NASA CONTRACTOR REPORT 166580

An Analysis of Three-Dimensional Transonic Compressors

(NASA-CR-166580) ANALYSIS OF THREE-DIMENSIONAL TRANSONIC COMPRESSORS (New York Univ., New York.) 110 p HC A06/MF A01 CSCL 01A

N84-32356

Unclas

G3/02 20514

Antoine Bourgeade

CONTRACT NGR-33-016-201

February 1983

NASA

NASA CONTRACTOR REPORT 166580

An Analysis of Three-Dimensional Transonic Compressors

Antoine Bourgeade
Courant Institute of Mathematical Sciences
New York University
New York, N.Y.

Prepared for
Ames Research Center
under grant NGR-33-016-201

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

Contents

AN ANALYSIS OF THREE-DIMENSIONAL TRANSONIC COMPRESSORS

Antoine Bourgeade

February 1983

ABSTRACT

This presentation sets forth a method for computing the three-dimensional transonic flow around the blades of a compressor or of a propeller. The method is based on the use of the velocity potential, on the hypothesis that the flow is inviscid, irrotational and isentropic.

The equation of the potential is solved in a transformed space such that the surface of the blade is mapped into a plane where the periodicity is implicit. This equation is in a nonconservative form and is solved with the help of a finite difference method using artificial viscosity and artificial time.

A computer code is provided and some sample results are given in order to demonstrate the influence of three-dimensional effects and the blade's rotation.

I. INTRODUCTION

Many scientists and engineers continue to study the ways and means of making better use of the energy resources at our disposal, even though the energy crisis is perhaps no longer considered so severe. Turbines and compressors, which both create and consume energy, have in recent years been the subject of many theoretical and experimental studies aimed at improving their design and efficiency. Although some major improvements have been introduced lately, the study of the transonic flow across a single stage of a turbine, or of a compressor, still remains highly complex. For this reason most of the theoretical work done so far has focused on the two-dimensional cascade problem [13-17] or on the mean flow problem [22].

Let us first discuss the physical background. A basic compressor consists of a succession of rotors and stators. These rotors and stators are situated on the hub and surrounded by the cowling; they are composed of a certain number of blades distributed around the hub, the shape of the blades depending on their use. If the hub is cut along a generator line and transformed into a plane, a so-called "cascade" of these blades is obtained. A propeller is a compressor without cowling.

The present study deals with the problem of transonic flow around compressor or propeller blades. From the mathematical point of view, this leads us to a system of partial differential equations of mixed type, in which the unknowns are the geometric and physical characteristics of the compressor. For a three-dimensional analysis these equations are too complex to be integrated without some simplifications. First of all, we suppose that the fluid we are concerned with is a polytropic and nonviscous gas and that a velocity potential exists. This necessitates a further hypothesis, namely, that the variation of the entropy is small so that the entropy itself remains essentially constant. Thus our system of partial differential equations becomes equivalent to a single equation, the potential equation, in both the steady and the time-dependent cases.

The potential equation has already been solved numerically in the three-dimensional case for oblique and swept-wings [2,14,15]. However, because of the periodicity of the compressor problem, the square-root transformation used in those works is not practical here. We therefore propose a new transformation, which maps the surface of the blade into a plane and includes periodicity implicitly.

The scheme we use is similar to the one used by Jameson and Caughey [15] in the development of the swept-wing code known as FLO22. We solve the finite-difference approximation to the potential equation by row relaxation. A typical run consists of 50 iterations on a $48 \times 6 \times 4$ grid, followed by 100 iterations on a $96 \times 12 \times 8$ refined grid. This takes 15 minutes on the CDC 6600 and 3 minutes on the STAR. The simplicity of our grid generation, together with the other hypotheses given above, limits our choice of blade geometries. Nevertheless, this method enables us to study how the speed of rotation influences the relative flow around the blades, and we have compared our three-dimensional results with data for two-dimensional cascades.

In Section 2 we shall derive the equations of motion in physical space and, in Section 3, we shall consider the potential equation, which is obtained after several changes of variable. The numerical scheme is presented in Section 4. The numerical results obtained are shown in Section 5 together with some Calcomp plots, while Section 6 provides a manual on the use of our computer code. Sections 7 and 8 contain a bibliography and a listing of the code.

This study has been supported by NASA under Grant No. NGR-33-016-201 and by the U. S. Department of Energy under Contract DE-AC-02-76ER03077 and I take this opportunity of expressing my sincere gratitude to Prof. P. Garabedian for his invaluable advice and to Dr. F. Bauer for her constant encouragement. I am also indebted to my entire family for their moral support.

II. THE PARTIAL DIFFERENTIAL EQUATIONS OF MOTION

This section sets forth the various equations used in our computational method and in the resulting code.

The general equations of fluid dynamics for an inviscid gas are well known [6]. These are:

(a) the equation of conservation of mass,

$$\rho_t + \rho u_x + \rho v_y + \rho w_z + \rho_x u + \rho_y v + \rho_z w = 0, \tag{1}$$

where ρ is the density, (u,v,w) are the velocity components, and (x,y,z) are the coordinates in physical space;

(b) the equations of conservation of momentum,

$$\begin{aligned} \rho(u_t + uu_x + vu_y + wu_z) + P_x &= 0 \\ \rho(v_t + uv_x + vv_y + wv_z) + P_y &= 0 \\ \rho(w_t + uw_x + vw_y + ww_z) + P_z &= 0 \, , \end{aligned} \tag{2}$$

where P is the pressure; and

(c) the equation of conservation of energy,

$$\frac{dS}{dt} = 0 \, , \tag{3}$$

where S is the entropy. It will be recalled that

$$\frac{d}{dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z}$$

is the material time derivative. We are neglecting here such external actions as gravity.

For the following calculations we shall assume that the entropy is constant throughout the fluid; the last equation will therefore not be used. But then, in order to represent weak shock waves which occur in transonic flows, we have to replace the Rankine Hugoniot shock conditions. This is done by permitting a jump in the horizontal component of momentum and by adding artificial viscosity terms to the partial differential equations. The approximation is adequate for Mach numbers close to 1.

Let us suppose now that the fluid is a polytropic gas. Therein:

$$P = A(S)\rho^{\gamma} \,, \tag{4}$$

where the function A(S), on the basis of our hypothesis, becomes a constant, and where

$$\gamma = \frac{C_P}{C_V}$$

is the adiabatic exponent of the gas, i.e. the ratio of the specific heats at constant pressure and at constant volume.

If c is the local speed of sound in the gas, we have

$$c^2 = \frac{dP}{d\rho} \,, \tag{5}$$

so that equations (2) become

$$\begin{aligned} &\rho(u_t + uu_x + vu_y + wu_z) + c^2\rho_x = 0 \,, \\ &\rho(v_t + uv_x + vv_y + wv_z) + c^2\rho_y = 0 \,, \\ &\rho(w_t + uw_x + vw_y + ww_z) + c^2\rho_z = 0 \,. \end{aligned} \tag{6}$$

Suppose now that the flow is steady, i.e. that the flow is independent of time, so that

$$(7) \qquad \frac{\partial\rho}{\partial t} = 0\,, \quad \frac{\partial u}{\partial t} = 0\,, \quad \frac{\partial v}{\partial t} = 0\,, \quad \frac{\partial w}{\partial t} = 0\,.$$

Then from (6) and (7) we obtain

$$(8) \quad \rho(u_x + v_y + w_z) - \frac{\rho}{c^2}(u^2u_x + uvu_y + uwu_z + vuv_x + v^2v_y + vwv_z + wuw_x + wvw_y + w^2w_z) = 0\,,$$

or, if we assume the existence of a velocity potential, ϕ, and if we divide by ρ/c^2,

$$(9) \quad c^2(\phi_{xx} + \phi_{yy} + \phi_{zz}) - (u^2\phi_{xx} + v^2\phi_{yy} + w^2\phi_{zz} + 2uv\phi_{xy} + 2vw\phi_{yz} + 2wu\phi_{zx}) = 0\,,$$

with $u = \phi_x$, $v = \phi_y$, $w = \phi_z$. This is the potential equation. It is hyperbolic for supersonic flow,

$$u^2 + v^2 + w^2 > c^2\,,$$

and elliptic for subsonic flow,

$$u^2 + v^2 + w^2 < c^2\,.$$

Equations (4) and (5) show us that

$$\frac{1}{\gamma-1}\frac{dc^2}{c^2} = \frac{d\rho}{\rho} \quad (10)$$

Using the velocity potential, equations (6) become

$$\left(\frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) + \frac{c^2}{\gamma-1}\right)_x = 0,$$

$$\left(\frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) + \frac{c^2}{\gamma-1}\right)_y = 0, \quad (11)$$

$$\left(\frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) + \frac{c^2}{\gamma-1}\right)_z = 0,$$

so that

$$\frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) + \frac{c^2}{\gamma-1} = \text{constant} \quad (12)$$

throughout the fluid. This is the Bernoulli equation, which enables us, knowing the velocity potential, to compute the speed of sound.

Suppose next that the flow is no longer steady with respect to the initial frame of reference, but that, if we consider a frame turning around the x-axis with a constant speed of rotation, ω, the flow is again independent of time. Then, instead of (7), we have

$$\frac{\partial\rho}{\partial t} = -\omega z\frac{\partial\rho}{\partial y} + \omega y\frac{\partial\rho}{\partial z}, \quad \frac{\partial u}{\partial t} = -\omega z\frac{\partial u}{\partial y} + \omega y\frac{\partial u}{\partial z}, \quad (13)$$

$$\frac{\partial v}{\partial t} = -\omega z\frac{\partial v}{\partial y} + \omega y\frac{\partial v}{\partial z}, \quad \frac{\partial w}{\partial t} = -\omega z\frac{\partial w}{\partial y} + \omega y\frac{\partial w}{\partial z}.$$

By introducing the cylindrical coordinates (x,θ,R), the equations (13) become

$$\frac{\partial\rho}{\partial t} = -\omega\frac{\partial\rho}{\partial\theta}\ , \quad \frac{\partial u}{\partial t} = -\omega\frac{\partial u}{\partial\theta}\ , \quad \frac{\partial v}{\partial t} = -\omega\frac{\partial v}{\partial\theta}\ , \quad \frac{\partial w}{\partial t} = -\omega\frac{\partial w}{\partial\theta}\ .$$

These new equations show us that the flow dependence on t and θ is characterized by a dependence on $(\theta - \omega t)$ alone.

In this time-dependent case a combination of equations (1) and (6) gives us

$$(14)\quad \rho(u_x + u_y + w_z) - \frac{\rho}{c^2}(u_0^2 u_x + u_0 v_0 u_y + u_0 w_0 u_z + v_0 u_0 v_x + v_0^2 v_y + v_0 w_0 v_z + w_0 u_0 w_x + w_0 v_0 w_y + w_0^2 w_z) = 0\ ,$$

where (u_0,v_0,w_0) are the components of the relative velocity defined by

$$(15)\qquad u_0 = u\ , \quad v_0 = v - \omega z\ , \quad w_0 = w + \omega y\ .$$

If we again assume the existence of a velocity potential, it must now satisfy the equation,

$$(16)\quad c^2(\phi_{xx}+\phi_{yy}+\phi_{zz}) - (u_0^2\phi_{xx} + v_0^2\phi_{yy} + w_0^2\phi_{zz} + 2u_0v_0\phi_{xy} + 2v_0w_0\phi_{yz} + 2w_0u_0\phi_{zx}) = 0$$

This equation is similar to equation (9); it is a second-order nonlinear partial differential equation which is hyperbolic for

$$u_0^2 + v_0^2 + w_0^2 > c^2\ ,$$

and elliptic for

$$u_0^2 + v_0^2 + w_0^2 < c^2 ,$$

We still have the relation (10) for the speed of sound, so that, using the velocity potential and with the help of (13), we obtain, instead of (6),

$$(17) \qquad \left(\frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) - \omega z\phi_y + \omega y\phi_z + \frac{c^2}{\gamma-1}\right)_x = 0 ,$$

and this gives us the new Bernoulli equation,

$$(18) \qquad \frac{1}{2}(\phi_x^2+\phi_y^2+\phi_z^2) - \omega z\phi_y + \omega y\phi_z + \frac{c^2}{\gamma-1} = \text{Constant}$$

along each line parallel to the x-axis.

