ICAS-84-1.3.2

COMPUTATION OF SUPERSONIC FLOW ABOUT COMPLEX CONFIGURATIONS

Bert G. Arlinger*

SAAB-SCANIA, Linköping, Sweden

Abstract

A computational method is presented for the solution of the three-dimensional Euler equations for supersonic flow about realistic aircraft configurations. Utilizing a space-marching technique with an explicit scheme the solution is advanced between planes normal to the free stream direction. A key feature of the method is the very versatile grid generated in each cross section plane. Fuselages of widely different shape, with or without inlets, a fin, etc., can be treated. The number of wings can be from 0 to 2 without any restrictions on plan form.

Computational results for a realistic aircraft configuration with canard and delta wing illustrates the capability of the code.

1. Introduction

A rather large variety of methods exists today for the computation of steady three-dimensional supersonic flows. Codes based on linear theory[1] or on variants improved by higher-order corrections[2] are often used in preliminary aerodynamic analysis. For more detailed studies, however, nonlinear methods have to be applied. Thus, codes have been developed that solve the full nonlinear potential equation[3] also for complex aircraft configurations[4]. Although methods of this type can be improved through the use of entropy correction techniques to yield quite accurate results even for cases with strong shock waves[5], the basic limitation of potential methods to zero vorticity flows makes a continued development of methods based on the Euler equations important.

Among the existing codes for the steady Euler equations those applying shock fitting technique[5,6] yield solutions with highly resolved shock waves for flows around simple configurations. For complex geometries, however, a shock capturing formulation is probably the best basis for a robust and versatile three-dimensional Euler code. Most methods of this type[7-9] have been applied to rather generalized configurations, largely due to the problem of grid generation for complex shapes.

The present work reflects an effort to develop a code based on the steady Euler equations applicable to detailed real aircraft configurations. An explicit space-marching algorithm was chosen because that was considered the simplest way to cope with the different constellations of boundary conditions appearing during the marching process over a complex geometry. Because the marching step size in an explicit scheme is linked to the grid cell dimensions in the cross section planes through a stability condition, the characteristic cell dimension in any cross section plane was chosen to exceed a prescribed minimum value. The grid was also made to adapt to wing and body contours in all the marching planes to facilitate the treatment of the boundary conditions.

Based on these ideas codes have been developed for wings[10] and for bodies[11] that demonstrate the feasibility of the approach. This work has now been generalized to wing-body-fin combinations of arbitrary shapes. Because the flow is required to be supersonic in the marching direction everywhere in the flow field, too blunt nose or leading edge shapes relative the grid mesh size are excluded.

The code, denoted SUMA, works for real aircraft fuselages with canopy, inlets, fin, etc. The number of wings can be from 0 to 2, with no restrictions on plan form or location. For instance, overlapping wings can be dealt with.

The steady Euler equations are solved using a shock-capturing finite volume formulation. The solution is advanced between planes normal to the free stream direction applying an explicit marching algorithm.

The key feature of the method is the very versatile grid generated in each cross section plane. The transfinite interpolation technique applied in the grid generation is described and illustrated by a number of representative grids around a realistic aircraft configuration of canard-wing type.

Computational results are presented for a wind tunnel tested simple configuration as well as for the detailed aircraft configuration.

2. Mathematical Formulation

The steady Euler equations in conservative form written in Cartesian coordinates are

$$\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \qquad (1)$$

where F, G and H are the fluxes in the three coordinate directions,

$$F = \begin{pmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ \rho uw \end{pmatrix} \quad G = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2+p \\ \rho vw \end{pmatrix} \quad H = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2+p \end{pmatrix} \qquad (2)$$

Here u, v and w represent the three velocity components and p and ρ denote the pressure and density.

In steady flow the energy conservation equation is integrated to

$$\frac{\gamma}{\gamma-1}\frac{p}{\rho} + \frac{1}{2}(u^2+v^2+w^2) = H \qquad (3)$$

where γ is the ratio of specific heats and H represents the total enthalpy which is constant in the whole field for uniform upstream flow.

* Head of Computational Aerodynamics
Member AIAA

Provided u is greater than the local speed of sound $c = (\gamma p/\rho)^{\frac{1}{2}}$, the five equations (1)-(3) constitute a hyperbolic system in the x direction, and the solution can thus be obtained by a marching procedure in that direction.

3. Finite Volume Scheme

The integral form of the steady Euler equations can be written as

$$\iint_{\partial\Omega} [F,G,H] \cdot d\bar{S} = 0 \tag{4}$$

for a closed volume with boundary $\partial\Omega$ and $d\bar{S}$ as surface area vector differential. The three-dimensional grid cells used in the finite volume formulation of the Euler equations are formed by the cross section grids in two consecutive marching planes. The grid cells thus have parallell end faces normal to the x direction, and denoting the area of these by S^n and S^{n+1} with superscript n indicating the x level, a semi-discretized version of Eq.(4) is

$$S^{n+1}F^{n+1} = S^nF^n - \iint_{\partial\Omega'} [F,G,H] \cdot d\bar{S} \tag{5}$$

where $\partial\Omega'$ denotes the four side faces of the cell. An explicit scheme based on Eq.(5) has to be of multi-step type. A two-step scheme of MacCormack type is presently used in SUMA, but others, like symmetric multi-step schemes, might as well be an alternative. The MacCormack scheme can be written

$$\begin{aligned} F^{(0)} &= F^n \\ S^{n+1}F^{(1)} &= S^nF^{(0)} - P_1T^{(0)} \\ S^{n+1}F^{(2)} &= S^nF^{(0)} - P_2T^{(1)} \\ F^{n+1} &= 0.5\,(F^{(1)} + F^{(2)}) \end{aligned} \tag{6}$$

where T is the flux tensor [F,G,H] and the side face flux terms P_1T and P_2T are defined as follows for a cell with indices i and j in the cross section plane directions:

$$\begin{aligned} P_1T &= Q_+ + R_+ \\ P_2T &= Q_- + R_- \end{aligned} \tag{7}$$

with

$$\begin{aligned} Q_+ &= T_{i+1,j}\cdot\bar{S}_{+i} + T_{i,j}\cdot\bar{S}_{-i} \\ Q_- &= T_{i,j}\cdot\bar{S}_{+i} + T_{i-1,j}\cdot\bar{S}_{-i} \\ R_+ &= T_{i,j+1}\cdot\bar{S}_{+j} + T_{i,j}\cdot\bar{S}_{-j} \\ R_- &= T_{i,j}\cdot\bar{S}_{+j} + T_{i,j-1}\cdot\bar{S}_{-j} \end{aligned} \tag{8}$$

