Multigrid Transonic Computations About Arbitrary Aircraft Configurations

B. Epstein,* A. L. Luntz,† and A. Nachshon‡
Israel Aircraft Industries, Lod, Israel

A three-dimensional full-potential code that is able to handle arbitrary aircraft configurations is described. This code is based on the use of Cartesian grids, local refinement, and multigrid calculations. The configuration is treated as an assembly of elements (bodies, wings, pylons, etc.). The algorithm incorporates the body boundary condition implementation for nonaligned grids. Multigrid treatment of local refinement and grid overlapping is described. Three application cases are discussed. First, a fighter configuration including body, wing, fairing, and canard; this example shows the importance of the grid-to-shock alignment. Second, a fighter configuration including body, wing, fairing, canard, and external fuel tank; the analysis shows an unexpectedly strong interference between body, wing, and fuel tank. Third, a civil aircraft configuration with different wing root fairings.

Nomenclature

a = $[1/M^2+(\gamma-1)(1-u^2-v^2-w^2)/2]^{0.5}$ – local sound speed

$B(D,R,\{F\})$ = body boundary condition operator

C = airfoil chord length

C_D = drag coefficient

C_L = lift coefficient

C_p = pressure coefficient

C_P^* = C_p critical for local Mach number equal to 1

CW = calculated potential jump at the trailing edge of a lifting surface

D = dummy point

E^H = linear extrapolation operator for grid H

$F,F(x,y,z)$ = full potential

G^H,G^h = grid boxes (volumes) of grids H, h

H,h = symbols of grids. Usually (but not always), H symbolizes coarse grid, h fine grid

I_h^H = trilinear interpolation from the vertices of a grid cell into a point inside the cell. I_h^H is the interpolation from grid h to grid H. This type of interpolation is used for coarse-to-fine and fine-to-coarse grid interpolation, as well as in overlapping

L,LF = full-potential differential operator

L^H = finite-difference approximation to L at grid H

M = freestream Mach number

n = vector normal to the body surface, directed outside

P = tangent plane to the surface

R,RR = reference point (to dummy D): a point outside the aircraft to whose template D belongs

$SB(D,R,F)$ = shortened boundary condition operator, appointing to D the potential value at its reference point R

T = airfoil thickness

t = coordinate along the normal n

u,v,w = local velocity components, $u=F_x$, $v=F_y$, $w=F_z$

W_{old}^h = the value of the potential jump W in grid h, from the previous iteration

x,y,z = rectangular Cartesian coordinates; x along the aircraft axis, downstream; y up; z spanwise

α = angle of attack

γ = the ratio of specific heats; $\gamma=1.4$ is used in the code

η = span station in units of the wing semispan

ϕ = $F-x\cos\alpha-y\sin\alpha$, reduced potential

Introduction

In recent years, 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 ways have been used to overcome these difficulties (for various approximations to the complete flow equations, including potential and Euler). One of the methods creates smooth local grids aligned to the body surface and then tailors them into one universal grid (e.g., Ref. 1 in the framework of finite elements, Ref. 2 with finite differences and finite volumes). This approach is virtually unlimited in handling complicated configurations, but is far from being fully automated (it usually demands months for an experienced engineer to complete a grid about a new complicated configuration).

Another way has been explored recently[3,4] using unstructured (but still 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. There are still some difficulties to overcome in order to make this very promising (and fully automatic) approach really universal for three-dimensional configurations. It also has all the drawbacks of fully nonregular grids.

The third way is the use of Cartesian grids. They provide a simple and good 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 several other serious problems, the first being the body boundary condition implementation. Several efforts have been made to use Cartesian grids for arbitrary configurations (e.g., Refs. 5–7). In these works, the authors see the creation of one single code able to handle arbitrary three-dimensional configurations as their main goal. To the best of the author's knowledge, this goal has not been achieved as yet.

Presented as Paper 88-4.7.3. at the 16th International Council of Aeronautical Sciences, Jerusalem, Israel, Aug. 28–Sept. 2, 1988; received Dec. 6, 1988; revision received Feb. 22, 1989. Copyright © 1988 by ICAS and AIAA. All rights reserved.

*Senior Research Scientist, Head of CFD Group.

†Senior Research Scientist, CFD Group.

‡Research Scientist, CFD Group.

The present paper describes a three-dimensional full-potential code that 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. The 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 solution to the problem of body boundary condition implementation in nonaligned grids has been found, as may be seen from the results. A given aircraft configuration is handled as a Boolean sum of bodies. The basic features of the code have been described in previous publications.[8,9] However, this paper starts with a description of the algorithm for completeness and also to provide some details not previously described.

The main purpose of this paper is to show the abilities of the code in solving problems of analysis and design of practical configurations. As the code solves full-potential flows, examples have been chosen where the flow is mainly potential. At the present time, a three-dimensional Euler code based on similar principles is completed and under evaluation, and is to be described in a future publication.

Three application cases are described in the present paper: 1) A fighter configuration including body, wing, fairing, and canard; this example shows the importance of the grid-to-shock surface alignment, which is possible with the flexible choice of local grids (and sometimes impossible in codes with fixed grids); 2) A fighter configuration including body, wing, fairing, canard, and external fuel tank; the analysis shows an unexpectedly strong interference between body, wing, and fuel tank; 3) A civil aircraft configuration with different wing-root fairings; this case shows the role of the fairing on the wing-pressure distribution.

Equation and Numerical Scheme

The equation solved is the "nonconservative" full-potential equation

$$LF \equiv (a^2 - u^2)F_{xx} + (a^2 - v^2)F_{yy} + (a^2 - w^2)F_{zz} - 2uvF_{xy} - 2uwF_{xz} - 2vwF_{yz} = 0 \qquad (1)$$

In the actual computation, the reduced potential ϕ is used.

Thus, the problem solved is inviscid. The boundary layer, if needed, has to be added to the aircraft shape prior to the potential calculation. The wake of a lifting surface is assumed to be composed of straight lines parallel to the x axis starting from the trailing edge of the lifting surface. The boundary conditions at infinity are $\phi = 0$ everywhere except the downwind ($x = +\infty$) infinity, where $\partial\phi/\partial x = 0$ is imposed. The boundary condition at the aircraft surface is the homogenous Neumann

$$\partial F/\partial n = 0, \; n - \text{ normal to the surface} \qquad (2)$$

Besides rectangular grids, grids with sweep and dihedral are used. Equation (1) is discretized using symmetrical schemes for derivatives at points that appear subsonic ($u^2 + v^2 + w^2 < a^2$) in the last iteration made and streamwise upwind schemes at supersonic points. The system of finite-difference equations is iterated by SLOR in x and/or y directions, with the Seidel-type sweep.