In the next section we shall transform these equations, by a change of coordinates, into the system which is solved by our code.

III. GRID GENERATION

We have obtained the equations to be solved, but we have still to impose the geometrical constraints due to the slip condition on the surface of the compressor blade and to the periodicity condition. In this section, we shall describe the geometrical space in which the equations will be solved. We shall therefore list all the mappings which are performed to transform the physical space onto the computational space.

For the purpose of the periodicity condition, we begin by introducing the angle θ of the cylindrical coordinates (x,θ,R) with respect to the x-axis. To accomplish this we use a conformal mapping

(19) $$(x,y,z) \rightarrow (x_0,\theta,Z_0)$$

defined by

$$(z+iy) = \exp(Z_0+i\theta), \quad x = x_0 .$$

The Jacobian matrix associated with this transformation is

(20) $$J_1 = \begin{matrix} 1 & 0 & 0 \\ 0 & P & Q \\ 0 & -Q & P \end{matrix} ,$$

with elements defined by

$$P = \frac{\cos\theta}{R}, \quad Q = \frac{\sin\theta}{R}, \quad R = \exp(Z_0) .$$

We also require the inverse matrix, namely,

$$J_1^{-1} = \begin{matrix} 1 & 0 & 0 \\ 0 & z & -y \\ 0 & y & z \end{matrix} .$$

If $(\vec{i},\vec{j},\vec{k})$ is the orthonormal basis for physical space, then the basis vectors connected with the new coordinates are $\vec{i}$, $\vec{u}$, and $\vec{v}$, where

$$\vec{u} = z\vec{j} - y\vec{k} , \qquad \vec{v} = y\vec{j} + z\vec{k} . \tag{21}$$

For the surface of the blade, the representation of the finite difference slip condition becomes greatly simplified and more accurate if the boundary surface lies on a coordinate plane. The idea, cf. [14], is to transform the surface of the blade onto a plane which will constitute the lower boundary of a half-space.

Unfortunately, if we apply the square-root transformation of reference [14], the periodic strip is transformed in such a way that the periodicity condition is difficult to satisfy. We therefore prefer a transformation which would map a periodic strip conformally onto a half-space, so that the periodicity condition becomes implicit.

The required transformation can be decomposed into two successive mappings. For the purpose of simplification we consider only plane sections orthogonal to the Z_0-axis. The first mapping transforms a periodic strip (cf. Figure 1a) onto a slit plane (cf. Figure 1b). The image of the two lines delimiting the strip is the negative real axis. We then apply a square-root transformation which maps this plane onto a half-plane (cf. Figure 1c).

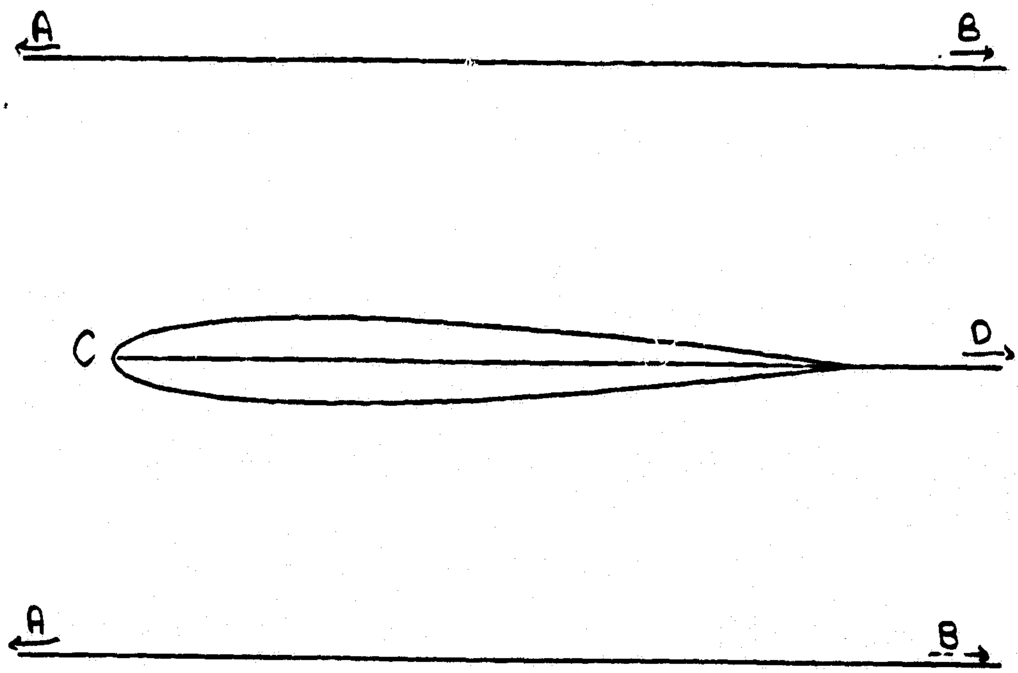


Figure 1a. The periodic strip.

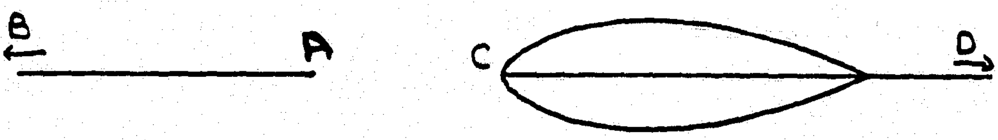


Figure 1b. The slit plane.

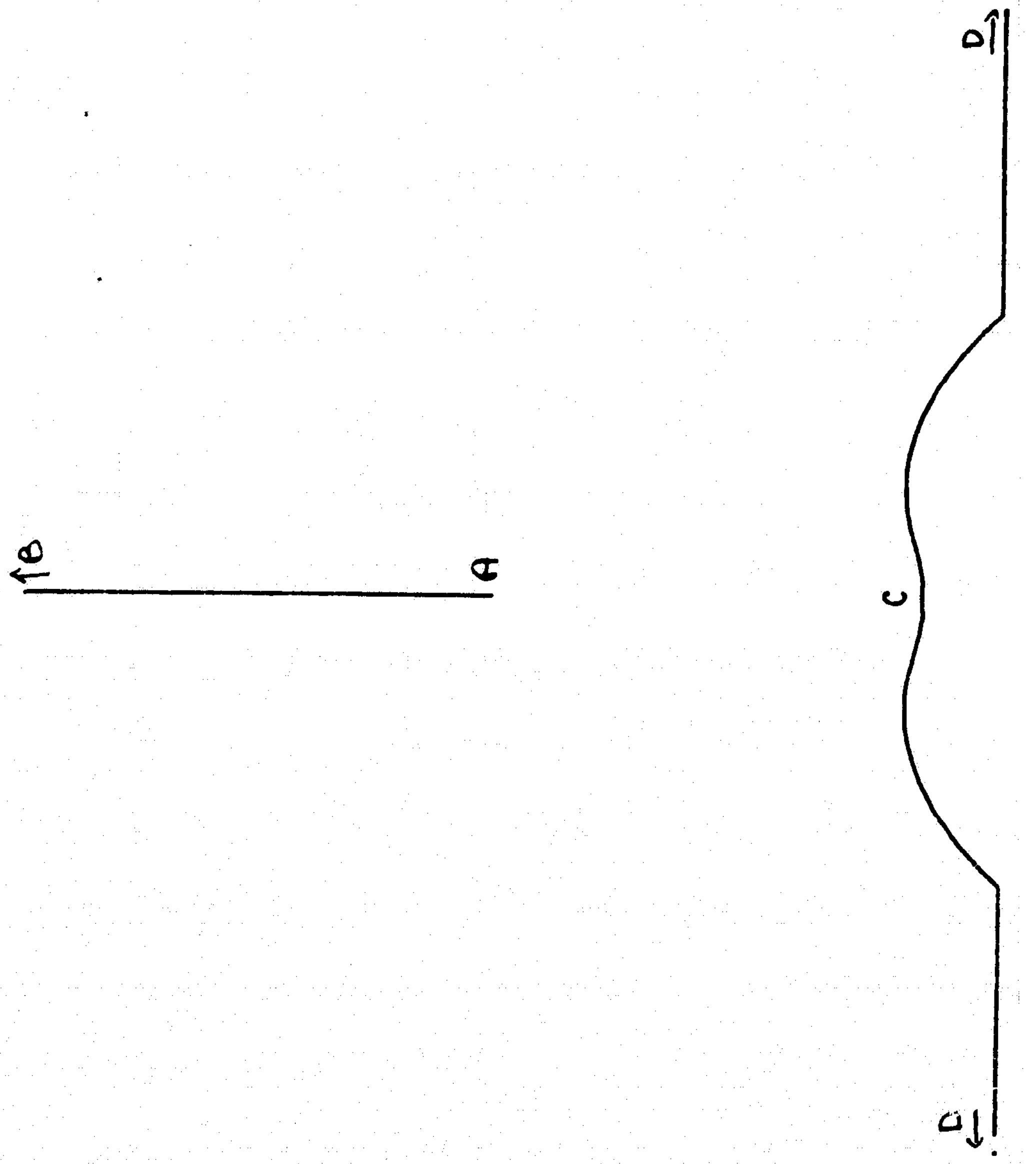


Figure 1c. The half plane with a bump.

To implement these transformations we draw in each plane section a singular line (the branch cut) from inside the blade, near the nose, out to downstream infinity. This is actually a half-line whose origin has the coordinates (x_s, θ_s) . We next perform the mapping

(22) $$(x_0, \theta, Z_0) \to (X, Y, Z) ,$$

defined by

$$(x_0 - x_s) + i(\theta - \theta_s) = N^{-1} \log\left(1 + \frac{(X+iY)^2}{T}\right) , \quad Z_0 = Z ,$$

where N and T are two constants. N is more precisely the number of blades on the compressor. The coordinates x_s and θ_s usually depend on Z. Their derivatives with respect to Z will be denoted x_z and θ_z , respectively.

The Jacobian matrix determined by this transformation is

(23) $$J_2 = \begin{matrix} a & -b & 0 \\ b & a & 0 \\ e & f & 1 \end{matrix} ,$$

whose elements are defined by

$$a = Hx_X , \; b = H\theta_X , \; e = -ax_Z - b\theta_Z , \quad f = -a\theta_Z + bx_Z , \quad H = x_X^2 + \theta_X^2 .$$

The inverse matrix is

$$J_2^{-1} = \begin{matrix} x_X & \theta_X & 0 \\ -\theta_X & x_X & 0 \\ x_Z & \theta_Z & 1 \end{matrix} ,$$

and the basis vectors related to this transformation are now $(\vec{A},\vec{B},\vec{C})$, with

(24) $\vec{A} = x_X\vec{i} + \theta_X\vec{u}$, $\vec{B} = -\theta_X\vec{i} + x_X\vec{u}$, $\vec{C} = x_Z\vec{i} + \theta_Z\vec{u} + \vec{v}$

Let us return to the potential equation (9), it can be reformulated as

(25) $$c^2\nabla^2\phi - (\nabla\phi \cdot \nabla)^2\phi = 0 ,$$

where

$$\nabla = \frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}\vec{k}$$

is a notation for the gradient. If we denote the final Jacobian matrix

$$A = J_1 \cdot J_2$$

by

$$A = (a_{ij})$$

and, if we use

$$d_1 = \frac{\partial}{\partial X} , \quad d_2 = \frac{\partial}{\partial Y} , \quad d_3 = \frac{\partial}{\partial Z}$$

for the derivatives with respect to the new coordinates, then the derivatives with respect to the initial coordinates are represented by

$$\frac{\partial}{\partial x} = \sum_j a_{1j} d_j \ , \quad \frac{\partial}{\partial y} = \sum_j a_{2j} d_j \ , \quad \frac{\partial}{\partial z} = \sum_j a_{3j} d_j \ .$$

Thus the Laplacian, which is defined by

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \ ,$$

can be rewritten as

$$(26) \quad \nabla^2 = \sum_{i,j,k} a_{ij} a_{ik} d_j d_k + \sum_{i,j,k} a_{ij} a^j_{ik} d_k \ ,$$

where the coefficients a^k_{ij} stand for the derivatives of the Jacobian matrix elements;

$$a^j_{ik} = d_j \, a_{ik} \ .$$

Let us set

$$(F_{ij}) = F = A^t \cdot A \ , \quad D_1 = \nabla^2 X \ , \quad D_2 = \nabla^2 Y \ , \quad D_3 = \nabla^2 Z \ .$$