Here $\bar{S}_{+i}$ denotes the side face area vector pointing in the direction of increasing i and $\bar{S}_{-i}$ is the opposite side face area vector pointing towards decreasing i, with analogous notation for $\bar{S}_{+j}$ and $\bar{S}_{-j}$, see Fig. 1. Each surface area vector is computed as 0.5 times the vector cross product of the two diagonal vectors. The flux tensor T is evaluted at the node points located at the mid points of the cell end faces as indicated in Fig. 1. In Eq. (7) the + and - subscripts can be shifted between the Q terms as well as between the R terms yielding 4 variants of the scheme.

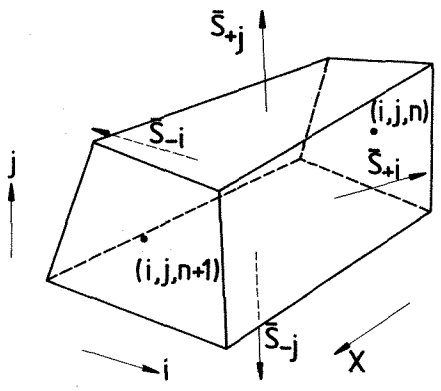


Fig. 1 Computational cell

An artificial viscosity term is included as an option in the scheme by modifying Eq.(6d) to

$$F^{n+1} = 0.5\,(F^{(1)} + F^{(2)}) + DF^n \tag{9}$$

The viscosity term DF^n is of the type used in several time-dependent Euler schemes[12] as a means to damp oscillations in the solution close to shock waves. It is used in the same function in the present code, but is not required for stability or other reasons. For a cell with cross section plane indices i and j the artificial viscosity term is defined by the expressions

$$DF_{ij} = \mu\Delta x \frac{\Delta y+\Delta z}{2S_{ij}^{n+1}} \left(d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j} + d_{i,j+\frac{1}{2}} - d_{i,j-\frac{1}{2}}\right)$$

$$d_{i+\frac{1}{2},j} = \varepsilon_{i+\frac{1}{2},j}\,(F_{i+1,j} - F_{ij}) \tag{10}$$

$$\varepsilon_{i+\frac{1}{2},j} = \max(\nu_{ij}, \nu_{i+1,j})$$

$$\nu_{ij} = \frac{|p_{i+1,j} - 2p_{ij} + p_{i-1,j}|}{p_{i+1,j} + 2p_{ij} + p_{i-1,j}}$$

with corresponding expressions for $d_{i,j+\frac{1}{2}}$ where i is the constant index. The coefficient μ is chosen between 0 and 1 and Δy and Δz are the prescribed typical grid dimensions in the cross section plane.

In smooth regions the viscosity term is of third order, while in regions with rapidly varying pressure ν_{ij} becomes 0(1) and thus the viscosity term of first order if μ is chosen 0(1).

4. Stability and boundary conditions

The limitation on the marching step size is obtained from the Courant-Friedrich-Lewy condition applied locally in two dimensions in each of two planes for each cell. Those planes are defined by the velocity vector and the two mean side edge normals $\bar{s}$ and $\bar{t}$ measured in the cross section planes. Denoting the cell mean width in these two directions by Δs and Δt and the mean slopes in the x direction of the corresponding cell side faces by s' and t' as indicated in Fig. 2, the two conditions on the step size Δx can be written

$$\Delta x \leq \frac{\Delta s\ (q_{ns}^2 - c^2)}{s'(q_{ns}^2 - c^2) - \frac{v_s}{u} q_{ns}^2 + c^2 \frac{q_{ns}}{u} \beta} \qquad (11)$$

$$\Delta x \leq \frac{\Delta t\ (q_{nt}^2 - c^2)}{t'(q_{nt}^2 - c^2) - \frac{v_t}{u} q_{nt}^2 + c^2 \frac{q_{nt}}{u} \beta}$$

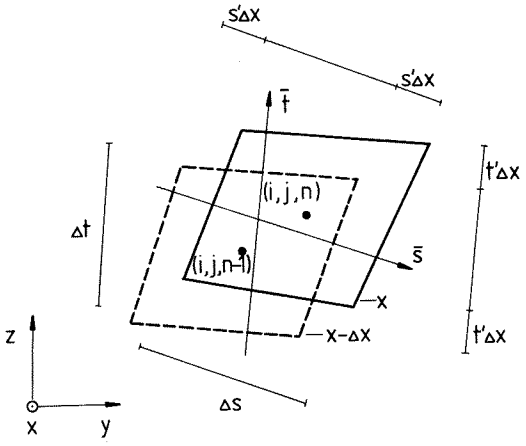


Fig 2. The end faces of a computational cell

Here v_s and v_t are the velocity components in the cross section plane in the $\bar{s}$ and $\bar{t}$ directions. The velocity vector is denoted by q and the two velocities q_{ns} and q_{nt} are defined by

$$q_{ns}^2 = q^2 - v_s^2$$

$$q_{nt}^2 = q^2 - v_t^2$$

and c and β represent the local sound speed and $(q^2/c^2-1)^{\frac{1}{2}}$ respectively.