The "nonconservative" form of the equation and the discretization has been chosen mainly for technical reasons. In any event, it was unrealistic to keep the conservation features with the complicated surface boundary condition implementation used, valid for arbitrary configurations (following section), and with the local refinement and overlapping of grids not aligned to each other. In general, we see the conservation form as having a limited value in the framework of a potential model. First, the conservation usually applies to the mass, not to the momentum or energy. Second, the usually accepted "viscous correction" – the addition of the boundary layer prior to inviscid calculations – makes the global mass conservation outside the enlarged body meaningless. Third, in the case of thick wing trailing edge or base (of a projectile), mass injection is used (as in the present code) or a thick wake is excluded from the computational space (which is equivalent to injection, in terms of conservation). As for the shock resolution, it is usually restricted by factors other than conservativity (as discussed in connection with the first application case). The comparison with experiment (see first and third application cases) shows that the present method is able to provide a satisfactory shock resolution.

Discretized Boundary Conditions

With nonaligned grids, the successful implementation of the body boundary condition is a major condition for the success of the method as a whole. Thorough analysis shows that the finite-volume approach with nonaligned grids,[6,7] although very attractive in some well-defined cases, becomes extremely complicated when the algorithm is supposed to deal with arbitrary configurations. For this reason, the finite-difference approach was chosen.

In this framework, the implementation of the body boundary condition implies introduction of dummy points. At these dummy points, potential values have to be assigned and corrected periodically in the iteration process, to express, together with the field potential values, the approximation to the exact boundary condition Eq. (2). With nonaligned grids, one dummy point D can belong to the templates of several reference points (R, RR; see Fig. 1). The potential value of $B(D,R,\{F\})$ at D will depend on the position of D and its reference point R and on the field potential values $\{F\}$.

The full potential is approximated along $\boldsymbol{n}$ by the function $bt^2 + d$, with zero derivative at $t = 0$, thus satisfying the boundary conditon Eq. (2) at the tangent plane P (and asymptotically at the body surface). The approximation goes as follows. First, several grid points $\{R_i\}$ are chosen in the vicinity of D, all "seen" from R, so that their projections $\{t_i\}$ on $\boldsymbol{n}$ are dispersed enough compared with the mesh size. Second, potential values at $\{t_i\}$ are calculated as those at $\{R_i\}$ corrected for

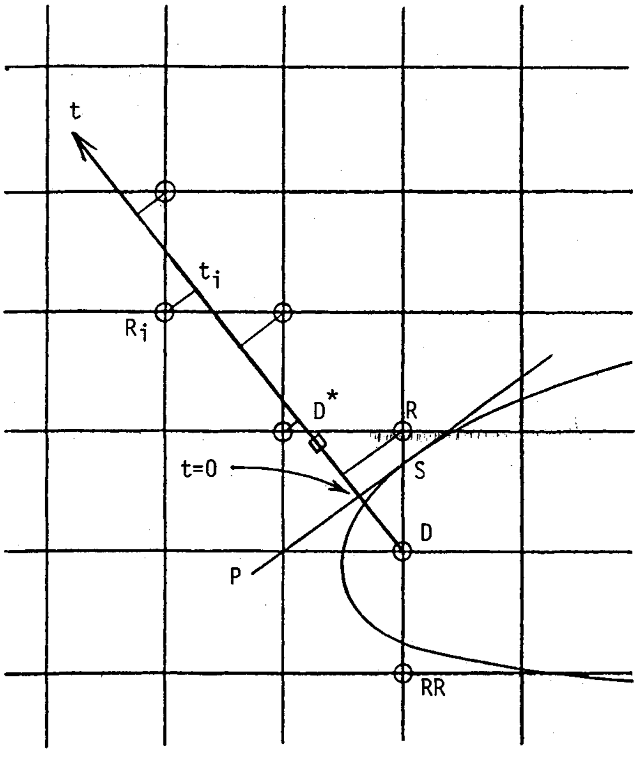

Fig. 1 Body boundary condition implementation with nonaligned grids.

the projection, using the stream velocity at R as calculated at the last iteration used. Third, coefficients b,d are sought as linear functions of the corrected potential values at $\{R_i\}$ using the least-square equations. Finally, these linear functions are substituted into bt^2+d, together with $t=t(D^*)$, D^* being a reflection of D through P. This gives the potential value at D as a linear function of potential values at $\{R_i\}$:

$$B(D,R,\{F\}) = \Sigma b_i\, F(R_i)$$

The coefficients $\{b_i\}$ depend on $\{t_i\}$ and $t(\mathrm{D}^*)$ only. Thus, points $\{R_i\}$ can be selected and coefficients $\{b_i\}$ calculated in the preprocessor for every couple (R,D).

Theoretically, the described procedure may fail at some points (not enough points R_i can be selected, etc.), and then instead of B, the operator SB is used. However, at fine enough girds, B can usually be constructed everywhere. This is the case with all the fine grids in the following application examples.

As to the boundary conditions at the local refinement box limits, they are fixed during the iteration process for every specific local grid as Dirichlet conditions and are corrected each time by the coarse-to-fine correction interpolation in the multigrid process. The exception is for the symmetry plane (if defined), where the reflection condition is used. At the front (upwind) side of any local refinement box, potential values at an additional grid plane are defined during the coarse-to-fine-grid interpolation. These values are used in the upwind schemes, only if and where the local stream velocity appears to be supersonic.

Prepocessor and Computational Code

The code consists of two main programs: PREPRO and MULTIG. The purpose of PREPRO is to translate the geometry of a given configuration in terms of computational grids. It works independently for each grid. The access to the geometry data is made through a number of subroutines that provide the following information: 1) for a given point, the indication whether the point is inside the aircraft or outside; 2) for two given points A and B, A being outside the aircraft, the indication whether B "is seen" from A (in other words, if the interval A-B does not intersect the configuration surface); if A-B intersects the surface, coordinates of the nearest to A intersection point S are provided, together with the components of the outside normal $\boldsymbol{n}$ to the surface at S; 3) location of the trailing edge of a lifting surface.

The configuration is treated as an assembly (Boolean sum) of elements. This means that elements may intersect each other; the given information is sought for each element independently, and then the set of results is used to choose the answer on the configuration as a whole. At present, the following elements are allowed: bodies (fuselage, external fuel tank, bomb, etc.), wings (also canards, horizontal tail, etc.), pylons (also fins), and nacelles. Each element is inputted as a set of curves (cross sections) in parallel planes, each curve as a set of points. Several elements of one type may be present in one configuration.

The PREPRO output, together with the multigrid data (grids, multigrid steps) and parameters (Mach, angle of attack, relaxation parameters, etc.), are the input to MULTIG. In the multigrid calculation process, fine-to-coarse corrections ("right-hand side," *RHS*) are used explicitly for all operators involved, except for the operator $B(D,R,\{F\})$ (with the present method, it seems to be too complicated technically to introduce *RHS* for B). This exception, of course, slows down the convergence rate.

