seems to be enough to provide the necessary dumping of the reflection, enabling overall convergence.

The body-boundary condition implementation, with grids that are not body aligned, is the most critical part of all, and its success is crucial to the success of the method as a whole. The assignment of the five variable values at a dummy point is done by extrapolation. The selection of points for the extrapolation and the use of the least-squares method are roughly similar to those used in the MGPOTAC code.[5,6] However, the extrapolation formulae used differ from those used in MGPOTAC and are different for different variables.

For a grid point R in the air and a dummy point D in its template (Fig. 1), we introduce a local coordinate system, with one of the axes along the normal n to the surface and two others in the tangent plane. In these coordinates, the components of the velocity vector $\boldsymbol{q}$ are ϕ, Ψ, χ respectively. The value of ϕ at the dummy point results from a linear extrapolation of the calculated values of ϕ at the selected points $\{R_i\}$, using the information that ϕ is zero at the surface point RD (this is actually the only information external to Eqs. (1), i.e., the only boundary condition used at the surface). Ψ, χ are extrapolated linearly from $\{R_i\}$ only. The pressure p is extrapolated to D with

$$\partial p/\partial n = -\rho|\boldsymbol{q}|(\partial \boldsymbol{q}/\partial \eta) \cdot \boldsymbol{n}$$

at RD, η being the length parameter along the streamline. (This value of $\partial p/\partial n$ is a consequence of the Euler equation and, thus, is not an additional boundary condition.)

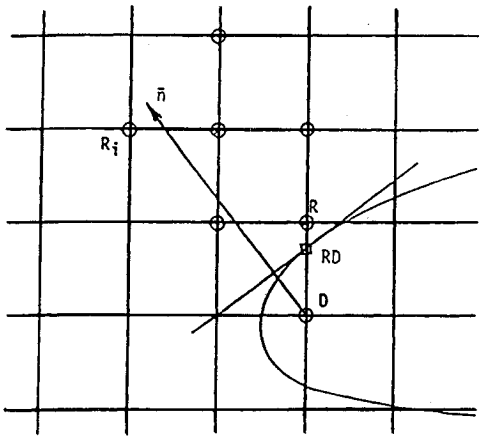

Fig. 1 Body-boundary condition implementation.

There are two options in the code to assign the fifth variable, namely ρ, at D. One is the assumption of constant, freestream value of enthalpy H at D (as we are interested in the steady state only). The other is the calculation of ρ from the extrapolated value of the entropy s. In most of the computations, the first option was used.

The extrapolation, described above for the assignment of the variable values at dummy points, gives satisfactory results where the variables are continuous in the direction of extrapolation, near the body surface. This is true in most applications.

However, in some supersonic applications, this assumption seems to be unjustified. An example is the case where the Mach cone discontinuity remains near the body surface (see Results). In such cases, the following lower order extrapolation is used: the normal (to the body surface) component of the velocity vector at the reference point R, is assigned to D with change of sign; the two other components of the velocity and the pressure p at R are assigned unchanged to D; and the density ρ is calculated from the assumption of constant (freestream) value of enthalpy H.

The switch to the latter body boundary condition implementation depends on the measure of linearity of the flowfield in the neighborhood of R.

Multigrid and Local Refinement

The use of the local refinement in multigrid calculations, with grids not aligned to the body surface and, possibly to each other, introduces several new problems, such as construction of the right-hand side correction RHS[11] near the grid limits, exchange of information between overlapping grids of one multigrid level, consistency between fine-to-coarse operator correction RHS, and coarse-to-fine solution correction with overlapping grids, and so forth. These problems have been dealt with in the design of the Multigrid Full Potential Code (MGPOTAC).[6] Some of those solutions have been adopted with the present Euler code, whereas others have been improved.

The treatment of the fine-to-coarse operator correction RHS (right-hand side) for Eq. (6) in the framework of local refinement is similar to that used in the MGPOTAC code.

The treatment of overlapping is largely improved, compared with MGPOTAC. There are no restrictions on grid overlapping in one multigrid level. For a point R belonging to several grid boxes $G_1^h, \ldots, G_k^h$ of one multigrid level, a "host" grid G^h is chosen among $\{G_i^h\}$, within which R is "most deeply" embedded (the actual criteria is the distance of R from the grid limits, in terms of number of grid cells). The notion of "host" grid is then used for exchange of information between overlapping grids, fine-to-coarse level solution interpolation, residual interpolation (weighting), and coarse-to-fine correction interpolation.

In the construction of interpolation formulae for body boundary condition implementation (previous section), the whole set of grid points of all overlapping grids, at a certain multigrid level, is used for the selection of extrapolation points, regardless of in which grid the template in question belongs.

As a result, in the special case of overlapping grids with their grid points coinciding in the overlapping zone, the calculation works as if it were one large grid (with the exception of residual smoothing).

The multigrid numerical procedure as a whole is not fully conservative, due to the lack of a special conservative tailoring at the boundaries of local refinements, in the fine-coarse grid interpolation. Still, the impact of the deviation from the conservativity, in the finite difference operator, is bounded and is confined to a region with a three-dimensional measure tending to zero as $h \to 0$. Thus, the computational schemes, with the body boundary condition implementation consistent with the problem, ensure essential conservativity, sufficient asymptotically for the application of the Lax-Wendroff theorem[12] and the proper representation of the essential flow features.

Cartesian Euler Code Structure

The code consists of 1) three preprocessors PP1, PP2, and PP3; 2) the Multigrid Euler computation code MGE; and 3) postprocessor PTP. The preprocessors interpret the geometry of a given aircraft configuration in terms of the computational grids chosen. Their output may be used as input for several runs of MGE and PTP, with different flight conditions.