In the actual implementation of Eq. (11) conservative values of Δx are obtained by adding the absolute values of all terms in the denominators. The smallest Δx thus obtained from all cells in the cross section plane is then reduced by a factor around 0.7 to 0.9 to obtain the next marching step.

The boundary conditions at the body and wing surfaces are zero normal flow. This means, that to evaluate the flux terms of Eq. (8) at a cell face coinciding with such a boundary only the surface pressure has to be computed. Because the node point closest to the surface is located half a cell width outside, this pressure has to be computed from some relation connecting it with the flow variables at the node points.

In SUMA a characteristic relation is used along a bicharacteristic ξ^- pointing downstreams towards the surface. The general three-dimensional characteristic relation in a plane containing the velocity vector may be written

$$\frac{\partial p}{\partial \xi^-} \pm \frac{\rho u'^2}{\beta} \frac{\partial \rho}{\partial \xi^-} \left(\frac{w'}{u'}\right) + \frac{\rho c}{\beta} \frac{\partial v'}{\partial \eta} = 0 \qquad (12)$$

where ξ^+ and ξ^- are the two bicharacteristics in the plane, η denotes a direction normal to the plane u' and w' are any two orthogonal velocity components in the plane with the normal velocity component v' equal to 0 and $\beta = ((u'^2+w'^2)/c^2-1)^{\frac{1}{2}}$. The two bicharacteristic directions have the slopes

$$\frac{u'w' \pm \beta c^2}{u'^2 - c^2}$$

relative to the u' direction.

Along wing surfaces Eq. (12) is applied locally in 2 dimensions disregarding the normal derivative, and along the body surface another variant of Eq. (12) is used that is written in cylindrical coordinates. The latter version is implemented so that the body surface local transversal curvature also is taken into consideration. Further details about the application of body and wing boundary conditions can be found in Ref. 10 and 11.

The solution is computed only on one side of the vertical symmetry plane. At the outer boundaries the condition used presently is simply to assume zero normal gradients in the flow variables. The grid outer boundaries are chosen under consideration of possible interference effects from disturbances reflected at these boundaries. To minimize the computational work the outer boundaries actually used in the marching process are initially quite close to the apex of the body, and then automatically shifted outwards in order to contain the disturbed region for all x values until the grid outer edges are reached.

5. Grid generation

Because the marching step sizes are unknown in advance the whole three-dimensional grid can not be generated before the solution process starts. Instead, at each marching step a new two-dimensional grid is computed in the next cross section plane to be solved.

The grid in each cross section plane adapts to the wing and body contours and tends to a Cartesian type grid far away from the body. It is generated by algebraic technique, essentially based on transfinite interpolation. That means that a grid point distribution is first computed along the body section - symmetry line contour and the outer boundaries to control the mesh. Wing surfaces also effect the mesh, and from the leading and trailing edges a grid surface is made to leave in the upstream and downstream directions. The grid point distribution along the cross section cut through

the wing surfaces or their continuations is thus also computed before the interior grid is generated and used as boundary conditions for that.

The smooth grid point distribution computed along the body contour is obtained by applying a prescribed wanted mesh size along an artificial arc length around the body. This artificial arc length corresponds to a modified version of the body section-symmetry line contour, having a normal the slope of which relative the horizontal direction nowhere exceeds a prescribed maximum value.

The modification is made so that the resulting distribution of normals have a monotonicly varying slope over the upper and lower body parts respectively. This is done because the distribution of normals is used as a second boundary condition along the body section - symmetry line contour to control the outgoing grid line slopes. With monotonicly varying slopes, problems, for instance with intersection of adjacent grid lines outside concave boundaries, are avoided.

The body sections have to be computed for arbitrary fuselage shapes and any angle of attack, so routines for geometry interpolation in circumferential and streamwise directions are used first at each new marching plane to accurately define the body section contour.

As for the wings, the program is written for 0 to 2 wings that can have any type of plan form, can lie in the same plane or not, and also can overlap. The horizontal type coordinate surfaces that contain the wings' leading and trailing edges and serve as inner boundary conditions in the transfinite interpolation procedure are made to gradually adapt to only body-controlled surfaces away from the wing regions. The grid point distribution given as boundary condition along these surfaces is essentially uniform except for a possible mesh stretching. The wing tips are assumed to have zero thickness (otherwise they are automatically sharpened) and a grid line is made to coincide with each of them.

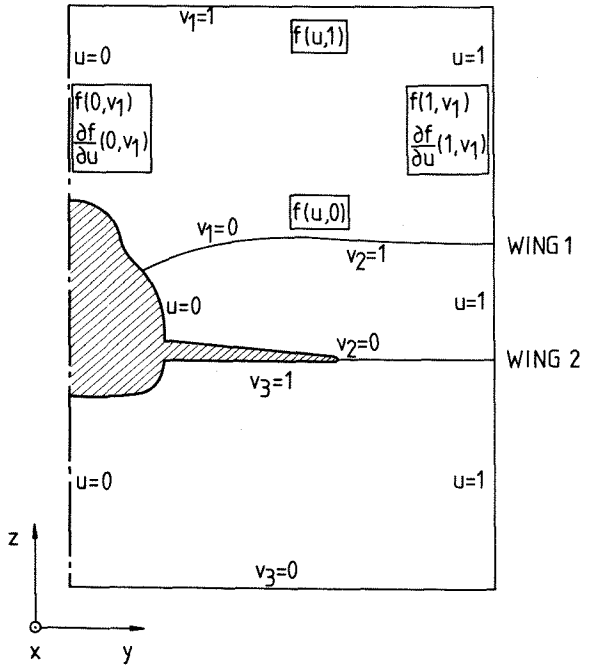


Fig. 3 Boundary conditions used in transfinite interpolation with f corresponding to y and z.