Local Refinement and Multigrid Level Interaction

Local refinement is one of the basic tools of the present method. Its treatment in the framework of multigrid has raised problems, not previously discussed in the literature, which were solved in the present work.

If the fine grid h is a local refinement of a coarse gird H, both the coarse-to-fine and the fine-to-coarse corrections[10] create problems for points inside the fine-grid box G^h near its limits. One of the problems is the fine-to-coarse operator correction *RHS* to the coarse-grid operator:

$$L^H F^H = RHS^H \tag{3}$$

where RHS^H is usually defined as

$$RHS^H = L^H(I_h^H F^h) - I_h^H(L^h F^h - RHS^h) \tag{4}$$

For a coarse-grid point R, which itself belongs to the fine-grid box G^h (so that I_h^H appoints for it a potential value), $L^H(I_h^H F^h)$ may be undefined, because some of the points in the template of R may be outside G^h. Even if all of the template points are inside G^h but some of them are dummy, the formulae $B(D,R,\{F\})$ for their potential-value calculation (as coarse-grid points) may include points outside G^h, so that RHS^H is still not defined by Eq. (4). At such points R, it is usually advised[11] to neglect the RHS^H. Experience shows that this advice is acceptable only when the local refinement ends with the solution becoming smooth, i.e., where RHS^H is small. The present case is different. Local refinement is used wherever better resolution is needed; refinement may end where it is not needed, although the solution may still be not smooth (and the RHS^H large). If so, the RHS^H has to be calculated everywhere in G^h; otherwise the solution process becomes unstable.

To define RHS^H everywhere in G^h, extrapolated field values are used:

$$F_{\mathrm{new}}^h = F_{\mathrm{old}}^H + E^H(I_h^H F_{\mathrm{new}}^h - F_{\mathrm{old}}^H)$$

instead of $I_h^H F_{\mathrm{new}}^h$ for real points and

$$F_{\mathrm{new}}^H(D) = B^H(D,R,\{F_{\mathrm{old}}^H\}) + SB^H(D,R,F_{\mathrm{new}}^H - F_{\mathrm{old}}^H)$$

for dummy, where B^H is the regular boundary condition operator in the coarse grid and SB^H is the shortened boundary condition operator, using as arguments potential values at the coarse-grid real points inside G^h only.

Multigrid Treatment of the Potential Wake Jump

The potential jump W across the wake of a lifting surface is a local value as a function of the span station, but it is an integral value for a fixed span station insofar as it remains constant along the x-coordinate line. For this reason, W needs multigrid treatment somehow different from that of the other functions involved (potential, boundary conditions at infinity, boundary conditions at the symmetry plane, and the grid box sides), whose values are purely local.

For a given span station in a given grid, the jump CW is the difference of potential values extrapolated to the wing surface from grid points above and below the wing near the trailing edge. For stations outside the wing span, $CW=0$. At some stations, direct calculation of the potential jump may be impossible (with coarse grids in the presence of bodies other than wing near the trailing edge, such as engine, external fuel tank, etc.). In such cases, interpolation is needed. For a function $g(k)$ of span station k, defined not necessarily for every k of a given grid (but at least for one of them), the operator $IW(g)$ extends the definition of g to all the span stations, with homogenous Neumann conditions at the box limits (if needed). The coarse-to-fine W interpolation is as follows:

$$W^h = W_{\mathrm{old}}^h + I_H^h(W^H - I_h^H W_{\mathrm{old}}^h)$$

Fine-to-coarse correction to the W calculation is

$$RHSW^H = IW(I_h^H W^h - CW^H) + I_h^H(RESW^h)$$

where

$$RESW^h = IW(RHSW^h - W_{\text{last}}^h + CW^h)$$

At the finest level, $RHSW^h = 0$. The potential jump relaxation (with the iteration process) is then

$$W_{n+1}^H = W_n^H + IW(RHSW^H - W_n^H + CW^H)$$

Overlapping

With local refinement, several grids may be used at the same multigrid level. This may occur for different reasons, such as limited core memory, and, as a result, the need to divide one large grid into several smaller ones or the use of different types of grids at the same refinement level (e.g., a rectangular grid covering a part of the fuselage and the fairing, and a swept grid along the wing leading edge). Some of these grids may overlap. In such a case, computational instability may occur if the coarse-grid G^H receives, at some point R, the RHS^H from one of the overlapping fine grids G_1^h, G_2^h and, after a relaxation with this RHS^H, sends the coarse-to-fine correction to the second overlapping grid.

To avoid this conflicting situation, restriction on the use of overlapping is introduced. For any set of grids $G_1^h, \ldots, G_k^h$ covered by one coarser level grid G^H, overlapping is only allowed such that every grid point of G^H is strictly inside not more than one of the fine grids. For the exchange of information between overlapping grids, grid limit points of each overlapping grid falling inside one of the other overlapping grids are assigned, after each iteration, potential values interpolated from one of those "neighbors," with an appropriate sequence of priorities for these grids.

Choice of Grids

The use of finite-difference approximation is based on the assumption (proven, under some constraints) that asymptotically, with unlimited grid refinement, the solution is independent of the specific type of grids used. In practice, however, only limited grid refinement may be used, and a proper choice of grids may spare the need for additional refinement.

The vast majority of existing transonic codes uses a fixed or nearly fixed set of grids. These grids have been chosen by the designer to provide optimal results in the limits of a given refinement. For example, in many codes tapered grids are used for wings mainly to provide comparable resolution near the root and tip of the wing. The grid-point density is usually increased near the leading edge (at the expense of other parts of the wing) for better interpretation of the fast changes in geometry and flow in this region. These measures ensure optimal results for many (but not all) wing shapes and flight conditions.

The present codes use homogeneous Cartesian (rectangular or swept) grids only. The use of such grids enables one single code to handle arbitrary configurations. Grids not uniformly spaced would complicate the method. Their use, together with uniformly spaced grids, would create some new problems in the framework of multigrid because of nonconstant refinement ratio.

In the present method, additional grid density at specific locations (such as wing tip or leading edge) is achieved through local refinement (Fig. 2). Standard sets of grids are recommended for a large number of configuration types to enable a project engineer to use the code with no need to choose the grids. At the same time, the option of flexible choice of grids in the present method is of great help with complicated, new, or unusual configurations or flight conditions. It enables additional resolution at any specific location of the configuration.

Fighter Configuration, Grid-to-Shock Alignment

This application case deals with a wing-body-fairing-canard fighter configuration. Computations were made for $M = 0.9$, $\alpha = 6$ deg (which corresponds to approximately $C_L = 0.32$). For these flight conditions, the experiment shows no signs of developed separation. Thus, good agreement may be expected between experiment and full-potential calculation. However, the computation using five grid levels with swept grids (Fig. 3a) on the finest (fifth) level shows a completely smeared shock (Fig. 4). Similar results are given by Jameson's FL022 wing-only full-potential code (Fig. 4 shows the comparison at the $\eta = 0.67$ span station, outboard to the canard).