The preprocessor PP1 deals with every computational grid separately. It determines which grid points are inside the aircraft and which points are real (outside the aircraft). For every real point, it finds all of the dummy points in its template and the intersection data of the interval real-dummy with the body surface. Preprocessor PP2 deals with relations between grids. It prepares the formulae to be used in MGE, for coarse-to-fine and fine-to-coarse levels interpolation, exchange of information between overlapping grids, and the multigrid fine-to-coarse operator correction (RHS). It constructs the formulae to calculate the variable values at dummy points, as functions of the last calculated values at real points.

With nonaligned grids, the integration of a certain field value (such as C_p) over the configuration surface, becomes a complicated task. Preprocessor PP3 is designed to prepare formulae to be used in the postprocessor PTP for actual configuration. The construction of these formulae consists of a panelling of the surface, based on the input definition of the bodies composing the configuration, and of extrapolation formulae to the surface. All three preprocessors use extensively the set of geometrical subroutines designed for MGPOTAC.[6]

The postprocessor PTP, besides integration, prepares data for graphical output (C_p distribution at different surface cuts, stream vector field at given planes, etc.).

Results

The ability of the present approach to handle complicated aircraft configurations has been demonstrated with potential calculations.[6]

The main challenge for Euler computations is the modelling of those regimes of flight that are not covered by full potential methods. Typical examples are vortex separation from the leading edge of swept wings and high supersonic flow. The following applications include these flight conditions.

Fighter Configuration: Wing-Body-Canard-Fairing

Figure 2 shows the configuration used in both the wind tunnel test[13] and in the computation. Trial computations have shown insensitivity of the wing pressure distribution to the presence of boundary layer for subsonic freestream and, thus, the nominal configuration was used with the code for $M < 1$.

Figure 3 shows the configuration planform, with the grids of the fourth, fifth, and sixth multigrid levels.

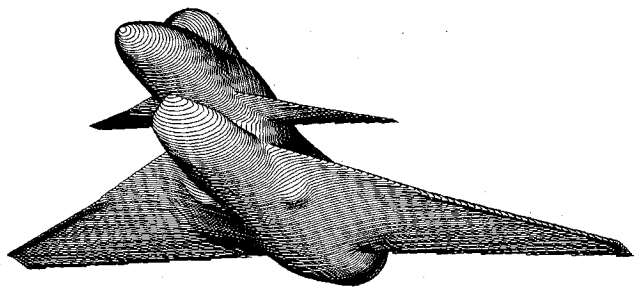

Fig. 2 Fighter configuration, wing-body-canard.

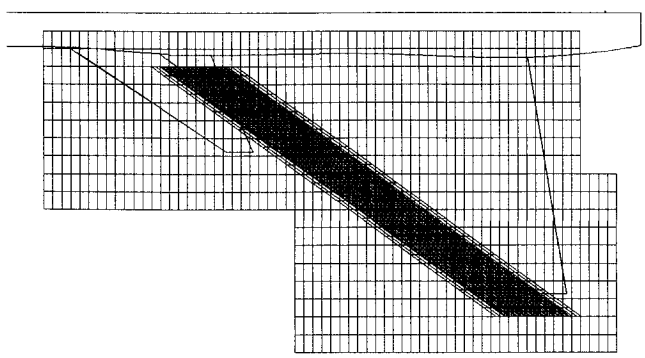

Fig. 3 Fighter configuration, planform with grids of the fourth, fifth, and sixth levels.

Figure 4 shows the calculated C_p distribution at four wing stations, as compared to experiment, for $M = 0.9$, $\alpha = 8$ deg. At these flight conditions, the flow is mainly potential.

At $\alpha = 24$ deg, the flow is fully separated (Fig. 5). This is a leading edge vortex separation, and Figs. 6, 7, and 8 display the velocity vector distribution at different planes. Fig. 8 shows the canard wake vortex and its interaction with the vortex separation from the wing leading edge, at different sections normal to the body axes.

Figure 9 shows the calculated C_L vs α at $M = 0.9$, up to $C_{L\max}$ at $\alpha = 26$ deg, as compared to the experimental data. It shows also the comparison of the calculated and experimental C_D. A constant friction drag has been added to the calculated C_D.

The total number of grid points used for this configuration, in all six multigrid levels, is about 85,000. The total CPU (job and monitor) time used, for each flight point, is about 4 h on CDC CYBER 860 (NOS/VE). This computer has no vectorization and its clock time is 40 ns. Trial runs show that on a CRAY XMP the expected speedup is about 20 times, so that computation at one flight point (M, α fixed), would take about 12 min for this configuration.

Sharp-edged Delta-wing

The flow around a sharp-edged delta-wing with leading edge sweep 65 deg (International Vortex Flow Experiment[14]) was computed. Results are provided for high angle of attack at transonic speed ($M = 0.85$, $\alpha = 18$ deg).