The generation of the inner grid thus starts out from the set of boundary conditions illustrated in Fig. 3 for a case with two wings in different planes. Wing 1 is represented by its coordinate surface emanating from its leading or trailing edge. The two wing boundaries divide the total cross section grid into 3 regions, each region being described by the two interpolation variables u and v ranging between 0 and 1 (this definition of u and v only used in connection with grid generation). The transfinite interpolation is made in each of the three grid regions for the two cross section coordinates y and z based on the type of boundary conditions shown in Fig. 3. The following relations are applied, written for a function f instead of y and z:

$$P_u f = \alpha_{10}(u)\, f(0,v) + \alpha_{11}(u)\, \frac{\partial f}{\partial u}(0,v) + \alpha_{20}(u)\, f(1,v) + \alpha_{21}(u)\, \frac{\partial f}{\partial u}(1,v)$$

$$P_v f = \beta_{10}(v)\, f(u,0) + \beta_{20}(v)\, f(u,1) \qquad (13)$$

$$P_{uv} f = P_u f + P_v (I - P_u) f$$

The notation used here is close to that of Ref. 13 with P_u and P_v denoting the interpolation operators in the u and v directions and P_{uv} as the final two-dimensional operator. I stands for the identity operator, and the letter P indicates that these operators define projections; thus $P_u{}^2 f = P_u f$, etc. The interpolation functions $\alpha_{10}(u)$, $\alpha_{11}(u)$, etc have to satisfy the boundary conditions

$$\begin{array}{llll} \alpha_{10}(0)=1 & \alpha'_{10}(0)=0 & \alpha_{10}(1)=0 & \alpha'_{10}(1)=0 \\ \alpha_{11}(0)=0 & \alpha'_{11}(0)=1 & \alpha_{11}(1)=0 & \alpha'_{11}(1)=0 \\ \alpha_{20}(0)=0 & \alpha'_{20}(0)=0 & \alpha_{20}(1)=1 & \alpha'_{20}(1)=0 \\ \alpha_{21}(0)=0 & \alpha'_{21}(0)=0 & \alpha_{21}(1)=0 & \alpha'_{21}(1)=1 \end{array} \qquad (14)$$

$$\begin{array}{ll} \beta_{10}(0)=1 & \beta_{10}(1)=0 \\ \beta_{20}(0)=0 & \beta_{20}(1)=1 \end{array}$$

where α'_{10} stands for $\frac{\partial \alpha_{10}}{\partial u}$, etc.

The internal behaviour of the α and β functions is in principle free of choice, and the expressions used in the present program are chosen so that the body influence on the grid as determined by $\alpha_{10}(u)$, is halved at a prescribed u value, u_c. This is achieved by the function F(u) used for $\alpha_{10}(u)$

$$F(u) = 0.5 \left\{ 1 + \cos [\pi(au + (1-a)u^b)] \right\} \qquad (15)$$

where, for a given a, b is determined so that $F(u_c)=0.5$. Eq. (15) is used for $u_c \geq 0.5$, and for $u_c < 0.5$ $F(u)$ is replaced by $1-F(1-u)$.

The following expressions were used for the interpolating functions:

$$\begin{aligned}
\alpha_{10}(u) &= F(u) \\
\alpha_{11}(u) &= (1-u)^2 \, (u-0.5\,u^2) \, F(u) \\
\alpha_{20}(u) &= 1 - F(u) \\
\alpha_{21}(u) &= u-1 + F(u) - \alpha_{11}(u) \\
\beta_{10}(v) &= 0.5[1+\cos(\pi v)] \\
\beta_{20}(v) &= 1 - \beta_{10}(v)
\end{aligned} \qquad (16)$$

The functions α_{20} and α_{21} where obtained from the requirement that a linearly varying function in u

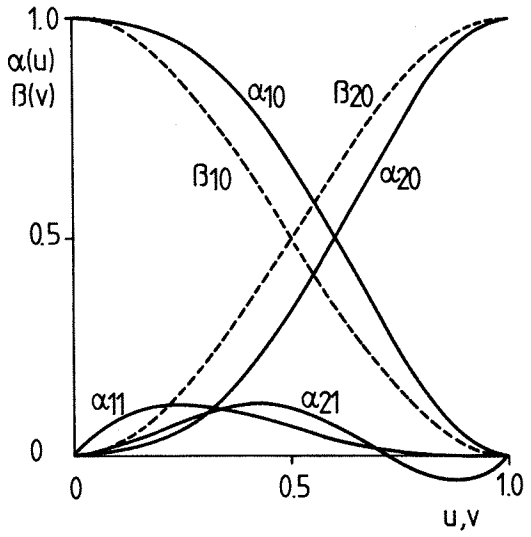


Fig. 4 Example on interpolation functions $\alpha(u)$, $\beta(v)$

should be interpolated identically. In Fig. 4 the α and β functions are illustrated for a=0.2 and u_c=0.6. The precise choice of interpolation functions is not critical for the resulting grid. Various modifications of the expressions in (16) have thus been investigated and, for instance, $\beta_{10}(v)$ is slightly different in the three regions in the grid examples shown below.

The actual implementation of the transfinite interpolation differs from Eqs. (13) due to the introduction of smoothing operators to prevent corners and kinks in the boundary values to show up in the flow field. Thus Eq. (13a), which interpolates the coordinate points in spanwise direction, is written for a vertical grid line with index i (i = 1 corresponds to the body section-symmetry line contour)

$$(P_u f)(u_i,v) = \alpha_{10}(u_i) S_v^{i-1} f(0,v) + \alpha_{11}(u_i) S_v^{i-1} \frac{\partial f}{\partial u}(0,v) + \alpha_{20}(u_i)\, f(1,v) + \alpha_{21}(u_i) \frac{\partial f}{\partial u}(1,v) \qquad (17)$$

Eq. (13b) is modified to

$$(P_v f)(u,v_j) = \beta_{10}(v_j)\, S_u^{j-1} f(u,0) + \beta_{20}(v_j)\, S_u^{M-j} f(u,1) \qquad (18)$$

for a grid line in spanwise direction with index j, ranging from 1 to M (a specific M for each grid region). Here S_v and S_u are smoothing operators in the v and u directions respectively, applied on the boundary values one more time per each step away from the boundary as indicated by the powers i-1, j-1 and M-j. The operator S_v computes a smoothed value f_j at point j from the simple 5-point scheme along a vertical line

$$S_v f_j = (1-\nu) f_j + \frac{\nu}{4} (f_{j-2} + f_{j-1} + f_{j+1} + f_{j+2})$$

with ν usually chosen between 0 and 0.5. The same scheme was used for S_u.