From the experiment, one may learn that the shock line at the wing upper surface is nearly parallel to the wing trailing edge, thus making a large angle with the swept spanwise grid

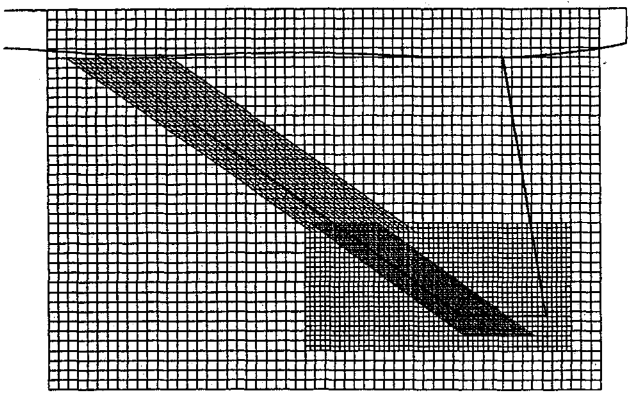

Fig. 2 Example of local grid refinement at wing tip and leading edge.

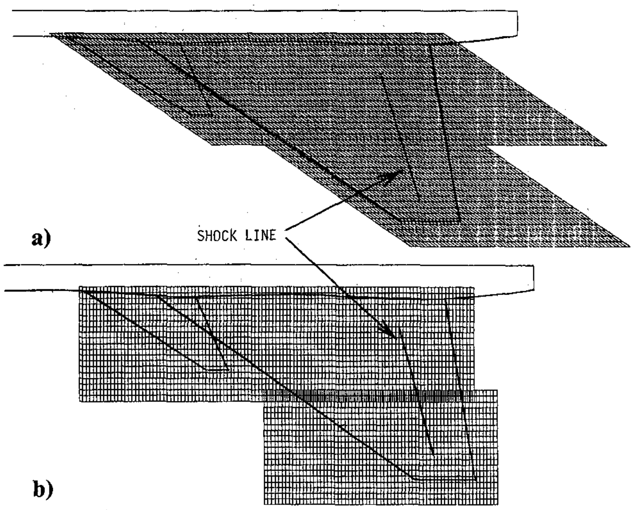

Fig. 3 Fighter planform, shock position, and local refinement grids: a) swept, b) rectangular.

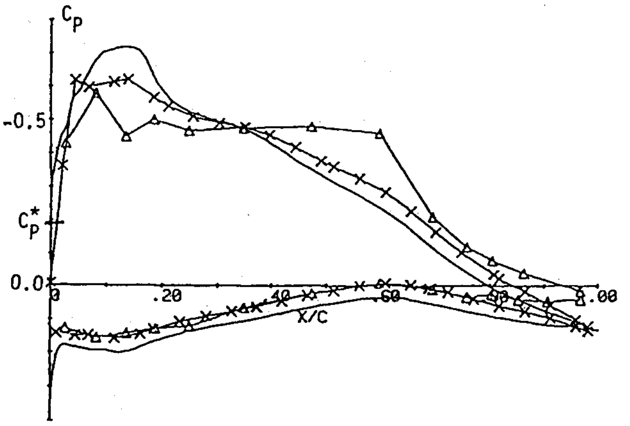

Fig. 4 Fighter configuration. Wing C_p distribution, $\eta = 0.67$: –Δ–Δ– experiment, ——— FL022, –×–×– present method, swept grids.

lines of the finest multigrid level (Fig. 3a). The template of any grid point near the shock uses points lying far ahead and behind the shock (Fig. 5a), which leads to the shock smearing in computation. With rectangular grids (Fig. 3b) used at the fifth level instead of swept grids, the corresponding template points (Fig. 5b) are much nearer to the shock. This may be the reason for a completely different, good agreement with the experiment (Fig. 6).

Flexible choice of local grids gives the possibility of an approximate grid-to-shock alignment for every configuration. It should be emphasized that no full grid-to-shock alignment is needed for a satisfactory shock resolution. Sometimes preliminary calculations may help to find appropriate local grids.

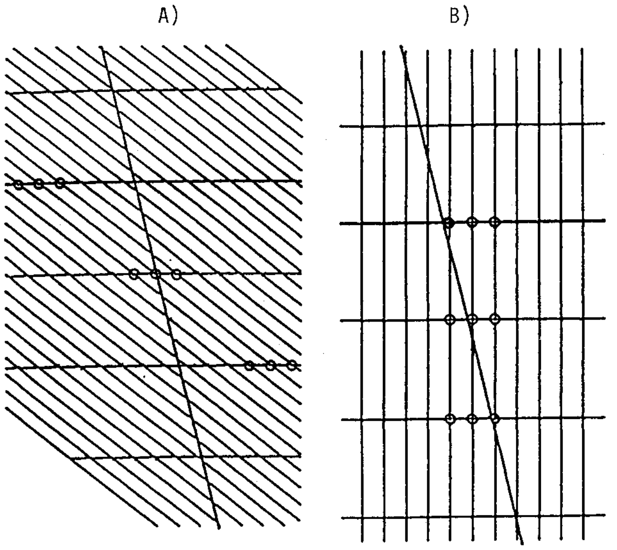

Fig. 5 Shock position, local grid refinement, template (symmetrical scheme): a) swept, b) rectangular.

Fighter Configuration with External Fuel Tank

This configuration includes fuselage, wing, canard, and a large external fuel tank (Fig. 7). The geometry used in the computation is an exact description of the wind-tunnel model, except for the exclusion of the fuel tank pylon. This computation used six grid levels, with the grids of the finest (sixth) level around the fuel tank and the part of the wing above it (Fig. 8). The computation was done at the cruise flight conditions $M = 0.9$, $\alpha = 2$ deg. With this configuration and these flight conditions, the presence of the boundary layer does not essentially affect the computational results. As could be expected, the shock originated by the irregularity of the fuel tank shape is amplified by the wing-tank close coupling and appears both on the fuel tank upper surface and on the neighboring wing lower surface (Figs. 9a and 9b). Less expected was the effect of the fuselage-fuel tank coupling in extending the shock from the tank to the fuselage (Fig. 9c).

At the fuel tank, the shock appears where the curvature of the tank centerline cut increases and then suddenly drops. As increased curvature of a convex surface usually accelerates the

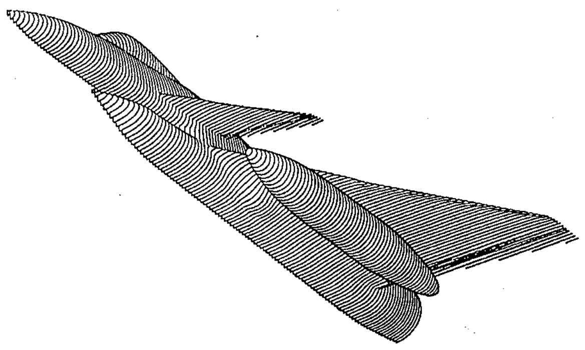

Fig. 7 Fighter configuration with external fuel tank.