The computed configuration consisted of the above mentioned delta-wing mounted on a sting, as in the wind tunnel experiment. Computational grids (five multigrid levels, seven grids) contain approximately 172,000 grid points. Figure 10

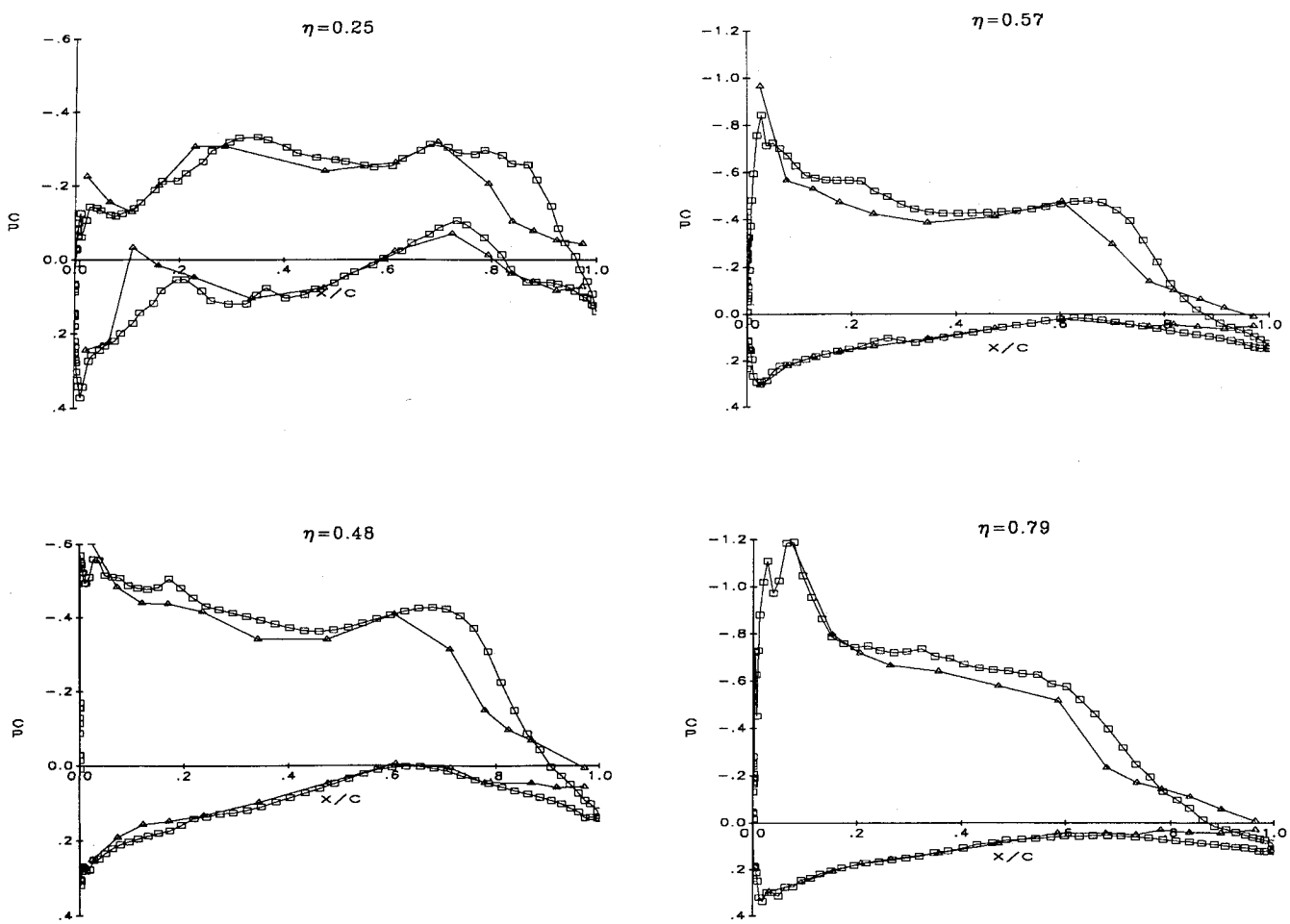

Fig. 4 Fighter configuration, C_p distribution, at four wing sections: $M = 0.9$, $\alpha = 8$ deg, □ = computation, △ = experiment.[13]