These notations allow us to simplify the equation (26) to arrive at

$$(27) \qquad \nabla^2 = \sum_{j,k} F_{j,k} \, d_j d_k + \sum_k D_k d_k \ .$$

For the second term of equation (25) let us denote the components of the velocity in the physical space by

$$\phi_1 = \frac{\partial \phi}{\partial x}, \quad \phi_2 = \frac{\partial \phi}{\partial y}, \quad \phi_3 = \frac{\partial \phi}{\partial z}.$$

Using the same computation as for the Laplacian, we obtain

$$(28) \qquad (\nabla\phi\cdot\nabla)^2 = \sum_{k,\ell} B_k B_\ell d_k d_\ell + \sum_\ell C_\ell d_\ell ,$$

where (B_1, B_2, B_3), the components of the velocity in the basis $(\vec{A}, \vec{B}, \vec{C})$, are given by

$$B_j = \sum_i a_{ij}\phi_i ,$$

and where the coefficients C_ℓ are defined by

$$C_1 = (\nabla\phi\cdot\nabla)^2 X , \quad C_2 = (\nabla\phi\cdot\nabla)^2 Y, \quad C_3 = (\nabla\phi\cdot\nabla)^2 Z .$$

After these necessary but somewhat tedious transformations the potential equation acquires the useful form

$$(29) \qquad \sum_{i,j} (c^2 F_{ij} - B_i B_j) d_i d_j \phi + \sum_i (c^2 D_i - C_i) d_i \phi = 0 .$$

This has the advantage of being relatively tractable, for an equation which is, really, quite complicated. In the time-dependent case, we need only to replace the ϕ_j's by the components of the relative velocity in the calculation of the coefficients of equation (29). Moreover, if a reduced potential is defined as the difference between the true potential and the uniform flow potential corresponding to the inlet speed, we can use it to compute the derivatives $d_i d_j \phi$ and $d_i \phi$ in equation (29) without changing that equation.

We next describe the grid used in our computational method. It is obtained with the help of a system of sheared coordinates. These are defined by considering coordinates parallel to the transformed surface of the blade. If

$$Y = S(X,Z)$$

is the equation of this surface, the transformation in question is defined by setting

$$I = X \; , \quad J = Y - S(X,Z) \; , \quad K = Z \; .$$

The Jacobian matrix is

$$(30) \qquad J_3 = \begin{matrix} 1 & -S_X & 0 \\ 0 & 1 & 0 \\ 0 & -S_Z & 1 \end{matrix} \; ,$$

and the related basis vectors are

$$(31) \qquad \vec{I} = \vec{A} + S_X \vec{B} \; , \quad \vec{J} = \vec{B} \; , \quad \vec{K} = \vec{C} + S_Z \vec{B} \; .$$

In order to carry out the computations on a finite domain, these last coordinates are stretched so that the final domain of computation becomes a cube the edges of which have length 1. After these transformations equation (29) remains of the same general type, but its coefficients have to be changed slightly. Complete details are specified in the listing of the code in Section VIII.

It remains to define the boundary conditions at the hub, at the cowling, and on the blade surface. We impose a slip condition for the flow on these boundaries; this leads to a Neumann problem for the velocity potential, since the normal derivatives at the boundaries must vanish. We shall express this condition in our system of coordinates.

The hub and the cowling are defined by equations of the form

$$Z = \text{Constant}.$$

At any boundary points the two tangential vectors are $\vec{i}$ and $\vec{u}$, and $\vec{v} = \vec{i} \times \vec{u}$ is a normal vector. The boundary condition at the hub and at the cowling may thus be expressed by the equation

$$\frac{\partial \phi}{\partial Z_0} = y\phi_2 + z\phi_1 = 0 \,. \tag{32}$$

On the other hand, the blade surface is defined by the equation

$$J = 0 \,.$$

The two tangent vectors $\vec{I}$ and $\vec{K}$ lead to the normal vector $\vec{I} \times \vec{K}$ whose coordinates in the basis $(\vec{i},\vec{u},\vec{v})$ are

$$\alpha = R^2(\theta_X + S_X x_X) \,, \quad \beta = S_X\theta_X - x_X \,, \quad \gamma = \frac{S_X e - f + S_Z}{H} \,.$$

Hence the boundary condition on the blade surface is given by

(33) $$\alpha \frac{\partial\phi}{\partial x_0} + \beta \frac{\partial\phi}{\partial\theta} + \gamma \frac{\partial\phi}{\partial z_0} = \alpha\phi_1 + (\beta z + \gamma y)\phi_2 + (\gamma z - \beta y)\phi_3 = 0 .$$

In equations (32) and (33), (ϕ_1,ϕ_2,ϕ_3) must be the derivatives of the true potential, i.e. the components of the velocity. For the time-dependent case, they are replaced, in equation (33), by the components of the relative velocity.

In the code the values of (ϕ_1,ϕ_2,ϕ_3) are computed only when necessary, and the only derivatives available are

$$U = \frac{\partial\phi}{\partial X} , \quad V = \frac{\partial\phi}{\partial Y} , \quad W = \frac{\partial\phi}{\partial Z} .$$

These derivatives are related to the components of the velocity through the relation

(34) $$\begin{matrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{matrix} = A \begin{matrix} U \\ V \\ W \end{matrix} .$$

The boundary conditions (32) and (33) become

(35) $$eU + fV + W = 0$$

and

(36) $$(\alpha a + \beta b + \gamma e)U + (-\alpha b + \beta a + \gamma f)V + \gamma W = 0 .$$

In three dimensional space an appropriate vortex sheet behind a blade must be considered. The shape of the vortex sheet is modeled by our code in the following way. In each plane cross-section, the upper and lower surfaces of the blade are extended behind the trailing edge by two lines parallel to a branch cut (Cf. Figures 1,4,7). These two lines represent the vortex sheet in that section, and if the blade is closed they are identical. For two points situated on opposite sides of the vortex sheet the pressures are the same, and the normal velocities are zero; only the tangential components of the velocity may be different. Since a shape is assumed for the vortex sheet, we require only continuity of the normal component of the velocity across the sheet. Computationally this reduces to a condition like

$$\phi_{YY} = 0 \tag{37}$$

after a jump in ϕ is removed. Moreover, the jump of the potential ϕ across the vortex sheet is supposed to be constant in each plane section. The Kutta-Joukowsky condition at the trailing edge determines this jump. This amounts to a linearized treatment of the vortex sheet.

IV. FINITE DIFFERENCE SCHEME

The success of codes for the design and analysis of supercritical wings [1,2,3,15] shows how effective the computational fluid dynamics has become for transonic flow. The first step in this development was the design of shockless airfoils by the hodograph method [1]. Then the introduction of a retarded difference scheme [20] allowed the analysis of flow at off-design conditions. This scheme incorporated artificial viscosity in order to capture shocks in the supersonic zone. It was then improved to permit the analysis of three-dimensional wings [14] and their design [11]. In our computational method we use the last-mentioned scheme, and this section explains how it is incorporated into our computer code.

All the equations we have described in the previous section are represented in the computer code by finite difference approximations. The flow conditions at each grid point are determined with the help of the reduced potential G. The first derivatives of this potential are calculated by using central differences. With these values we compute the approximate velocity. This allows us to determine, with the help of Bernoulli's equation (12), whether, at the point considered, the flow is subsonic or supersonic.

If the flow is subsonic the second derivatives used for the computation of the potential equation residual are approximated by central differences of the form

$$G_{XX} = \frac{G_{i+1,j,k} - 2G_{i,j,k} + G_{i-1,j,k}}{(\Delta X)^2}, \tag{38}$$

and

$$(39) \qquad G_{XY} = \frac{G_{i+1,j+1,k} - G_{i+1,j-1,k} - G_{i-1,j+1,k} + G_{i-1,j-1,k}}{4\Delta X\,\Delta Y} .$$

For supersonic points, on the other hand, we have to introduce artificial viscosity in order to capture weak shocks. This is accomplished by using a retarded difference scheme [15]. Thus we separate the equation (29) into two groups of terms:

$$(40) \qquad (c^2-q^2)\,G_{ss} + c^2(\nabla^2 G - G_{ss}) = 0 ,$$

where q is the speed and s a coordinate in the flow direction. The first term represents the second derivatives in the flow direction; the derivatives in the other directions form the second term. The second derivatives used in this second expression are computed by using equations (38) and (39). But for the second derivatives of the first expression we use retarded difference approximations of the following types:

$$(41) \qquad G_{XX} = \frac{G_{i,j,k} - 2G_{i-1,j,k} + G_{i-2,j,k}}{(\Delta X)^2} ,$$

and

$$(42) \qquad G_{XY} = \frac{G_{i,j,k} - G_{i,j-1,k} - G_{i-1,j,k} + G_{i-1,j-1,k}}{\Delta X\,\Delta Y}$$

provided that the velocity has positive components in the X direction and in the Y direction. Equations (41) and (42) are only first-order accurate and introduce the truncation errors

$$- \Delta X G_{XXX} \,, \qquad - \frac{\Delta X G_{XXY} + \Delta Y \; G_{XYY}}{2} \,.$$

For the potential equation (40), at supersonic points, these terms represent a positive artificial viscosity which, when the flow is aligned with the X direction, reduces to

$$(q^2 - c^2) \, \Delta X \, G_{XXX}$$

as in the scheme of Murman and Cole [20].

Equations (38) to (42) are used at each iteration of a run. To describe the iteration process, which is done by row relaxation, it is helpful to introduce an artificial time t, which increases by the quantity Δt at each iteration. The right-hand sides of equations (38), (39) and (41) become

$$\frac{G^0_{i+1,j,k} - 2(1-1/\omega) \, G^0_{i\,j,k} - (2/\omega) \, G^N_{i,j,k} + G^N_{i-1,j,k}}{(\Delta X)^2} \,,$$

$$\frac{G^0_{i+1,j+1,k} - G^0_{i-1,j+1,k} - G^N_{i+1,j-1,k} + G^N_{i-1,j-1,k}}{4 \Delta X \, \Delta Y} \,,$$

$$\frac{2G^N_{i,j,k} - G^0_{i,j,k} - 2G^N_{i-1,j,k} + G^0_{i-2,j,k}}{(\Delta X)^2} \,,$$

where the superscripts N and O denote new or old values of the potential and where ω is the overrelaxation factor. These expressions represent approximations to

$$G_{XX} - \frac{\Delta t}{\Delta X}\left(G_{Xt} + \frac{1}{\Delta X}\left(\frac{2}{\omega} - 1\right)G_t\right), \quad G_{XY} - \frac{1}{2}\frac{\Delta t}{\Delta X}G_{Yt}, \quad G_{XX} + 2\frac{\Delta t}{\Delta X}G_{Xt} .$$

Hence the equation solved is

$$(43) \quad \sum_{i,j} (c^2F_{ij} - B_iB_j)G_{X_iX_j} + \sum_i [(c^2D_i - C_i)G_{X_i} + \alpha_i G_{X_it}] + \delta G_t = 0 ,$$

where the coefficients α_i and δ are determined by the new approximations (38), (39) and (41).

By considering an orthonormal system of coordinates (p,r,s), where the flow direction is still the s direction, we arrive at the equation

$$(4) \quad (c^2-q^2)G_{ss} + q^2G_{rr} + q^2G_{pp} + \beta_1G_{st} + \beta_2G_{rt} + \beta_3G_{pt} + \delta G_t = 0 .$$

If

$$\tau = t - \frac{1}{2}\left(\frac{\beta_1}{c^2-q^2}s + \frac{\beta_2}{q^2}r + \frac{\beta_3}{q^2}p\right)$$

is a new time coordinate, then equation (44) is transformed into

$$(45) \quad (c^2-q^2)G_{ss} + q^2G_{rr} + q^2G_{pp} - \left(\frac{\beta_1^2}{c^2-q^2} + \frac{\beta_2^2}{q^2} + \frac{\beta_3^2}{q^2}\right)G_{\tau\tau} + \delta G_\tau = 0.$$

In order to ensure the convergence of the iteration scheme to a solution of the steady-state equation, we want equation (45) to be a damped three-dimensional wave equation. Given the form of equation (45) this is equivalent to the condition

$$(46)\qquad \left(\frac{\beta_1^2}{c^2-q^2}+\frac{\beta_2^2}{q^2}+\frac{\beta_3^2}{q^2}\right)(c^2-q^2) > 0 .$$

At subsonic points the damping condition is always satisfied. At supersonic points it may no longer be true. The choice of approximations (41) and (42) help to ensure a large coefficient β_1 , but near the sonic line this may not be enough. One way to ensure that condition (46) is satisfied, is to increase β_1 by adding a term of the form

$$\beta \frac{\Delta t}{\Delta X} G_{st} = \beta \frac{\Delta t}{\Delta X} (\phi_1 G_{xt} + \phi_2 G_{yt} + \phi_3 G_{zt}) = \beta \frac{\Delta t}{\Delta X} (B_1 G_{Xt} + B_2 G_{Yt} + B_3 G_{Zt})$$

to equation (43), where β is a positive parameter large enough to increase β_1 by the right quantity.