The grid generating procedure works fast and efficiently for a wide variety of body and wing shapes. The body can also have air inlets and a fin. In case of air inlets the stream tube that goes into the inlet is just made to disappear into the body. Presently the inlet is assumed to be of pitot type. Two cross section grids are generated at the inlet station, adapted to the geometry immediately before and after the inlet station respectively. The flow variables are then interpolated from the pre- to the post-intake grid and the marching procedure is continued.

To illustrate the present grid generation technique a number of cross section grids where generated for a realistic military aircraft configuration shown in Fig. 5. The contours drawn in the figure consti-

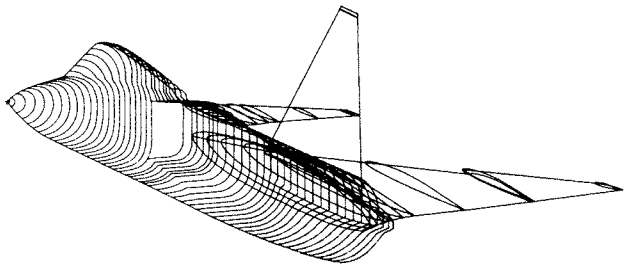
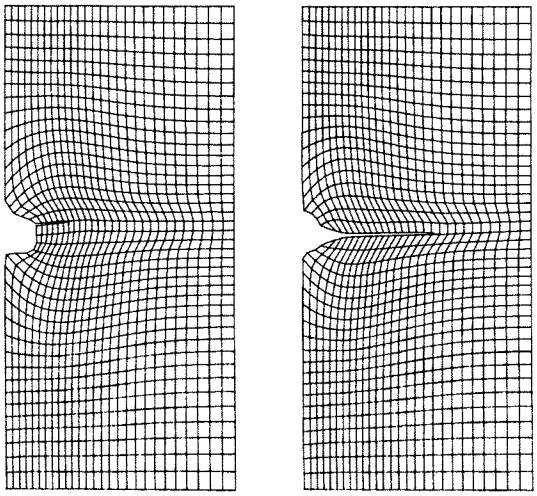


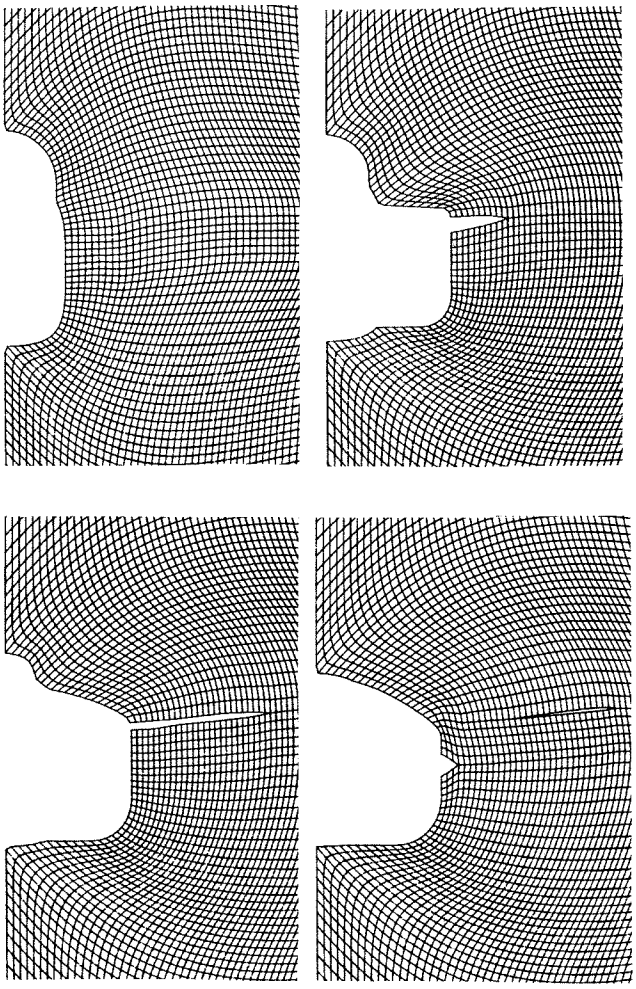
Fig. 5 Geometry indata for realistic aircraft configuration.

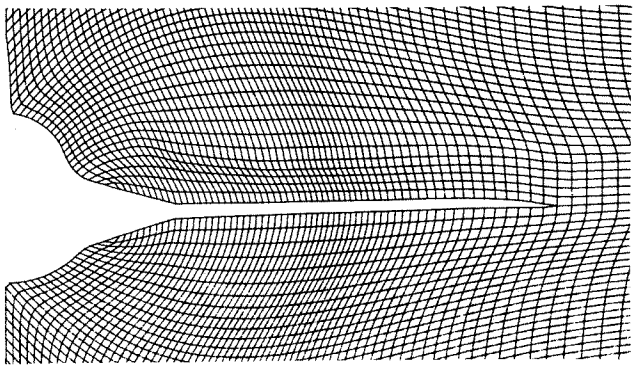
tute the geometry input data used in the computations. The resulting grids in a number of cross sections are shown in Fig. 6. The coarse grids in Fig. 6a show the Cartesian character far away from the body and also that some stretching normally is

applied in the horizontal and vertical directions. The 5 grids that follow represent a mesh used in an actual flow calculation with results shown below. As seen, the grids adapt to quite different shapes of wing-body contour. SUMA is presently written to handle up to 22000 grid cells per cross section plane which enables very high resolution of geometry details.



a) Coarse grids





b) Parts of grids used in flow computations

Fig. 6 Grids in various cross section planes through realistic aircraft configuration.