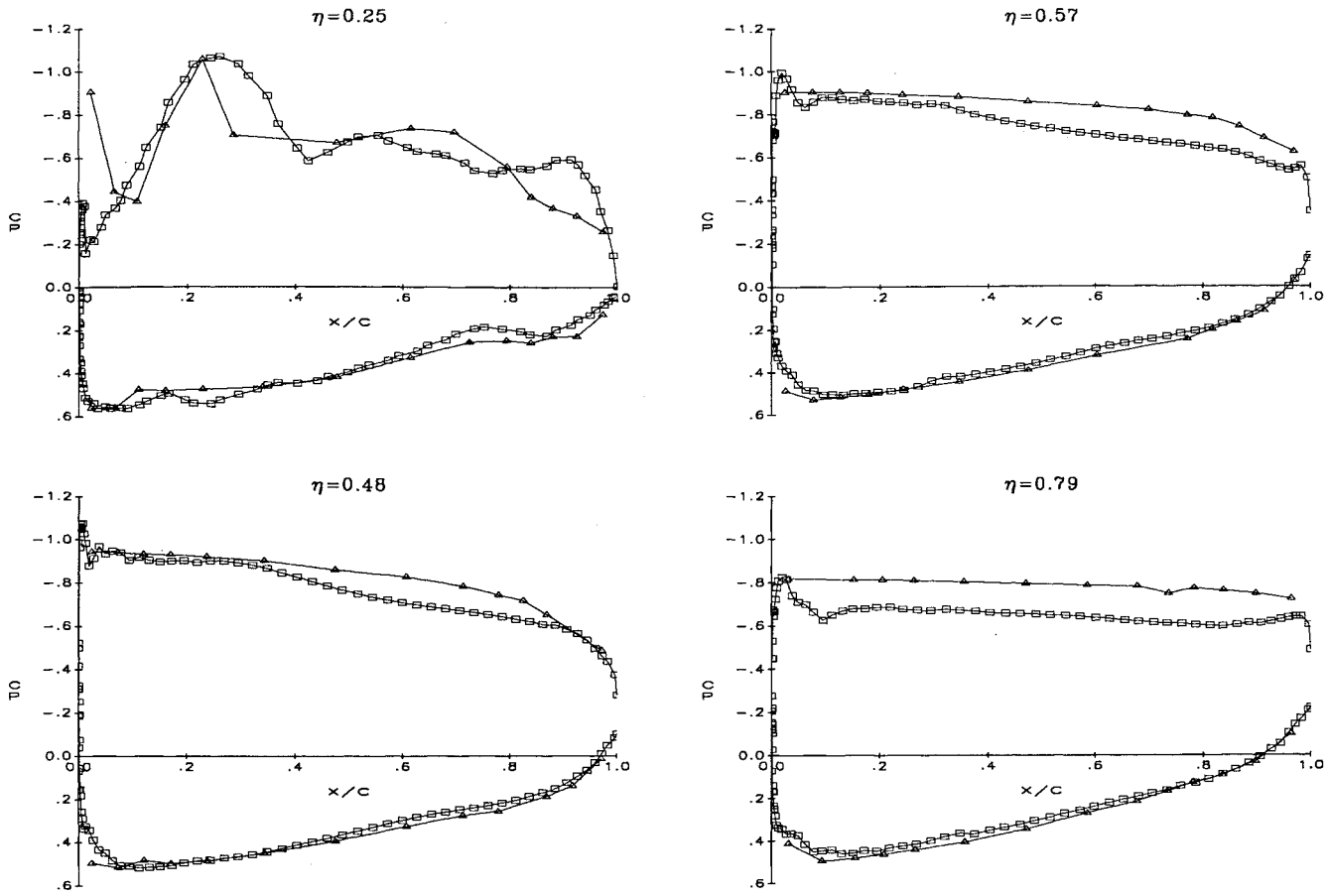

Fig. 5 Fighter configuration, C_p distribution, at four wing sections: M = 0.9, α = 24 deg, □ = computation, △ = experiment.[13]

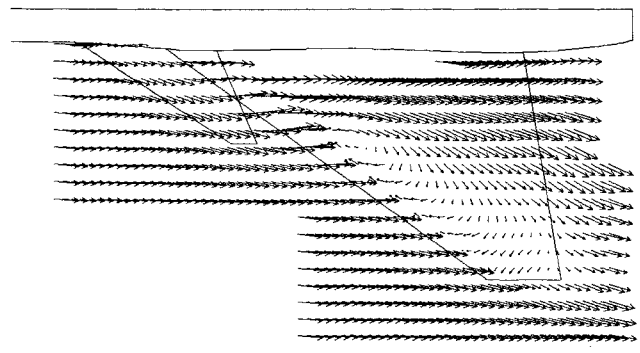

Fig. 6 Fighter configuration, planform, with velocity flowfield above the wing: M = 0.9, α = 24 deg.

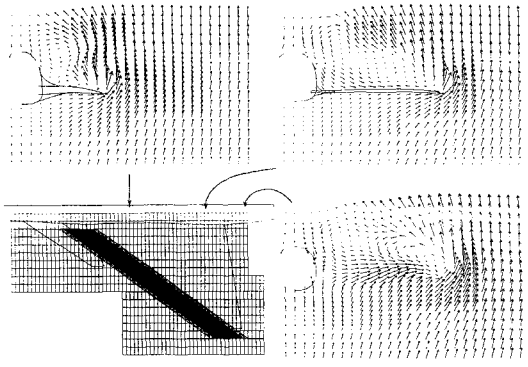

Fig. 8 Fighter configuration, velocity vector field at three cross sections: M = 0.9, α = 24 deg.

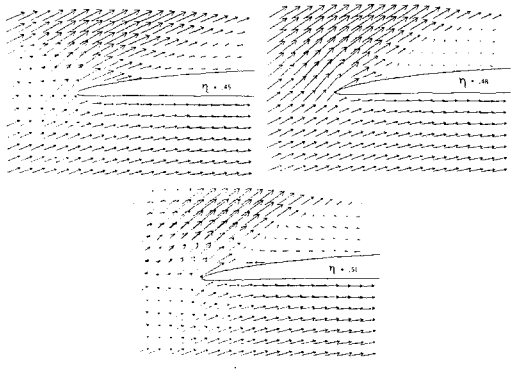

Fig. 7 Fighter configuration, velocity vector field at three consecutive span sections (leading edge area): M = 0.9, α = 24 deg.

shows grids of the two finest levels with Fig. 11 showing the grid resolution of the finest grid in the leading edge area at 26% and 95% of the wing semispan.

Comparisons of the five-level computation by the present method, with a computation using body surface aligned grids[15] (175,000 grid points), and the experiment of Ref. 14 are given in Fig. 12 at x/c of the root chord of 30%, 60%, and 80%. The computation in Ref. 15 is done for wing alone, which may account for the discrepancy in C_p values at the lower surface, for the 30% section.

Supersonic Flow

Results of computations at supersonic Mach numbers are presented for three different forebody configurations: blunt nose, pointed nose, and slightly rounded nose. For all these cases the flowfield consists of two main regions. The first region is upstream of the bow shock, where the flow is un-

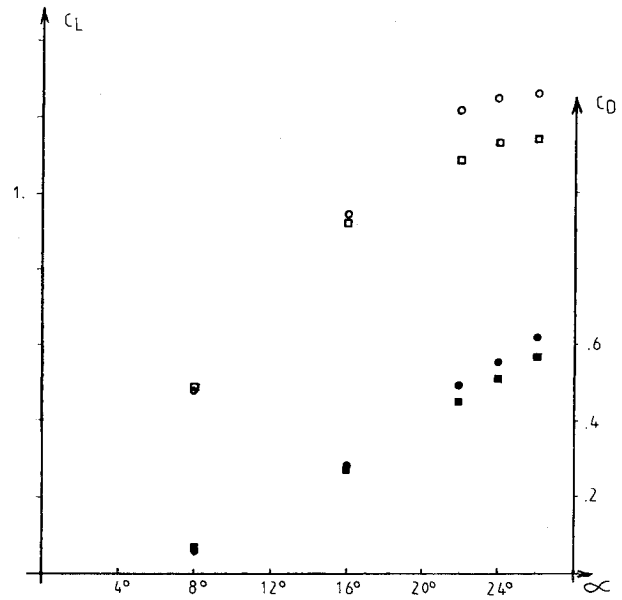

Fig. 9 Fighter configuration, C_L and C_D vs α: ○ = C_L experiment,[13] □ = C_L calculation, ● = C_D experiment,[13] ■ = C_D calculation.