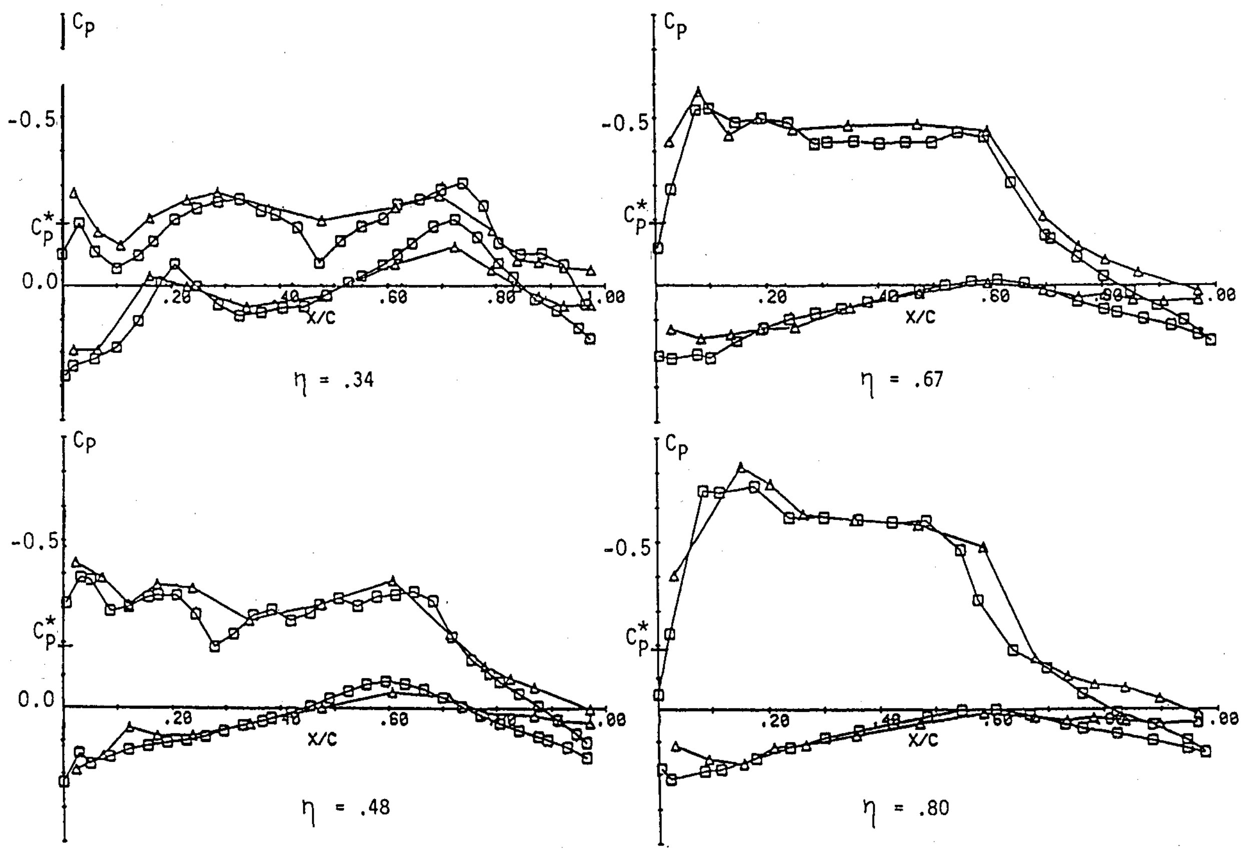

Fig. 6 Fighter configuration, C_p distribution at four wing sections: $M = 0.9$, $\alpha = 6$ deg: –Δ–Δ– experiment, – □ – □ – present method, rectangular grids.

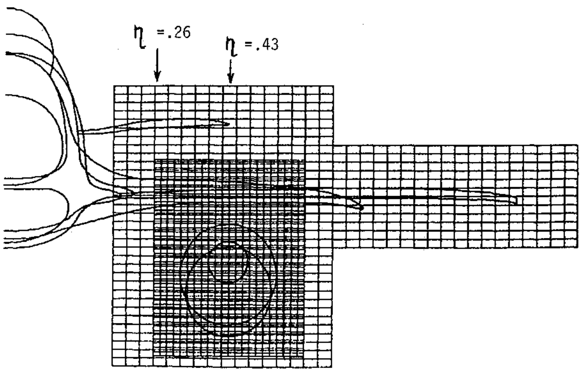

Fig. 8 Fighter with fuel tank (cross sections), with grids of two finest levels, front view.

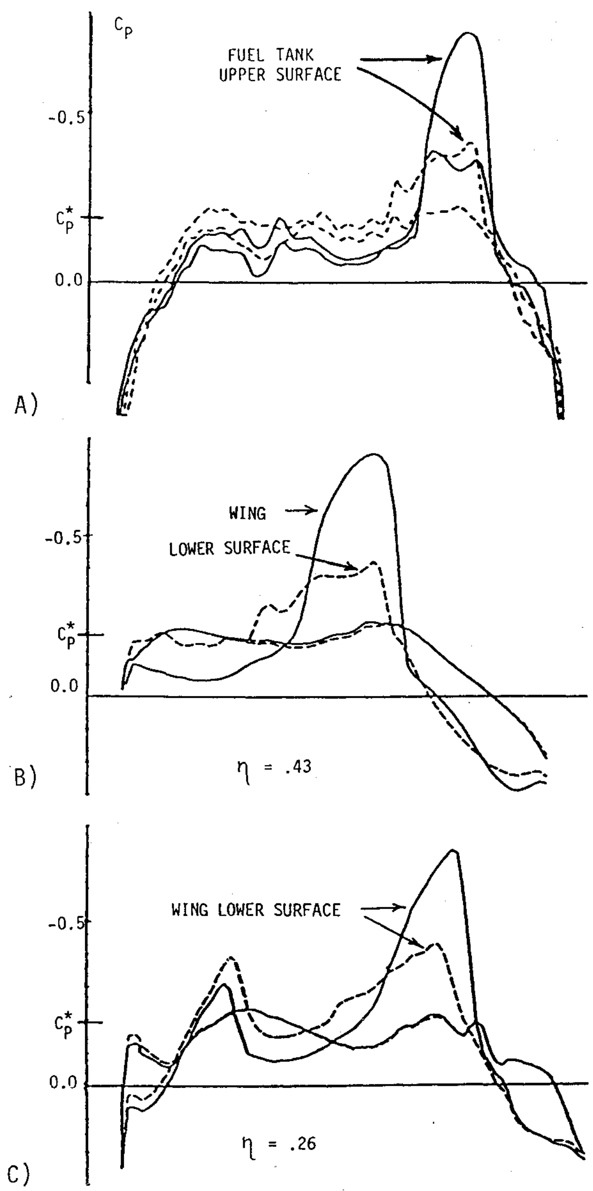

Fig. 9 Fighter with fuel tank. C_p distribution: a) tank centerline, b) wing section above tank, c) section inboard to tank: ——— original tank, ------ modified.