The mixed derivatives used above, G_{Xt} , G_{Yt} , G_{Zt} are obtained by using approximations of the type

$$G_{Xt} = \frac{G^N_{i,j,k} - G^0_{i,j,k} - G^N_{i-1,j,k} + G^0_{i-1,j,k}}{\Delta t\, \Delta X}.$$

We have described the difference scheme for interior points. For the boundary conditions at the hub and cowling, we transform equation (35) into a difference equation in order to compute the value of the reduced potential at these boundaries. For the blade surface and the vortex sheet we introduce ghost points behind the boundary. These points are used to compute the value of the potential at points on the blade as if they were interior points. The values of the potential at the ghost points behind the blade are determined by using equation (36), i.e. the slip condition. For the points behind the vortex sheet the potential is determined by the Kutta-Joukowski condition. To compute the value of G at the remaining boundary points we use an approximation to the outlet velocity.

The program permits not only compressor blades but also propellers to be modeled. Since propellers have no cowling, the inlet speed is equal to the outlet speed. Thus, for a propeller run, all the boundary conditions except those at the hub, on the blade and on the vortex sheet are replaced by

$$G = 0 .$$

With program FLO22 as the starting-point, we have used the result stated in this section to write a new program in FORTRAN named CSCDF22. This program is listed in Section VIII.

V. RESULTS

In the preceding sections we described our computational method. In order to show how this method works, we present in this section seven examples of runs made with the program CSCDF22.

The first case was run to test the validity of our code. A compressor blade had been designed with code K [3]. Figure 2 shows the result of this run. We used the coordinates of this two-dimensional blade profile to create a three-dimensional blade. The profiles were identical at each span station. Since we chose to run the program for a cascade configuration and in the compressor case, we were solving a two-dimensional cascade problem. The inlet speed and the inlet angle were the same as in the design run, but the blade was markedly cambered and we chose a gap-to-chord ratio of 1.5. Figures 3 and 4 show a representation of the cascade and the grid at the hub which is the same at each span station. The pressure distribution is given in Figure 5.

The other six examples are all run with the same blade. The profile is defined at three different span stations. The basic profile is the NACA-0012 profile. At each span station the chord and thickness are the same but, for the realistic axial flow configuration, the coordinates of the singular points are different. The blade has a sweep angle of 14 degrees and the dihedral angle decreases from 0. degrees at the hub to -10. degrees at the tip. The angle of twist is equal to 0. degrees at the hub, 2. degrees at the cowling, and 1. degree in between. For the axial flow configuration the distance between the hub and the axis is 2. This is also the distance between the hub and the cowling (or the tip).

The first example with this blade is a run for a compressor cascade configuration which appears in Figure 6. Figures 7 and 8 show a representation of the grid and the pressure distribution. The different flow parameters are shown in Figure 8. M1 is the inlet Mach number. M2 is the outline Mach number. DEV is the difference between the outlet angle and the inlet angle given by ALP.

The four next runs are also analyses of compressor flow, but now in an axial flow configuration and with eight blades around the hub. For each run the parameter OM determining the speed of rotation has different values. These values are 0.0, 0.5, 1.0 and 2.0. The other flow parameters are identical. Figures 9 and 10 represent the grid at the hub and cowling. It will be noticed how different they are, although the profile is almost the same at the hub and cowling; but the gap-to-chord ratio is twice as large at the cowling. Figure 11 shows how the blade looks in a plane orthogonal to the axis. The different pressure distributions obtained for each run are given in Figure 12, 13, 14 and 15. These figures clearly illustrate how the flow evolves as the speed of rotation increases: the shock on the upper surface weakens and then disappears while, on the other hand, a shock appears on the lower surface and intensifies.

The last example is similar to the fifth, except that it is a propeller analysis. Figure 16, 17, and 18 show the geometry of the case, and the pressure distribution is given in Figure 19. by comparing Figures 14 and 19, we observe that, for example 5, the shock appearing at the cowling which is caused by the speed of rotation, is amplified significantly by the cowling.

These numerical experiments show that our computational method, in spite of its restrictive hypotheses, enables us to analyze the three-dimensional transonic flow around compressor blades. This is a first step towards the study of compressors or propellers in three dimensions. Considering what happened in the study of transonic flow past swept wings, we venture to suggest that the next steps could be to improve this method in order to include the computation of the wave drag and to design blades with a prescribed pressure distribution [4,11].