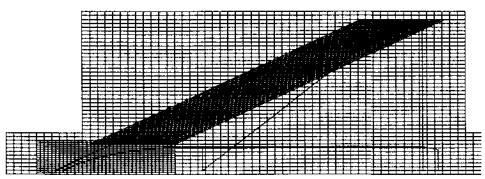

Fig. 10 Delta wing, planform with grids of the fourth and fifth levels.

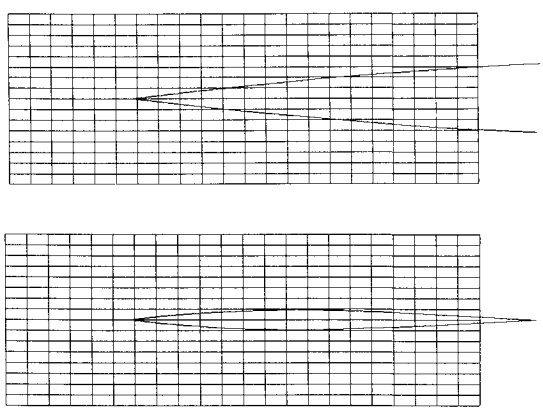

Fig. 11 Delta wing. Grid resolution of the finest grid in the leading edge area: a) Near root, η = 0.26 (covering 21% of the chord); b) Near tip, η = 0.95 (covering 86% of the chord).

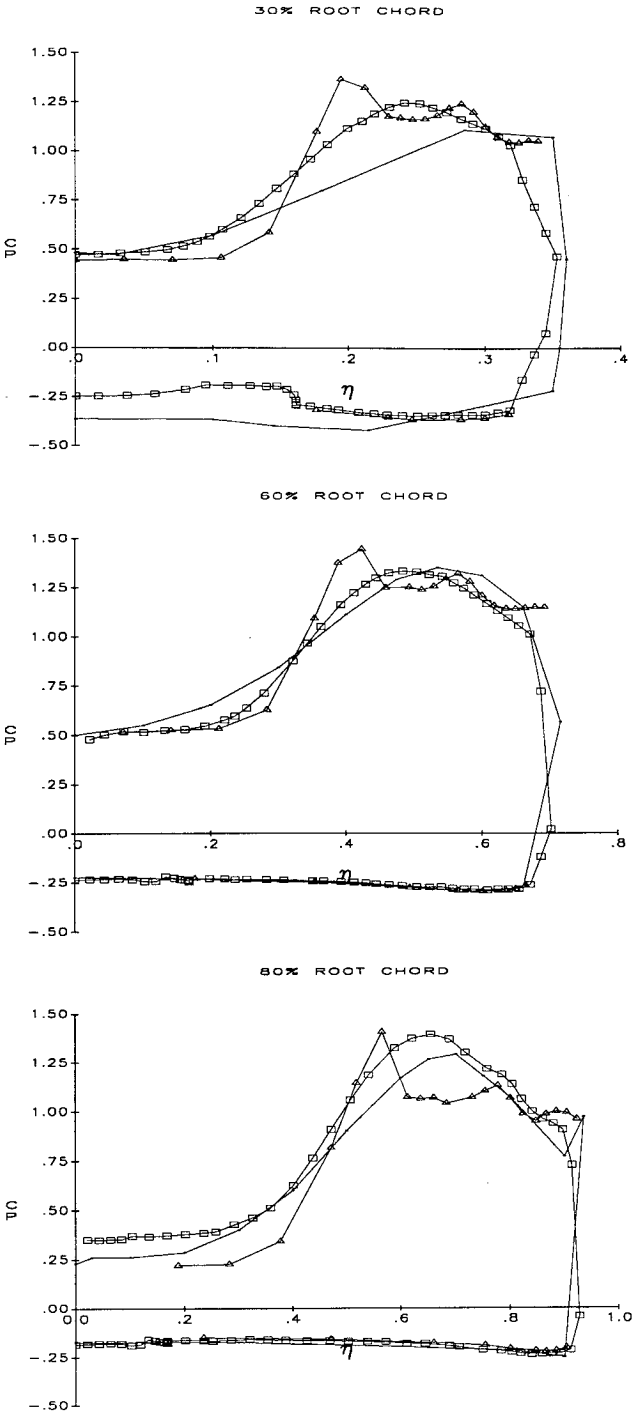

Fig. 12 Delta wing. C_p distribution at x/c of the root chord of 30%, 60%, and 80%: □ = present computation, —— = computation,[15] △ = experiment.[14]

disturbed, and the second, after the bow shock, where the flow is disturbed and the velocity may be subsonic or supersonic. If the velocity after the shock is supersonic, the flowfield is determined only by events upstream (neglecting viscous effects). Bow shocks are attached or detached depending on the sharpness of the nose. The results include pressure distributions on the configuration surface, and the computed flowfield visualization where the location of the bow shock can clearly be seen. For the blunt nose the computations are compared with theoretical results, whereas for the pointed and rounded noses comparison is with wind tunnel tests. Nominal geometries are used in all computations; no boundary-layer corrections are made.