external flow, the fuel tank shape was modified to smooth its curvature (Fig. 10). With the modified tank shape, the calculated C_D of the configuration was reduced by 24 counts, out of approximately 200 counts of form drag of the vehicle. The

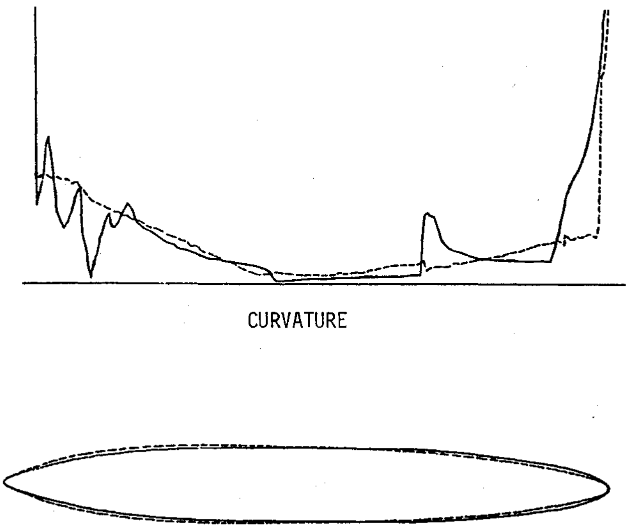

Fig. 10 Fuel tank centerline cut and its curvature: ——— original shape, ------ modified.

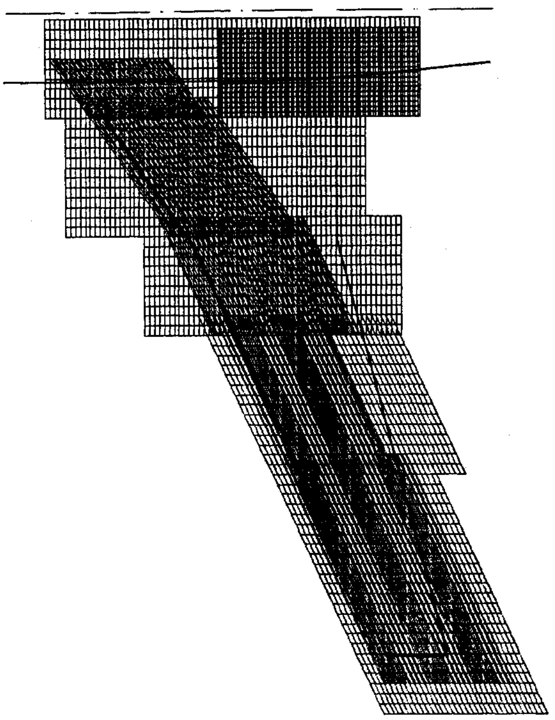

Fig. 11 Civil aircraft, wing planform with the grids of the three finest levels.

major part of this reduction is due to the weakening of the wing/tank/fuselage interaction-induced shock (Fig. 9). On the isolated fuel tank, the modification has reduced the drag by four counts only. Six multigrid levels, with a total of 16 grids and about 400,000 grid points, have been used. Computation for one flight point (M, α fixed) takes about 150 min CPU time on a CDC 875. The experiment confirmed the computational prediction: with $\alpha = 2$ deg and Mach number between 0.9 and 0.95, the drag reduction is between 20 and 25 counts, out of about 330 counts of total drag.

Civil Aircraft, Wing-Body Fairing

The civil aircraft configuration includes fuselage, low-mounted supercritical wing, and fairing. The aspect ratio is 8.8 and the midchord line sweep is 20 deg. T/C is 13% at the root section, 10% at the wing crank, and 9% at the wing tip.

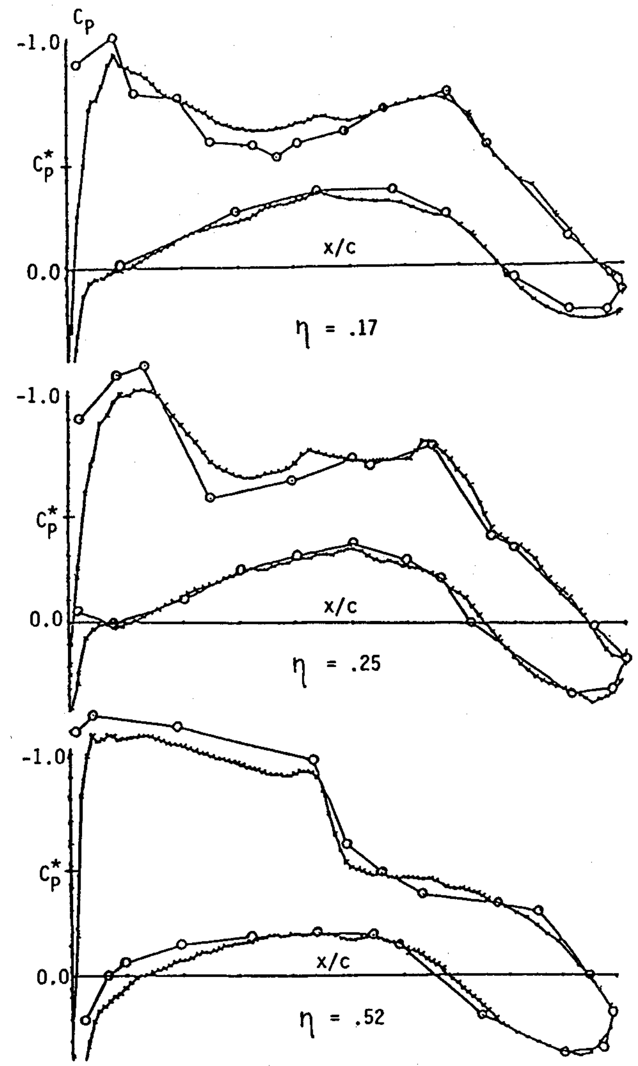

Fig. 12 Civil aircraft, *Cp* distribution at several wing sections: $M = 0.8$, $\alpha = 2$ deg. –○–○– experiment, –×–×– present method.