6. Computational results

A comparison with experimentally determined pressures on a wing-body model was obtained by computing the flow around a double-cone-cylinder body with a straight rectangular wing[14]. The wing section was a 10 % thick double wedge. With a wing chord of 2.1 and a maximum body radius of 1.5 the typical grid size was chosen 0.1 and the flow field was computed for $M_\infty = 2.0$ and $\alpha = 3.3^\circ$. The results, shown in Fig. 7, agree very well with the experimental data. The spanwise pressure variation is seen to be quite large, due to disturbances emanating from the wing root and tip and from the kinks in the forebody contour.

In Fig. 8 the computed pressure contours are shown in two cross section planes for the same case. The expansion regions are clearly seen above and below the wing in Fig. 8a taken at an x station corresponding to 70.2 % relative chord. The stronger shock wave below the wing is also noted in this figure as well as in the next, which shows the cross section plane at 66.4 % chord behind the trailing edge. Here also the upper surface trailing edge shock is seen.

A series of computational results is shown next for the aircraft configuration of Fig. 5. The grid used is the same as that examplified in Fig. 6b with 10400 grid cells per cross section plane. The configuration was run at $M_\infty = 1.8$ and $\alpha = 0$, and in Fig. 9 pressure contour plots are presented for a series of cross section planes.

The bow shock wave is seen at stations 1 and 2, and in the latter also the canopy shock wave appears. Station 3 is located just aft of the inlet, the sloping sides of which generate another shock wave. Among the flow field features at the later stations the fairly strong main wing shock wave is seen, as well as the fin leading edge disturbance at station 6.

The number of grid cells at each cross section plane engaged in the marching procedure varied from 72 at the nose apex to 10400 , yielding a total of $5.08 \cdot 10^6$ cells. The computing time was 162 CPU seconds on a CRAY-1, thus illustrating the effectiveness of the marching technique.

7. Concluding remarks

A computer code (SUMA) has been presented for the computation of supersonic flow about complex aircraft configurations. The three-dimensional Euler equations are solved in a space-marching procedure using an explicit finite volume scheme on a grid generated on consecutive cross section planes. The very versatile grid is a key feature of the method, enabling its application to complex aircraft configurations with, for example, inlets canopy, fin, two wings of arbitrary planform and location, etc.

A comparison of computed and experimental results shows very good agreement, and an application on a complex canard configuration illustrates the versatility of the method.

Among possible further developments, investigations on symmetric multi-step operators might be mentioned, as well as experiments on iterative processes to deal with local subsonic regions.

References

1. Carmichael,R.L. and Erickson,L.L., "Pan Air - A Higher Order Panel Method for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations", AIAA Paper 81-1255, 1981.

2. "Formulation of Aerodynamic Prediction Techniques for Hypersonic Configuration Design", NASA CR 158994, February 1979.

3. Grossman,B. and Siclari,M.J., "The Nonlinear Supersonic Potential Flow Over Delta Wings", AIAA Paper 80-0269, Jan. 1980.

4. Szema,K.Y. and Shankar,V., "Nonlinear Computation of Wing-Body-Vertical Tale-Wake Flows at Low Supersonic Speed", AIAA Paper 84-0427, Reno, 1984.

5. Siclari,M.J. and Visich,M., "Shock Fitting in Conical Supersonic Full Potential Flows with Entropy Effects", AIAA Paper 84-0261, Reno, 1984.

6. Marconi,F. and Koch,F., "An Improved Supersonic Three-Dimensional, External, Inviscid Flow Field Code", NASA CR 3108, 1979.

7. Kutler,P., Lomax,H. and Warming,R.F., "Computation of Space Shuttle Flow Fields Using Noncentered Finite-Difference Schemes", AIAA J., Vol 11, No 2, pp. 196-204, Feb. 1973.

8. Kutler,P., "Computation of Three-Dimensional Inviscid Supersonic Flows," Progress in Numerical Fluid Dynamics, Lecture Notes in Physics, Vol 41, Springer-Verlag, Berlin, 1975, pp. 287-374.

9. Rizzi,A.W., Klavins,A. and MacCormack,R.W., "A Generalized Hyperbolic Marching Technique for Three-Dimensional Supersonic Flow with Shocks", Proc. Fourth Int. Conf. on Num. Methods in Fluid Dynamics, Lecture Notes in Physics 35, Springer-Verlag, 1975, pp. 341-346.

10. Arlinger,B.G., "Computation of Supersonic Flow Around Three-Dimensional Wings", ICAS Paper 82-6.1.3, Seattle, 1982.

11. Arlinger,B.G., "Computation of Supersonic Flow Around Bodies", AIAA Paper 84-0259, Reno,1984.

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

13. Eriksson,L-E., "Transfinite Mesh Generation and Computer-Aided Analysis of Mesh Effects", Doctoral Dissertation, Department of Computer Sciences, Uppsala University, 1984.

14. Berler,I., Nichols,S., "Interference Between Wing and Body at Supersonic Speeds", Report No. CAL/CF-1569, Part VI, Data Report, Cornell Aeronautical Laboratory, Buffalo, N.Y.,1951.