Blunt Nose

The configuration is a hemisphere connected to a frustum of half-angle 5 deg, followed by a uniform cylinder. In this case the bow shock is detached for any supersonic Mach number. After the bow shock, in front of the nose, there is a subsonic pocket. The computation has been done for M = 1.4 at zero angle of attack and used five grids at five multigrid levels. On the axis, the bow shock is normal to the axis, hence the stagnation C_p after the shock is known to be 1.5. The calculated C_p at the nose apex is equal to 1.4. Also, the stand-off distance along the axis between the bow shock and the nose, is known theoretically for a sphere.[16] This location is indicated in Fig. 13, which shows the extent of the subsonic

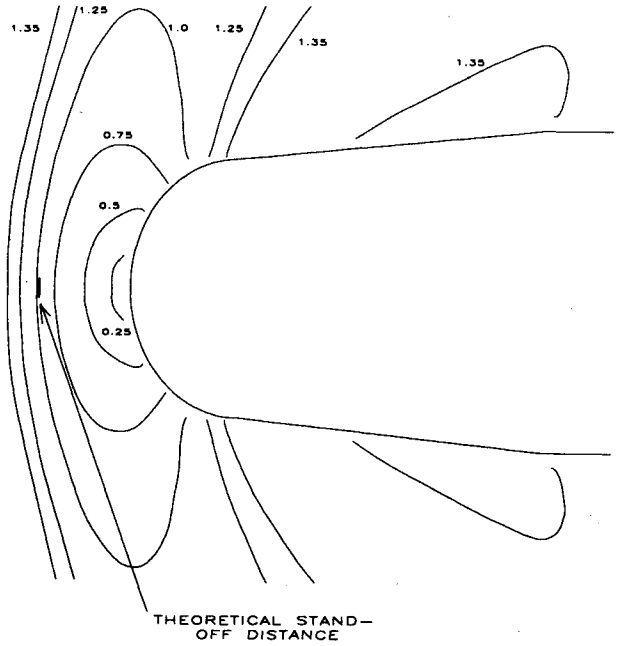

Fig. 13 Blunt nose. Mach contours at $M = 1.4$, $\alpha = 0$ deg.

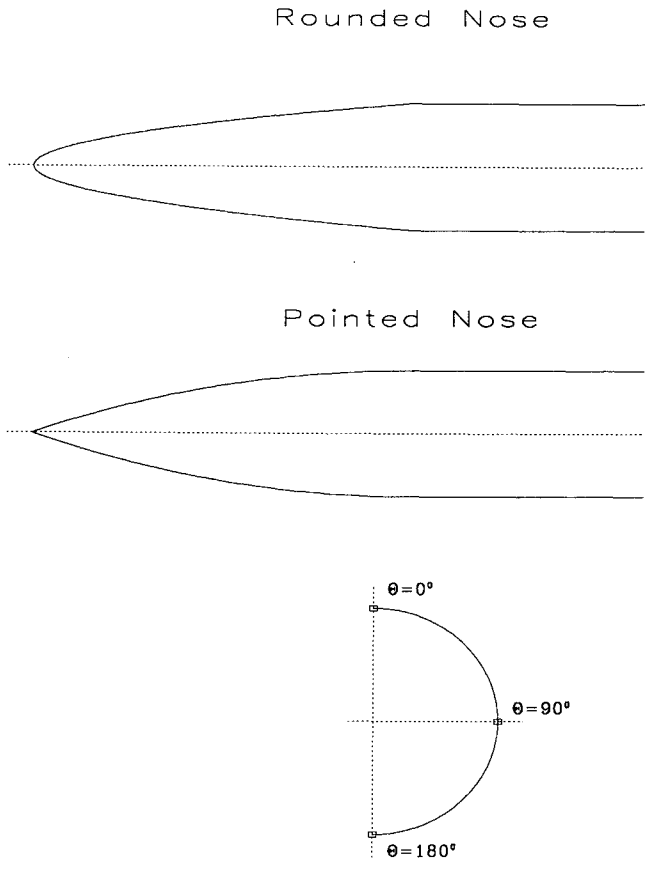

Fig. 14 Geometry of forebodies.

pocket, in the case of a sphere. The fact that in the present case the aftbody differs from a sphere may be responsible for a slight increase in the calculated stand-off distance.

Pointed Nose and Rounded Nose

The following two configurations are more important for practical applications. The generating line of the pointed forebody is a segment of an arc connected tangentially to a horizontal line, and the generating line of the rounded nose

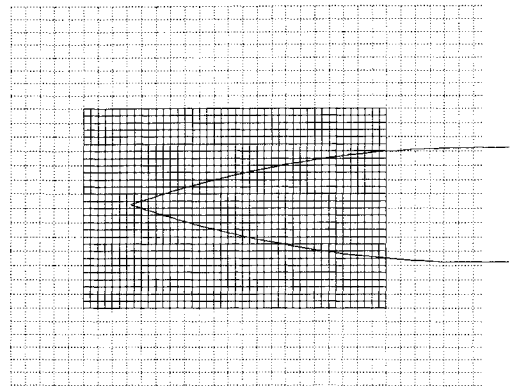

Fig. 15 Pointed nose. The configuration with grids of levels four and five.