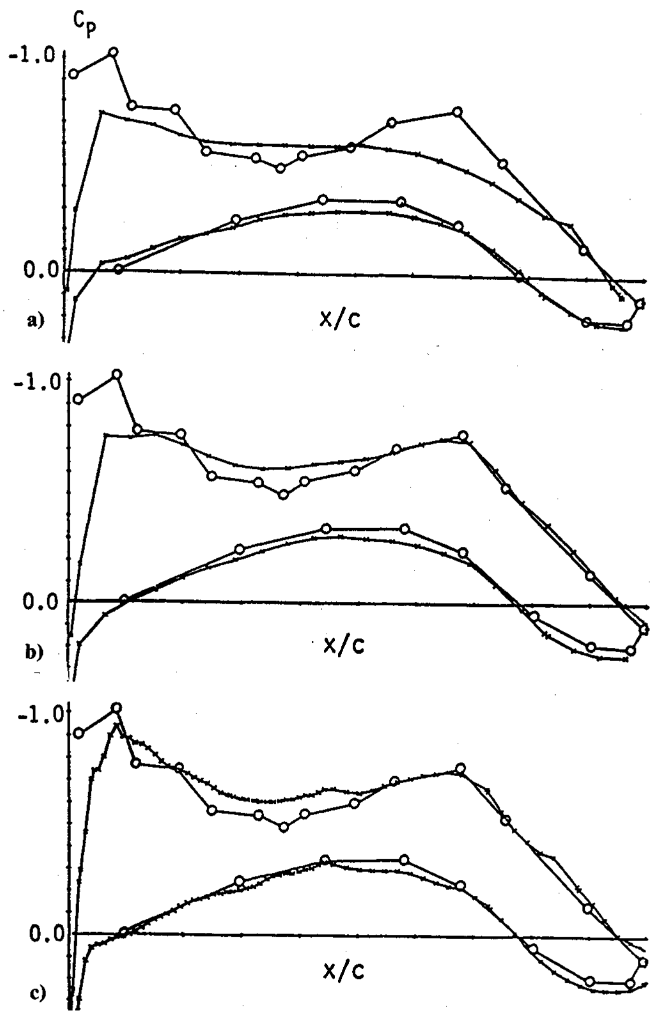

Fig. 13 Civil aircraft, C_p distribution at $\eta = 0.17$, calculated at a) fifth level, five levels used, b) fifth level, seven levels used, c) seventh level, seven levels used: –○–○– experiment, –×–×– present method.

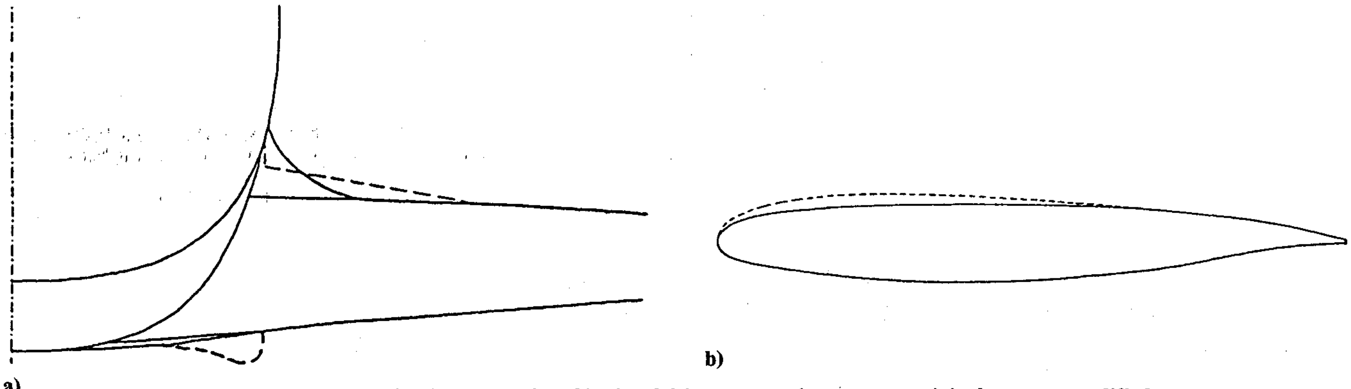

Fig. 14 Civil aircraft: a) cross section, b) wing-fairing root section: ——— original, ------ modified.

The wing twist is +3 deg at the root, −1.5 deg at the tip. A wind-tunnel experiment has been made with the model at $Re = 3.5 \times 10^6$.

The boundary layer was calculated using our iterative FL022 – Integral Boundary-Layer procedure and then added to the wing-fairing geometry (original and modified) prior to the computations. Computations used seven grid levels at the body, fairing, and inboard wing and eight levels at the outboard wing (Fig. 11). Comparison shows good agreement between computation and experiment (Fig. 12). Figure 13 shows, for the example of station $\eta = 0.17$, the impact of grid refinement and the two-way interaction between coarse and fine grids in the multigrid process.

Pressure distribution at the inboard sections (including fairing) shows a double shock, both in computation and experiment. It is due to several features of the geometry:

1) The root-profile upper surface is too flat, between 20 and 50% of the chord. This leads to a lower stream velocity in that region and creates a "valley" in the C_p distribution (Fig. 12, $\eta = 0.17$).

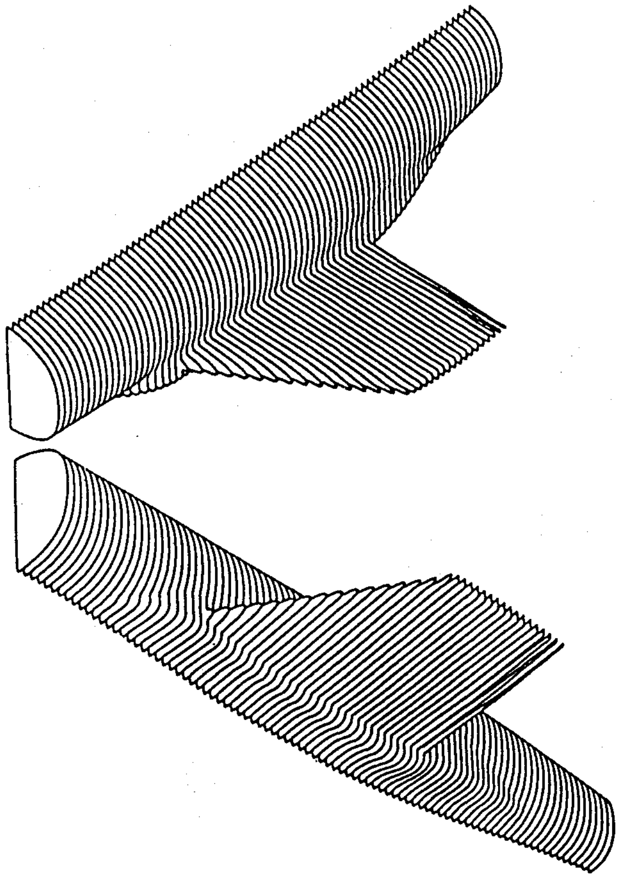

Fig. 15 Civil aircraft, the modified fairing.

2) The curvature at the upper surface rear part of the root profile is too high, thus overaccelerating the flow.

3) Another source of the flow overacceleration in this region is the curvature of the fairing on the side of the body.