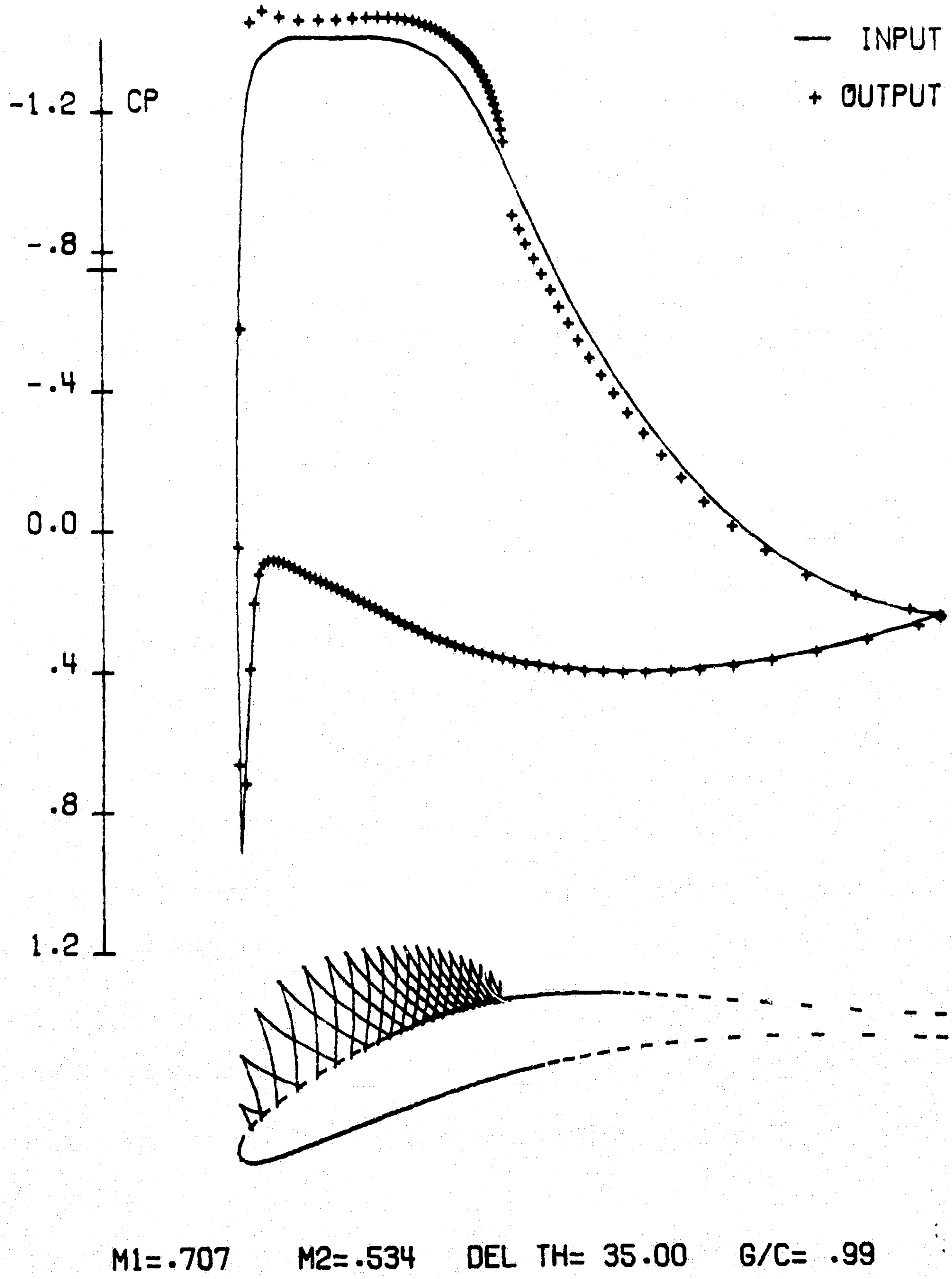


Figure 2

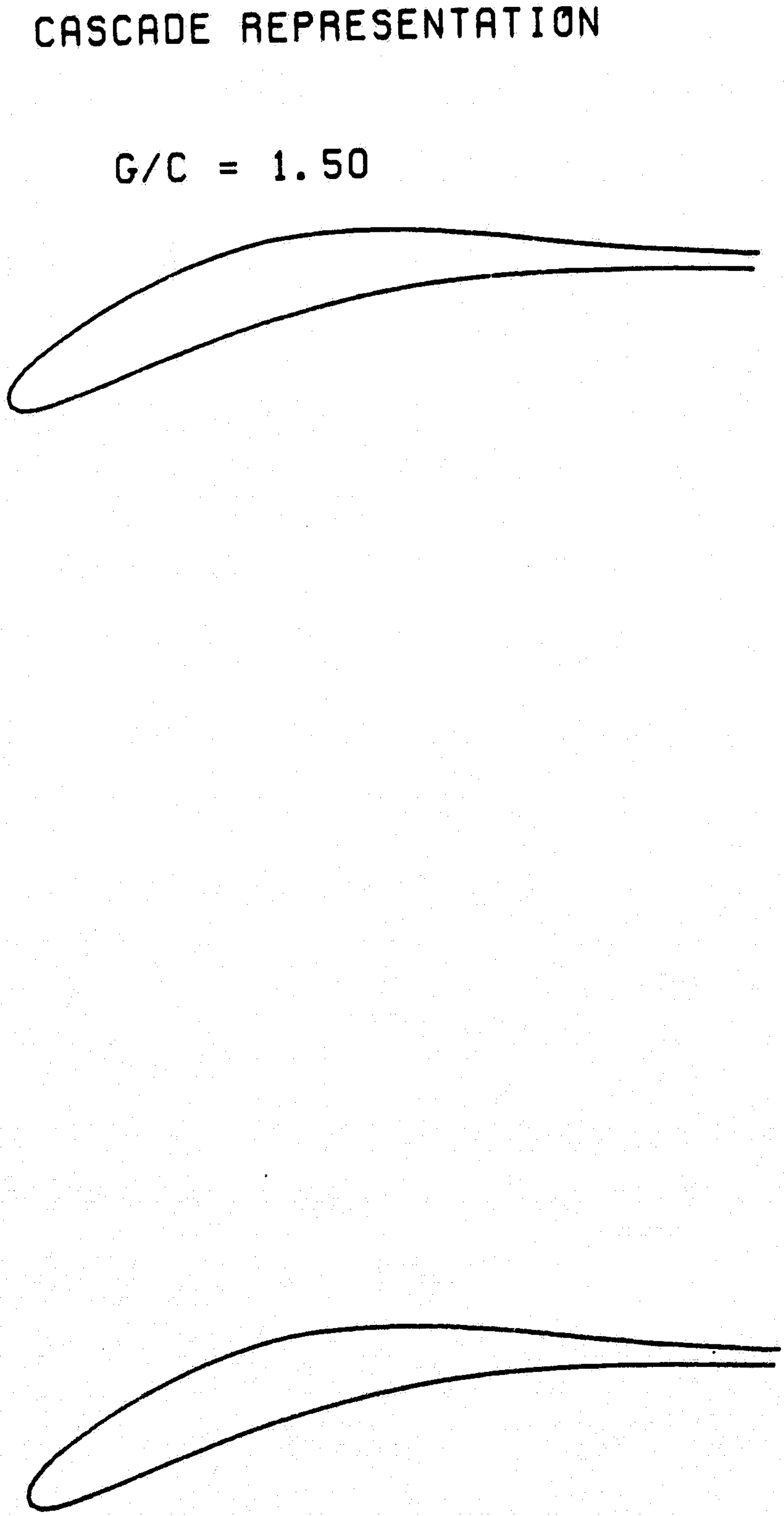


Figure 3

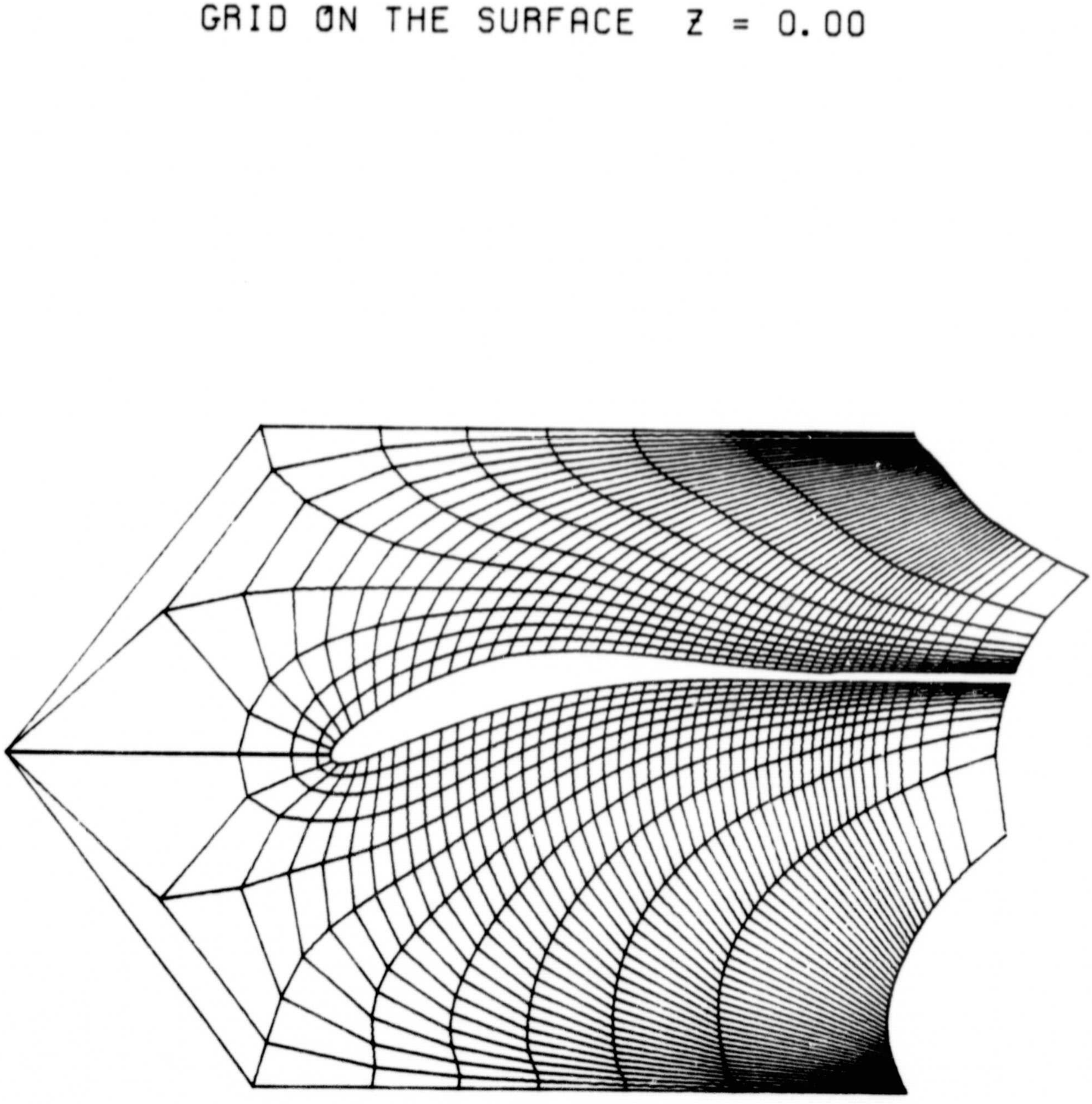
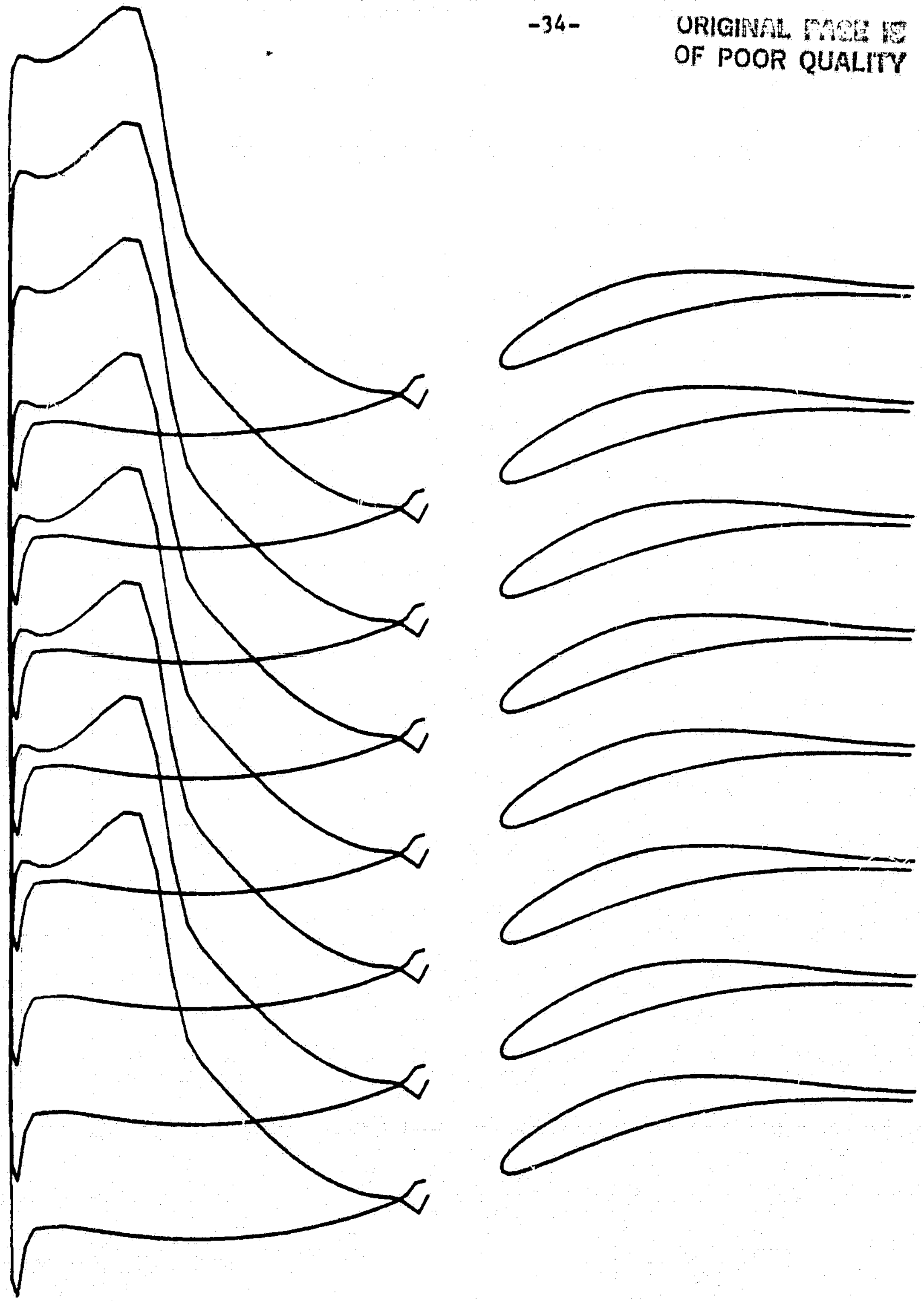