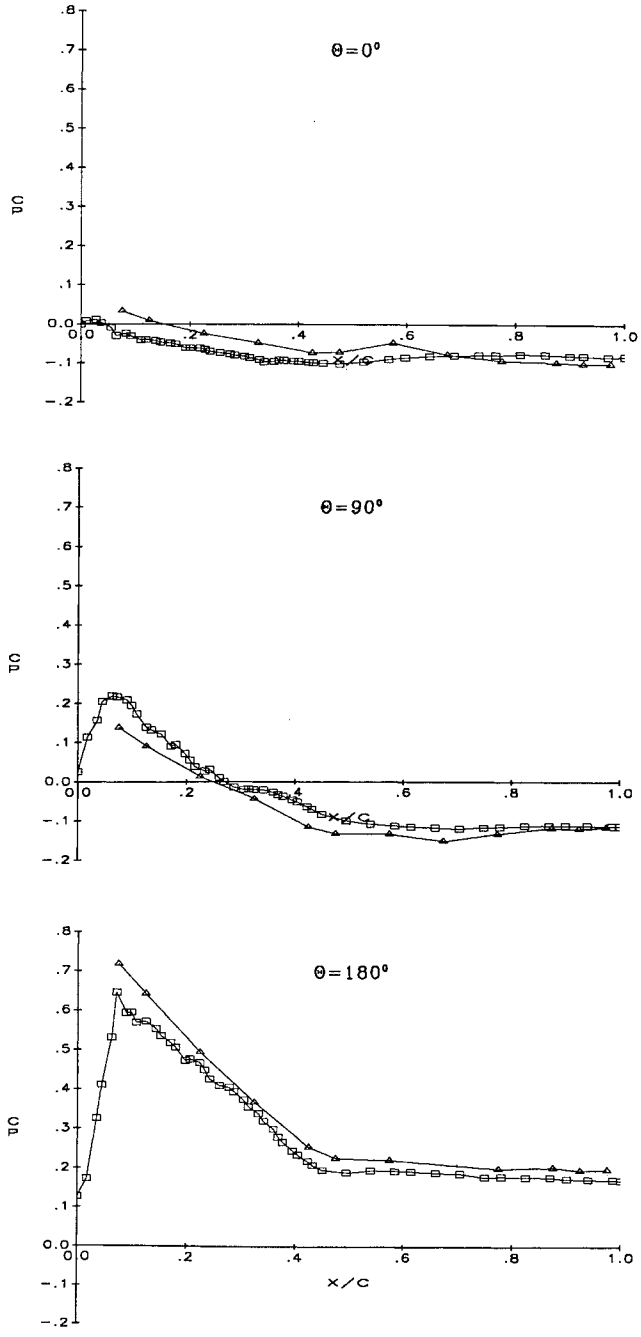

Fig. 16 Pointed nose. $M = 2.3$, $\alpha = 20$ deg. C_p distribution at $\Theta = 0$, 90, and 180 deg; □ = computation, △ = experiment.[17]

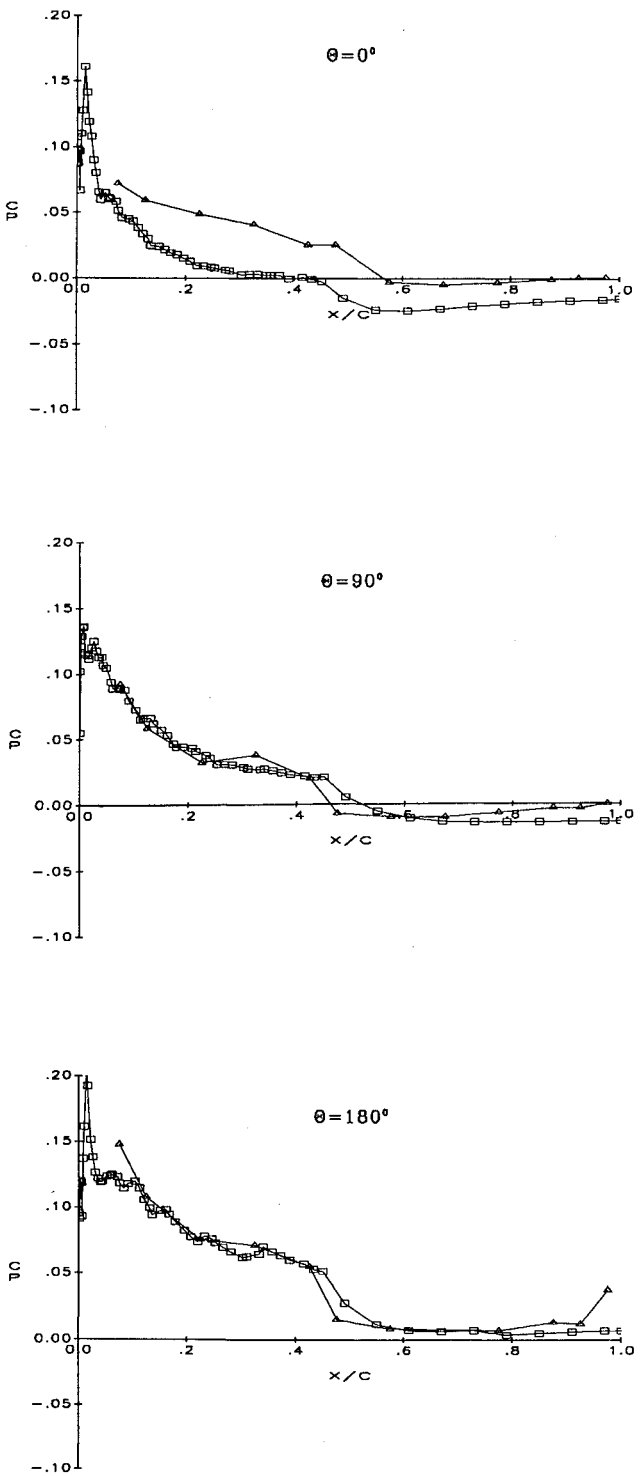

Fig. 17 Rounded nose. $M = 4.6$, $\alpha = 4$ deg. C_p distribution at $\Theta = 0$, 90, and 180 deg; □ = computation, △ = experiment.[17]