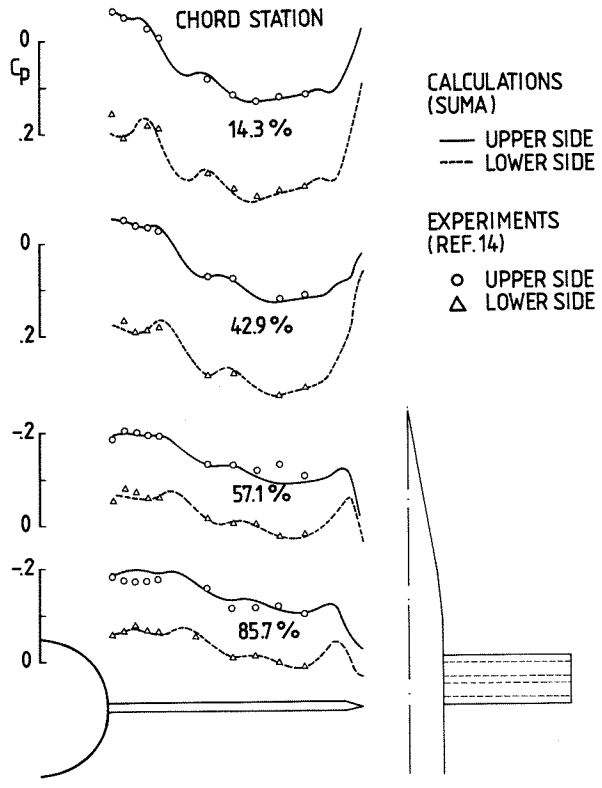
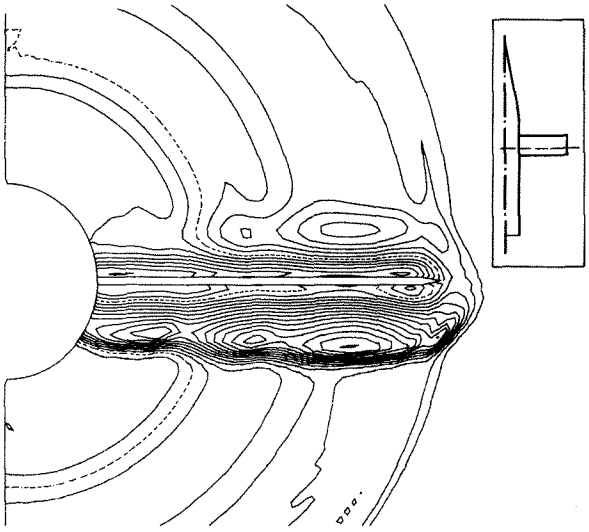
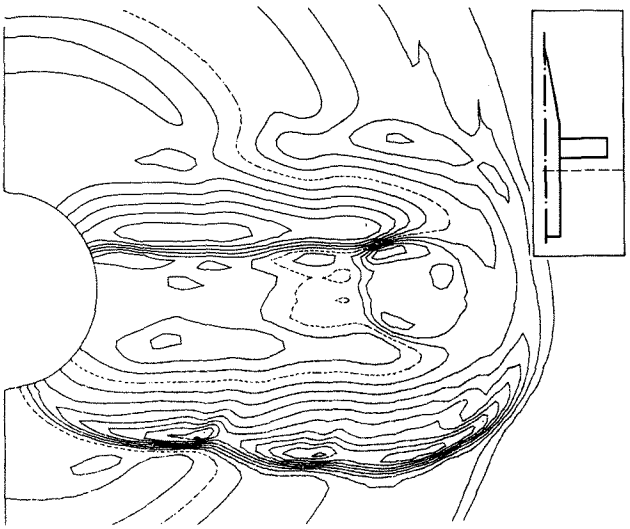


Fig. 7 Computed and measured wing pressures on wing-body model at $M_\infty = 2.0$, $\alpha = 3.3^\circ$.

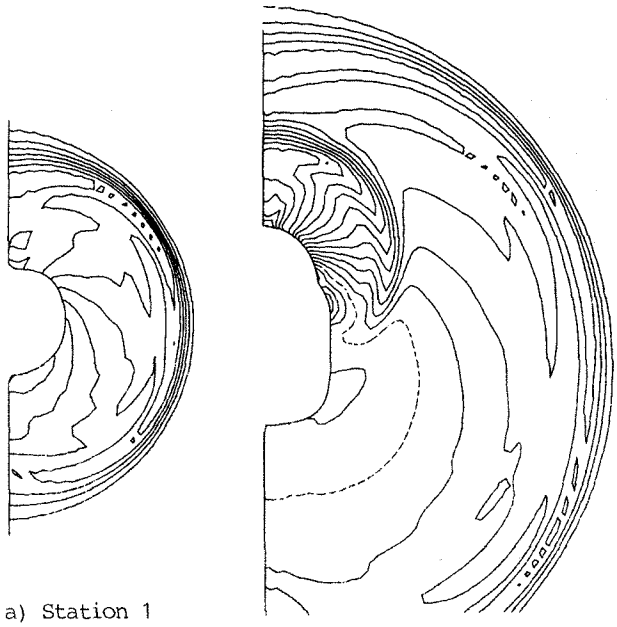


a) x = 12.03



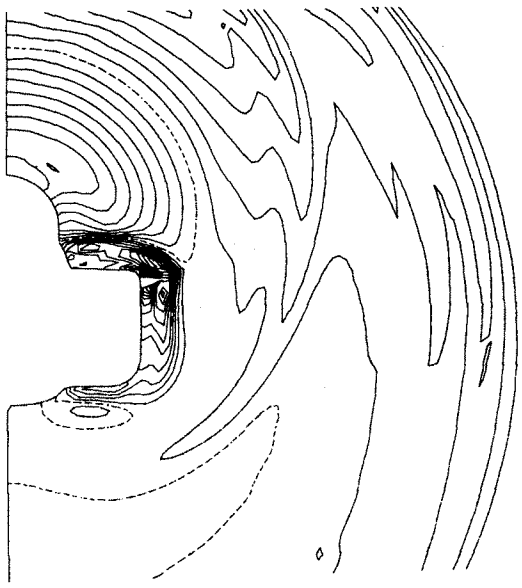
b) x = 14.05

Fig. 8 Pressure contours (Δc_p=0.025) in cross section planes for wing-body model at M_∞ = 2.0, α = 3.3°. Dashed line: C_p=0 (x_{LE} = 10.556 , x_{TE} = 12.656)