4) The third source of the flow acceleration at the root trailing edge is the low position of the wing and the crossflow from the wing to the underneath of the body.

Modification of the fairing was made to eliminate or diminish these undesirable features of the geometry. To increase the resolution of the airflow computations supporting this redesign, a new grid was added at the fine, seventh grid level in the fairing/trailing-edge region (Fig. 11). Calculations were made at $M=0.8$, $\alpha=1.5$ deg (both C_L and M slightly higher than cruise conditions). The new fairing (Figs. 14a and 15) consists of two elements. The first is a local "wall," eliminating feature 3 and decreasing feature 4 (Fig. 14 and Fig. 15). The second element is an addition to the inboard wing section, to correct features 1 and 2 (Fig. 14a).

Figure 16 shows the new C_P distribution at several inboard sections as compared to the previous distribution. The C_P distribution improved considerably even beyond the crank ($\eta=0.37$), far outside the fairing (which ends at $\eta=0.17$). The resulting computed decrease in C_D is 17 counts, out of a total of about 200 counts of form drag.

Calculation for both configurations – with the old and new fairing – used the same set of grids. Eight multigrid levels were used with a total of 20 grids and about 300,000 grid points. Computation for one flight point takes about 120 min CPU time on a CDC 875.

Conclusions

A three-dimensional full-potential code that is able to handle arbitrary aircraft configurations is described. This code is based on the use of Cartesian grids, local refinement, and multigrid calculations. The configuration is treated as an assembly of elements (bodies, wings, pylons, etc.). The algorithm incorporates the solution of the problem of body boundary condition implementation for nonaligned grids. Some other new problems, such as the multigrid treatment of local refinements and grid overlapping, had to be resolved during the code design. Three application cases are described:

1) A fighter configuration including body, wing, fairing, and canard; this example shows the importance of an approximate grid-to-shock alignment, which is possible with the flexible choice of local grids (and sometimes impossible in codes with fixed grids).

2) A fighter configuration including body, wing, fairing, canard, and external fuel tank; the analysis shows an unexpectedly strong interference between body, wing, and fuel tank.

3) A civil aircraft configuration with different wing root fairings; this case shows the role of the fairing on the wing pressure distribution.

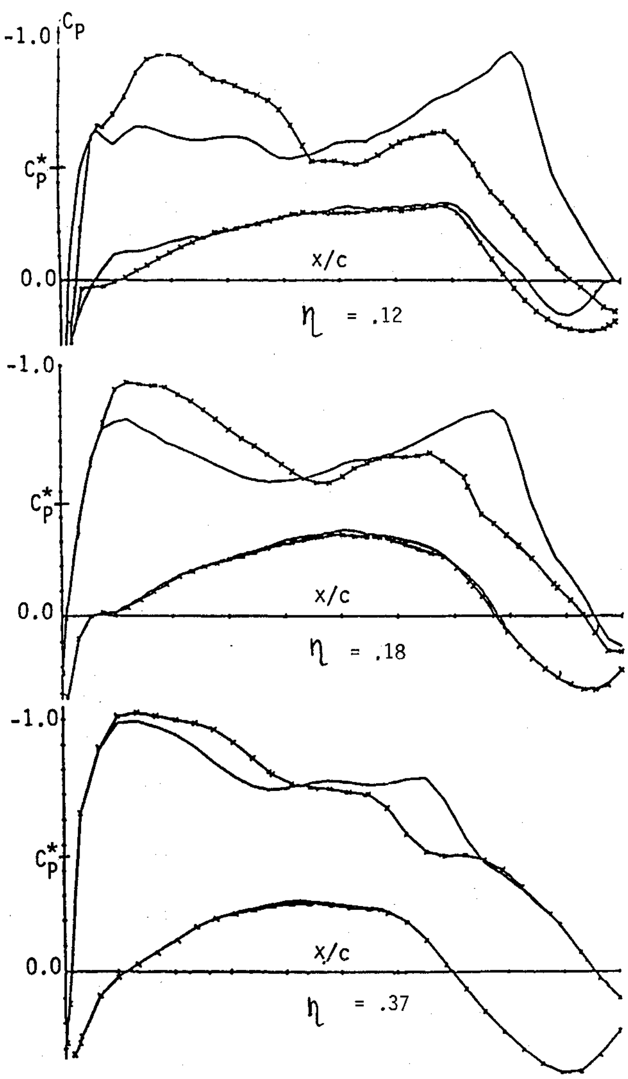

Fig. 16 Civil aircraft, calculated C_p distribution. $M=0.8$, $\alpha=1.5$ deg: ——— orginal fairing, –×–×– modified.

Acknowledgments

Prof. A. Brandt's consultations in the Multigrid have been very useful in the algorithm design. I. Darel supported the project in the IAI and helped to clarify many engineering problems in the applications. B. Bilu and Dr. S. Yaniv took part in the computational work. We are deeply grateful to them all.

References

[1]Bristeau, M. O., Pironneau, O., Glowinski, R., Periaux, J., Perrier, P., and Poirier, G., "On Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least Squares and Finite Elements

Methods (II): Applications to Transonic Flow Simulations," *Proceedings of the Third International Conference on Finite Elements in Nonlinear Mechanics*, edited by J. St. Doltsinis, North Holland, Amsterdam, 1985, pp. 363-394.

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

[3] Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft,' AIAA Paper 86-1103, Jan. 1986.

[4] Jameson, A. and Baker, T. J., "Improvements to the Aircraft Euler Method," AIAA Paper 87-0452, Jan. 1987.

[5] Reyhner, T. A., "Three-Dimensional Transonic Potential Flow about Complex Three-Dimensional Configurations," NASA Contractor Rept. 3814, July 1984.

[6] Wedan, B. and South, J. C., "A Method for Solving the Transonic Full-Potential Equation for General Configurations," AIAA Paper 83-1889, July 1983.

[7] Gaffney, R. L., Hassan, H . A., and Salas, M. D., "Euler Calculations for Wings Using Cartesian Grids," AIAA Paper 87-0356, Jan. 1987.

[8] Luntz, A. L. and Epstein, B., "A Multigrid Full Potential Transonic Code for Arbitrary Configurations," GMD-Studien No. 110, *Proceedings of the 2nd European Conference on Multigrid Methods,* GMD Sankt Augustin, FRG, May 1986, pp. 101-110.

[9] Luntz, A. L. and Epstein, B., "Full Potential Transonic Code for Arbitrary Configurations," *Collection of Papers of the 28th Israel Annual Conference on Aviation and Astronautics,* Feb. 1986, pp. 87-92.

[10] Brandt, A., "Multi-Level Adaptive Computations in Fluid Dynamics," *AIAA Journal*, Vol. 18, Oct. 1980, pp. 100-108.

[11] Brandt, A., *Multigrid Techniques: 1984 Guide,* The Weizman Institute of Science, Rehovot, Israel, 1984, p. 93 .