is a segment of a square root function connected to a horizontal line (not tangentially). The exact geometries are described in Ref. 17, referred to as the circular-arc-cylinder and the blunt-nose-cylinder, respectively. Ref. 17 also includes tables of C_p values measured during wind tunnel tests for longitudinal lines at various circumferential angles around the body. Figure 14 shows the two geometries. Comparisons of the computationl results with the wind tunnel results are presented for three circumferential angles (0, 90, and 180 deg) as indicated in Fig. 14. The computation used five multigrid levels. The pointed nose configuration, together with the grids at levels four and five, are shown in Fig. 15.

Figure 16 shows a comparison with experiment for the pointed nose configuration at flight conditions $M = 2.3$, $\alpha = 20$ deg. Figure 17 shows comparison for the rounded nose configuration at $M = 4.6$, $\alpha = 4$ deg. Because of the nontangential join between the square root surface and the cylinder there is a corner on the rounded nose configuration. This causes an expansion fan centred at this corner in the flowfield, which manifests as a jump in the C_p distribution (see Fig. 17).

Conclusions

A new computational procedure has been devised to solve the Euler equations for three-dimensional flows about arbitrary aircraft configurations. The method is based on multigrid approach, local refinement, and exclusive use of Cartesian (rectangular or sheared) grids. This avoids the interactive construction of body-fitted meshes, a notoriously difficult and time-consuming process for complicated geometries. As a result, turn-around times needed for practical engineering applications have been reduced considerably.

To ensure the accuracy of the computational scheme with nonaligned grids, a carefully formulated implementation of the body-boundary condition is used. Computation examples show a fair agreement with wind-tunnel experiment in different flight regimes, including vortex separation from the leading edge of swept wings and high supersonic flow.

Acknowledgments

A. Brandt's consultations in Multigrid have been very useful in the design of the method. T. D. Rubin took part in computational work and in discussion of results.

References

[1]Fritz, W., "Numerical Grid Generation around Complete Aircraft Configurations." AGARD 58th Meeting on Applications of Computational Fluid Dynamics in Aeronautics, Aix-en-Provence, France, Apr. 7–10 1986.

[2]Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA 24th Aerospace Sciences Meeting, AIAA Paper 86-01103, Reno, NV, Jan. 1986.

[3]Jameson, A., Baker, T. J., "Improvements to the Aircraft Euler Method." AIAA 25th Aerospace Sciences Meeting, AIAA Paper 87-0452, Reno, NV, 1987.

[4]Gaffney, R. L., Hassan, H. A., and Salas, M. D., "Euler Calculations for Wings using Cartesian Grids." AIAA 25th Aerospace Sciences Meeting, AIAA Paper 87-0356, Reno, NV, 1987.

[5]Luntz, A. L., and Epstein, B., "A Multigrid Full Potential Transonic Code for Arbitrary Configurations." GMD-Studien 110, *Proceedings of the Second European Conference on Multigrid Methods*, Köln, Germany, 1985, pp. 103–110.

[6]Epstein, B., Luntz, A. L., and Nachshon, A., "Multigrid Transonic Computations about Arbitrary Aircraft Configurations." *Journal of Aircraft*, Vol. 26, No. 8, 1989, pp. 751–759.

[7]Jameson, A., and Baker, T. J., "Solution of the Euler Equation for Complex Configurations." AIAA 6th Computational Fluid Dynamics Conference, AIAA Paper 83-1929, Danvers, MA, 1983.

[8]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Method Using Runge-Kutta Time-Stepping Schemes." AIAA 14th Fluid and Plasma Dynamics Conference, AIAA Paper 81-1259, Palo Alto, CA, 1981.

[9]Jameson, A., "The Evolution of Computational Methods in Aerodynamics." *Journal of Applied Mechanics*, Vol. 50, Dec. 1983, p. 1063.

[10]Kreiss, H. O., "Initial Boundary Value Problem for Hyperbolic Systems." *Communications on Pure and Applied Mathematics*, Vol. XXIII, No. 3, 1970, pp. 277–298.

[11]Brandt, A., "Multi-Level Adaptive Computations in Fluid Dynamics." *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 100–108.

[12]Lax, P., and Wendroff, B., "Systems of Conservation Laws." *Communications on Pure and Applied Mathematics*, Vol. XIII, No.

2, 1960, pp. 217–237.

[13]Engineering Division Document 860447, Israel Aircraft Industries, Ben Gurion Airport, Israel, June 1986.

[14]US/European Vortex Flow Experiment. Test Report of Wind Tunnel Measurements on the 65° Wing in the NLR High Speed Wind Tunnel HST, National Aerospace Laboratory, TR 85046 U, The Netherlands, May 1985.

[15]Hitzel, S. O., Wagner, B., and Leicher, S., "Euler-Simulation of the Vortex-Flow-Experiment. A Critical Consideration," *Proceedings of the Symposium on Euler Code Validation*, Aeronautical Research Inst., Stockholm, Sweden, Oct. 1986.

[16]Liepmann, H. W., and Roshko, A., "Elements of Gasdynamics," Wiley, 1957, p. 105.

[17]Landrum, E. J., "Wind-Tunnel Pressure Data at Mach Numbers from 1.6 to 4.63 for a Series of Bodies of Revolution at Angles of Attack from 4° to 60°," NASA TM X-3558, Washington, DC, 1977.