Figure 4



PRESSURE DISTRIBUTION BLADE PROFILE

M1 = .71, M2 = .57, DEV = -32.0, ALP = 35.0

Figure 5

CASCADE REPRESENTATION

G/C = 1.50

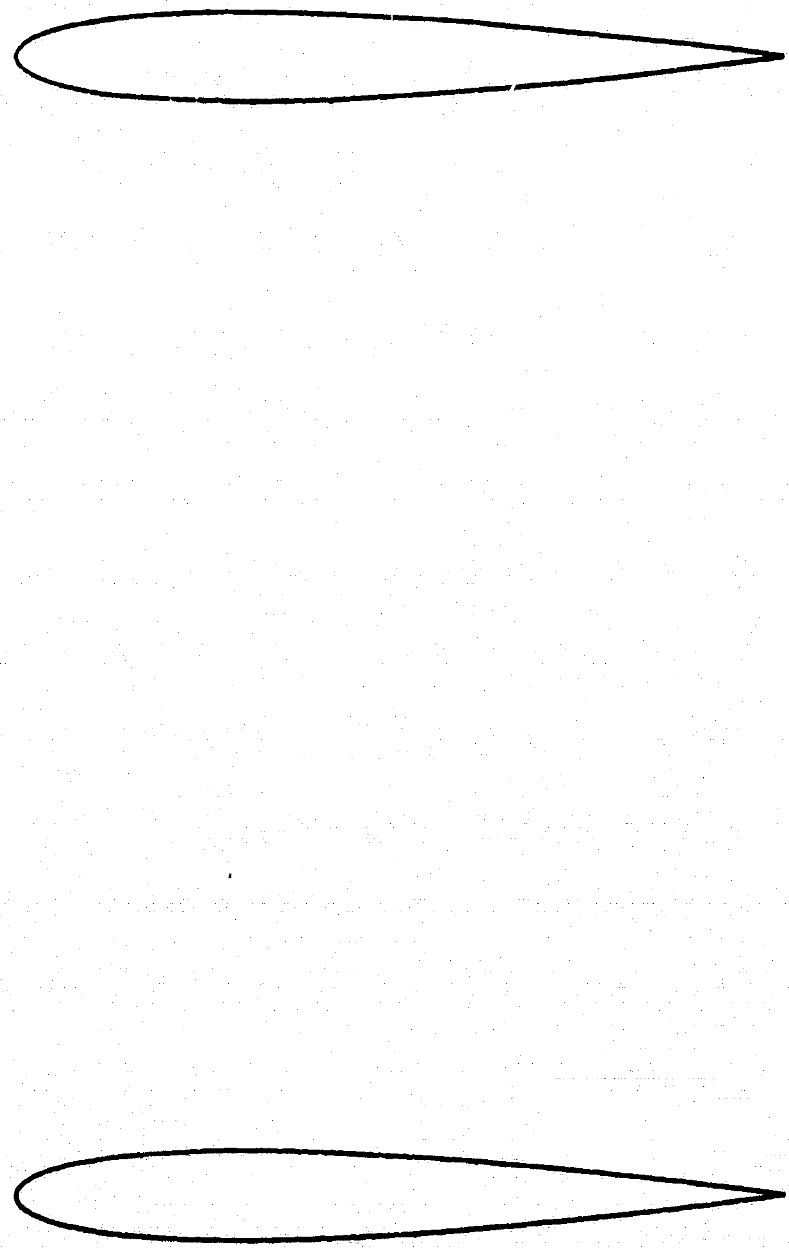


Figure 6

GRID ON THE SURFACE Z = 0.00

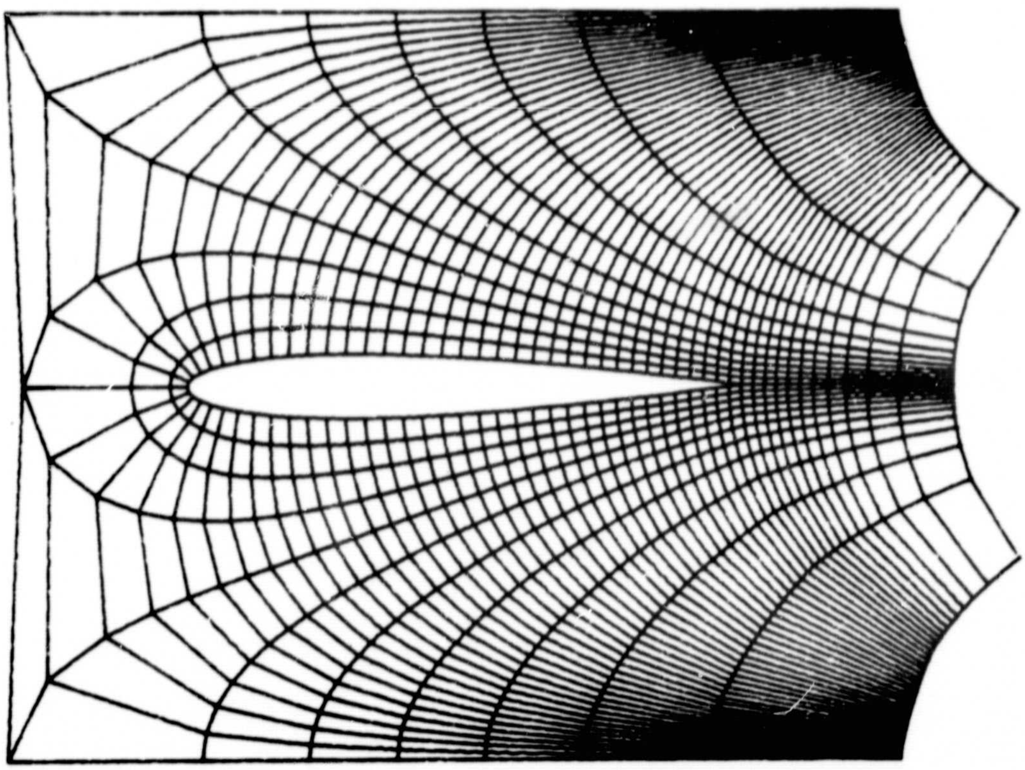
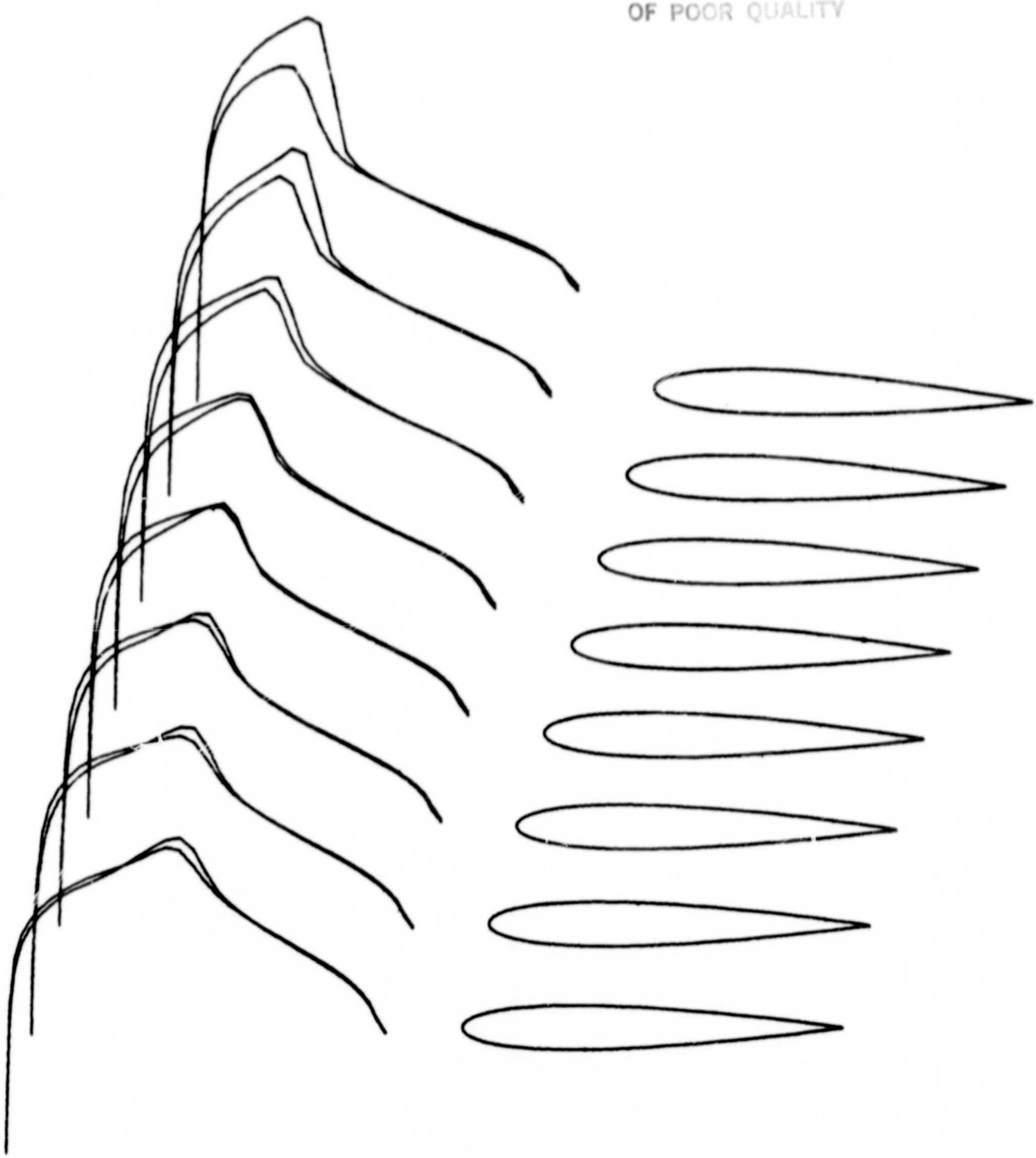