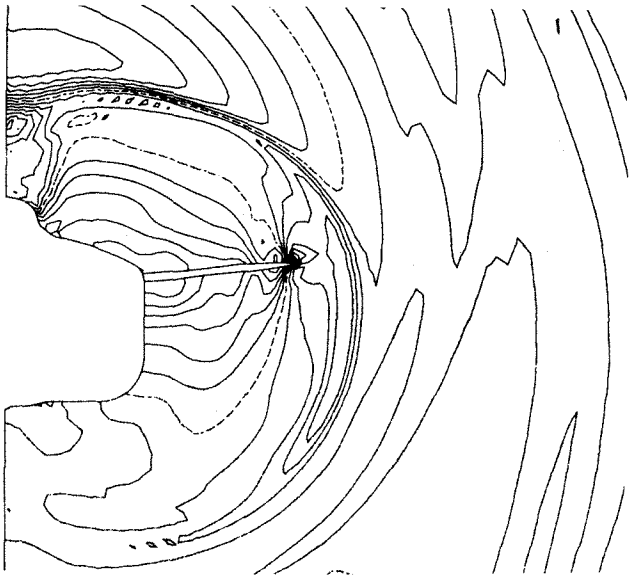
a) Station 1

b) Station 2

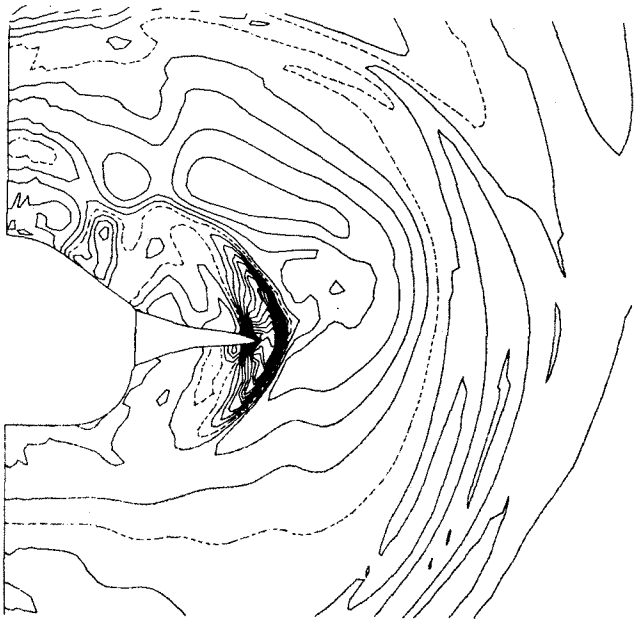


c) Station 3

Fig 9. Pressure contours (Δc_p=0.025) in cross section planes through complex aircraft configuration at M_∞ = 1.8, α = 0. Dashed line: c_p=0.

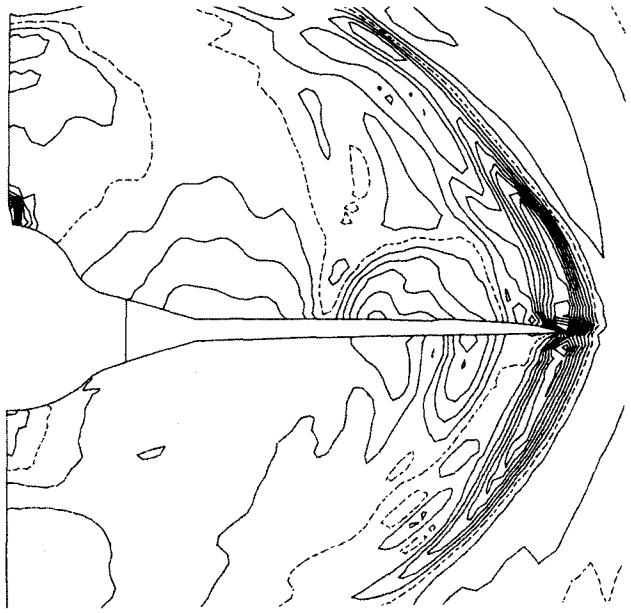


d) Station 4

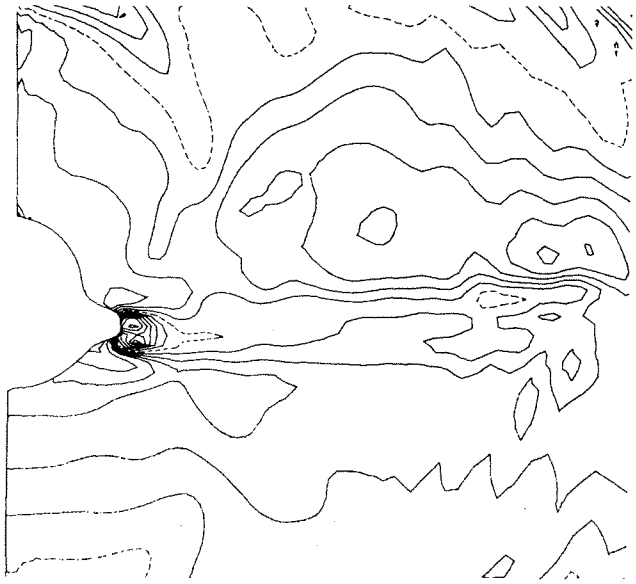


e) Station 5

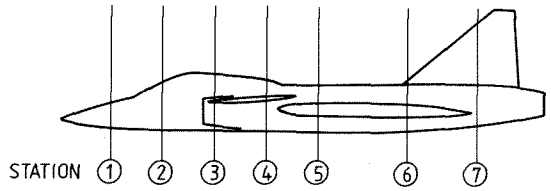
5 × 10⁶ mesh points
160 sec on CRAY I



f) Station 6



g) Station 7



h) Cross section stations

Fig. 9 Continued