Figure 7



PRESSURE DISTRIBUTION BLADE PROFILE

M1 = .74, M2 = .70, DEV = .2, ALP = 0.0

Figure 8

GRID ON THE SURFACE Z = 2.00

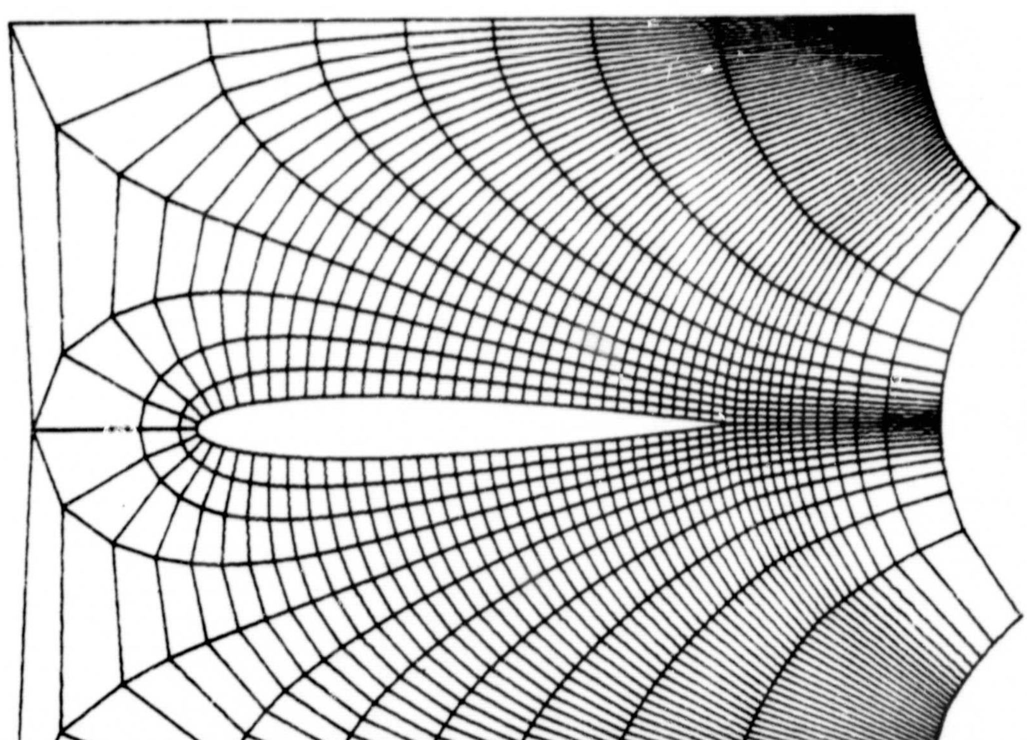


Figure 9

GRID ON THE SURFACE Z = 4.00

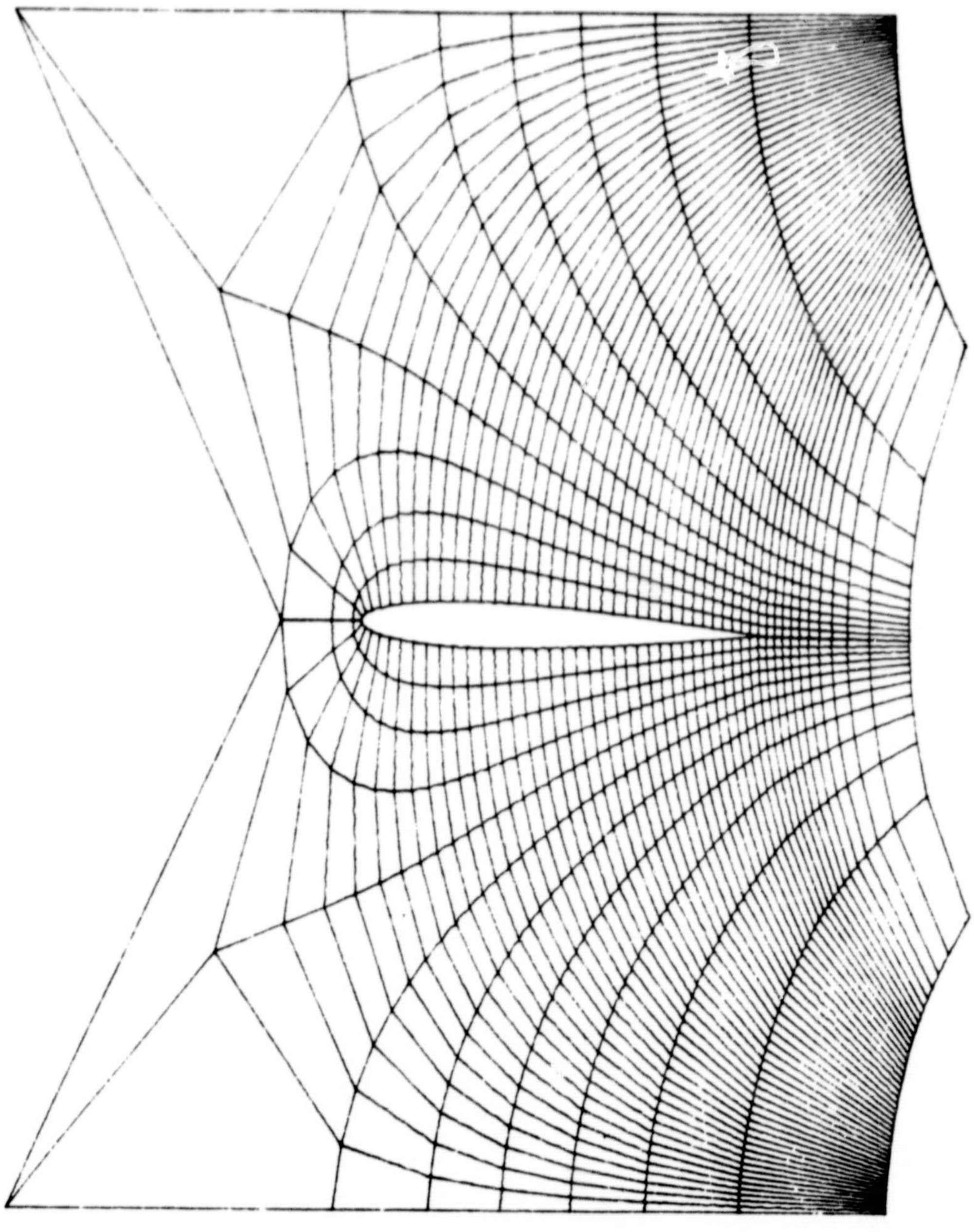


Figure 10

CUT IN THE PLANE X = .500

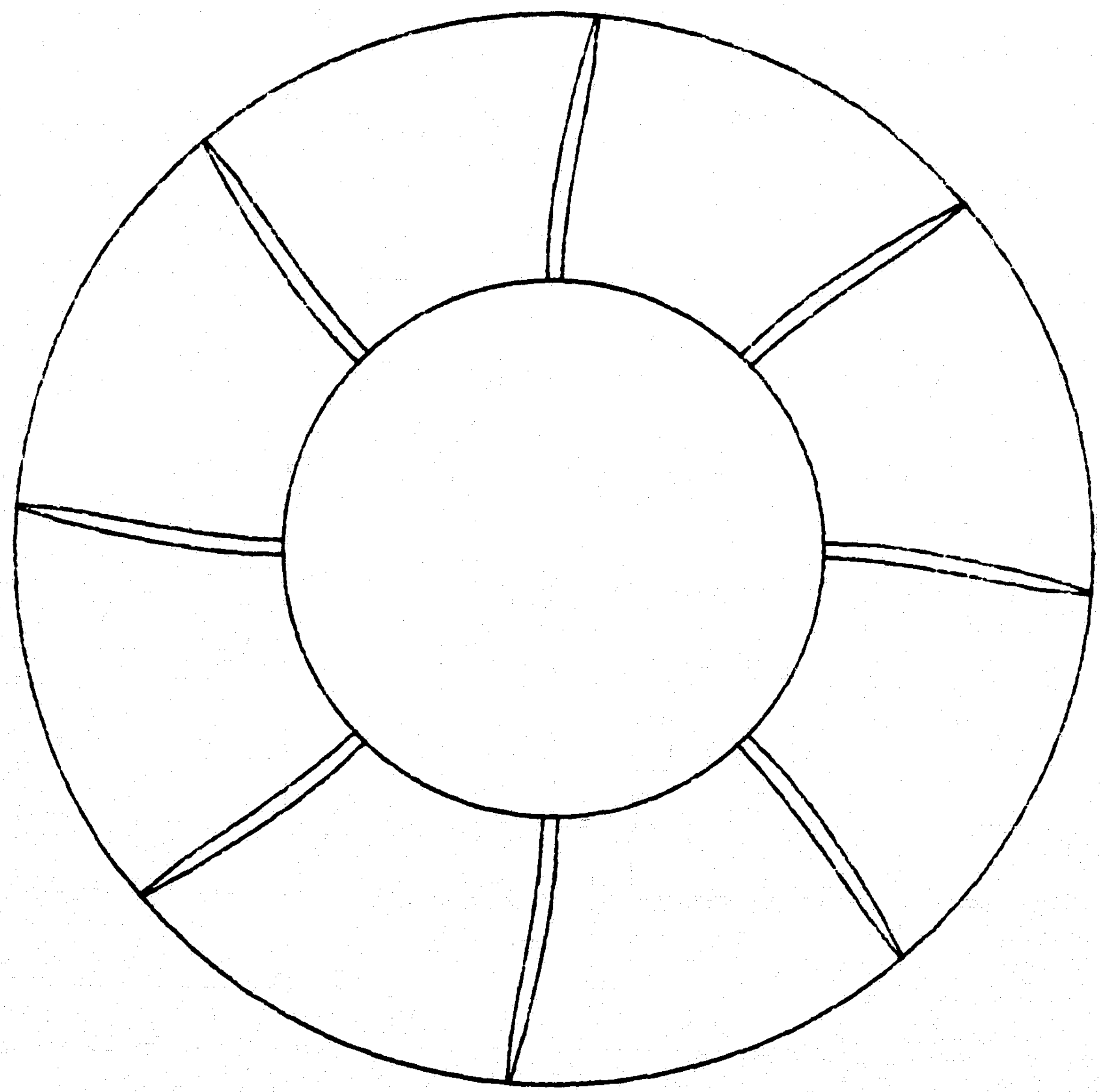


Figure 11

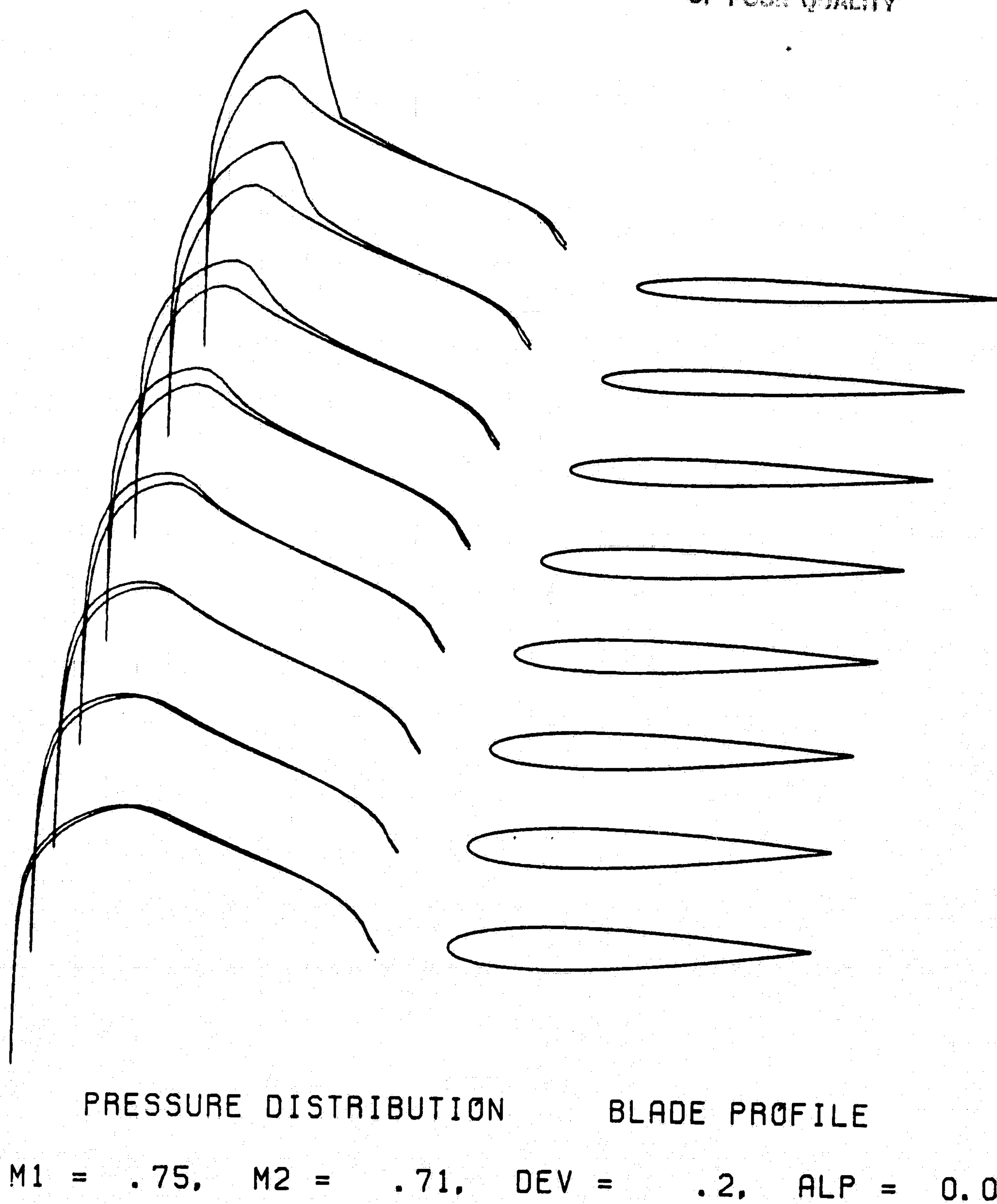


Figure 12

ORIGINAL PAGE IS
OF POOR QUALITY

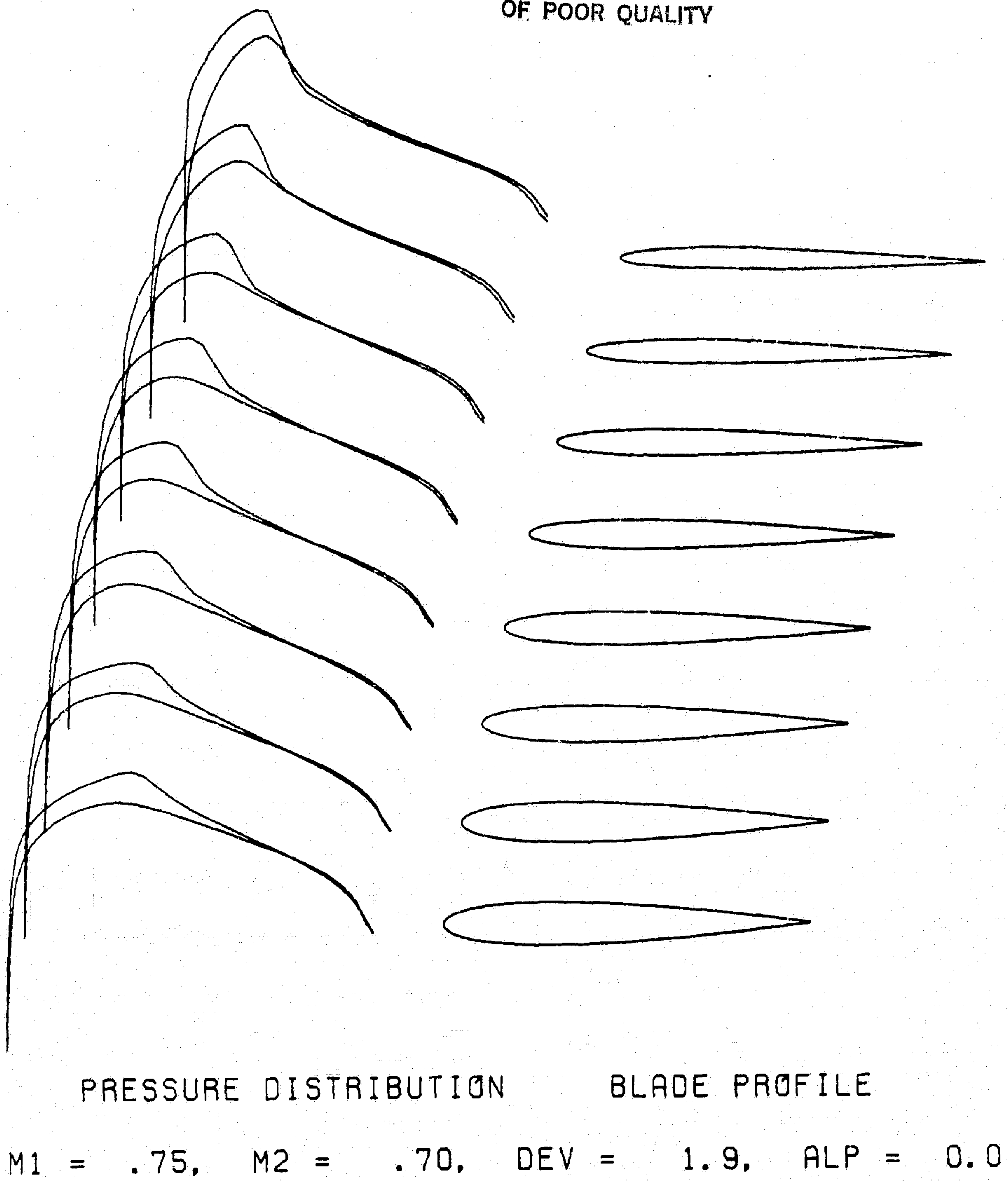


Figure 13

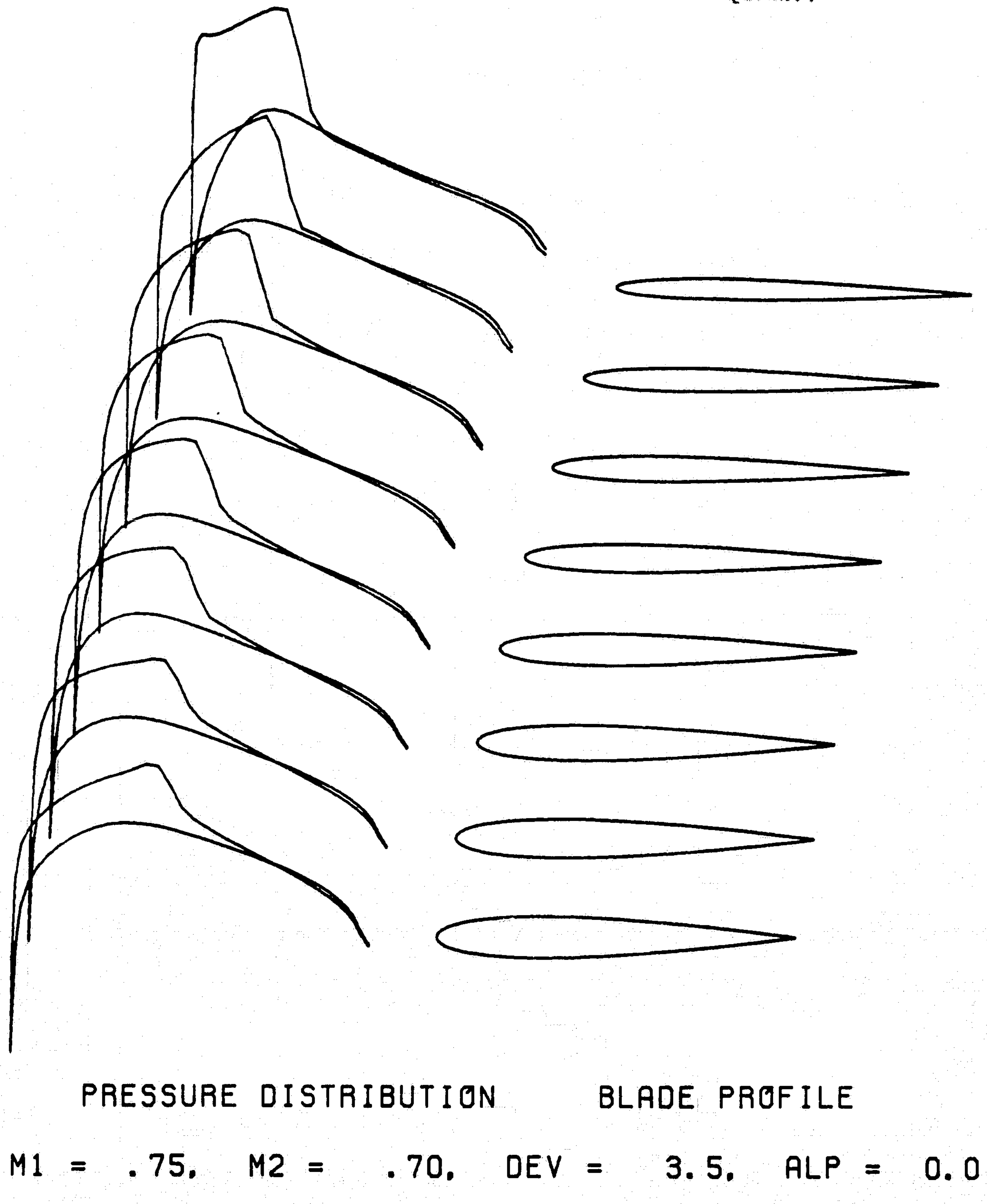

PRESSURE DISTRIBUTION
BLADE PROFILE
M1 = .75, M2 = .70, DEV = 3.5, ALP = 0.0

Figure 14

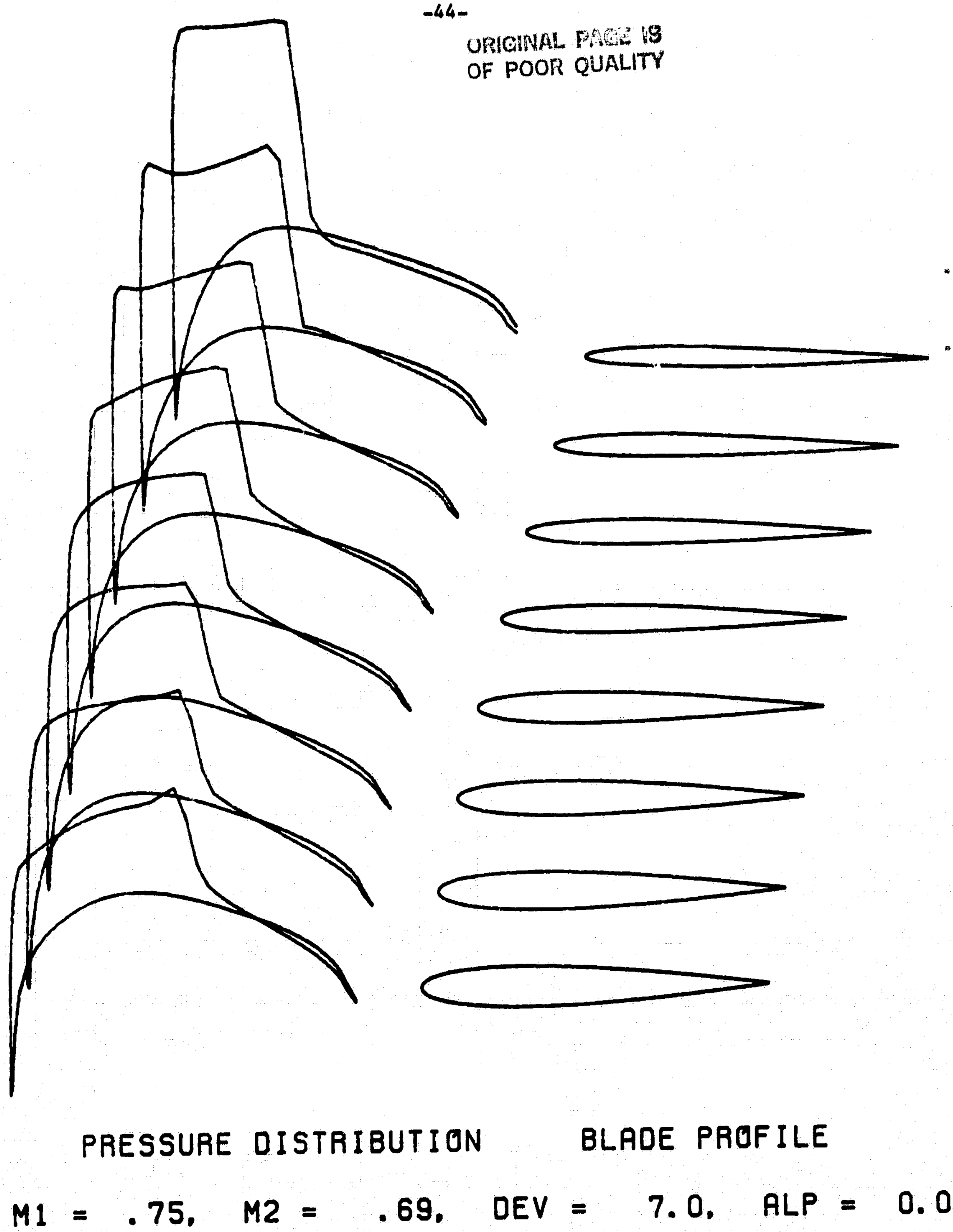

PRESSURE DISTRIBUTION
BLADE PROFILE
M1 = .75, M2 = .69, DEV = 7.0, ALP = 0.0

Figure 15

GRID ON THE SURFACE Z = 2.00

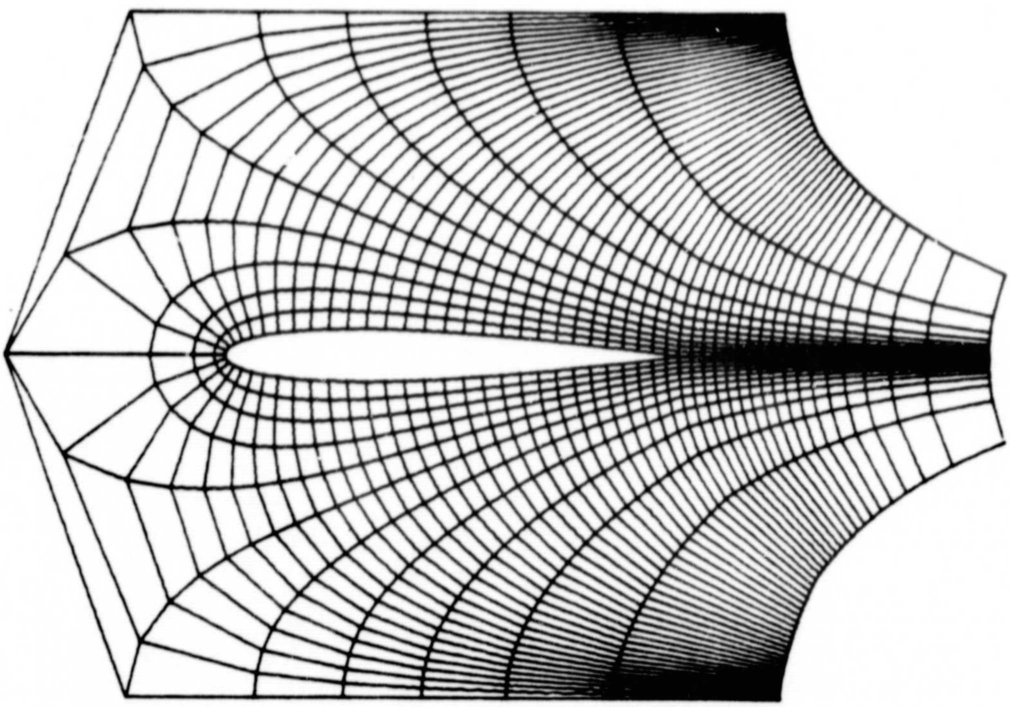


Figure 16

GRID ON THE SURFACE Z = 4.00

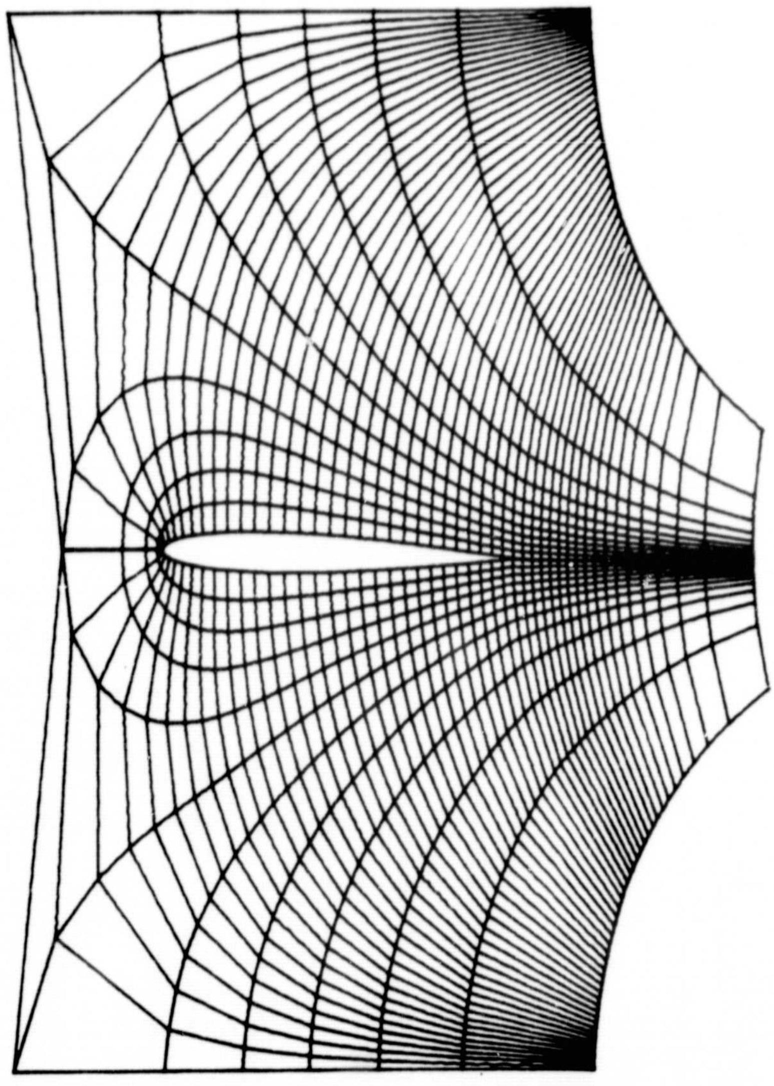


Figure 17

CUT IN THE PLANE X = .500

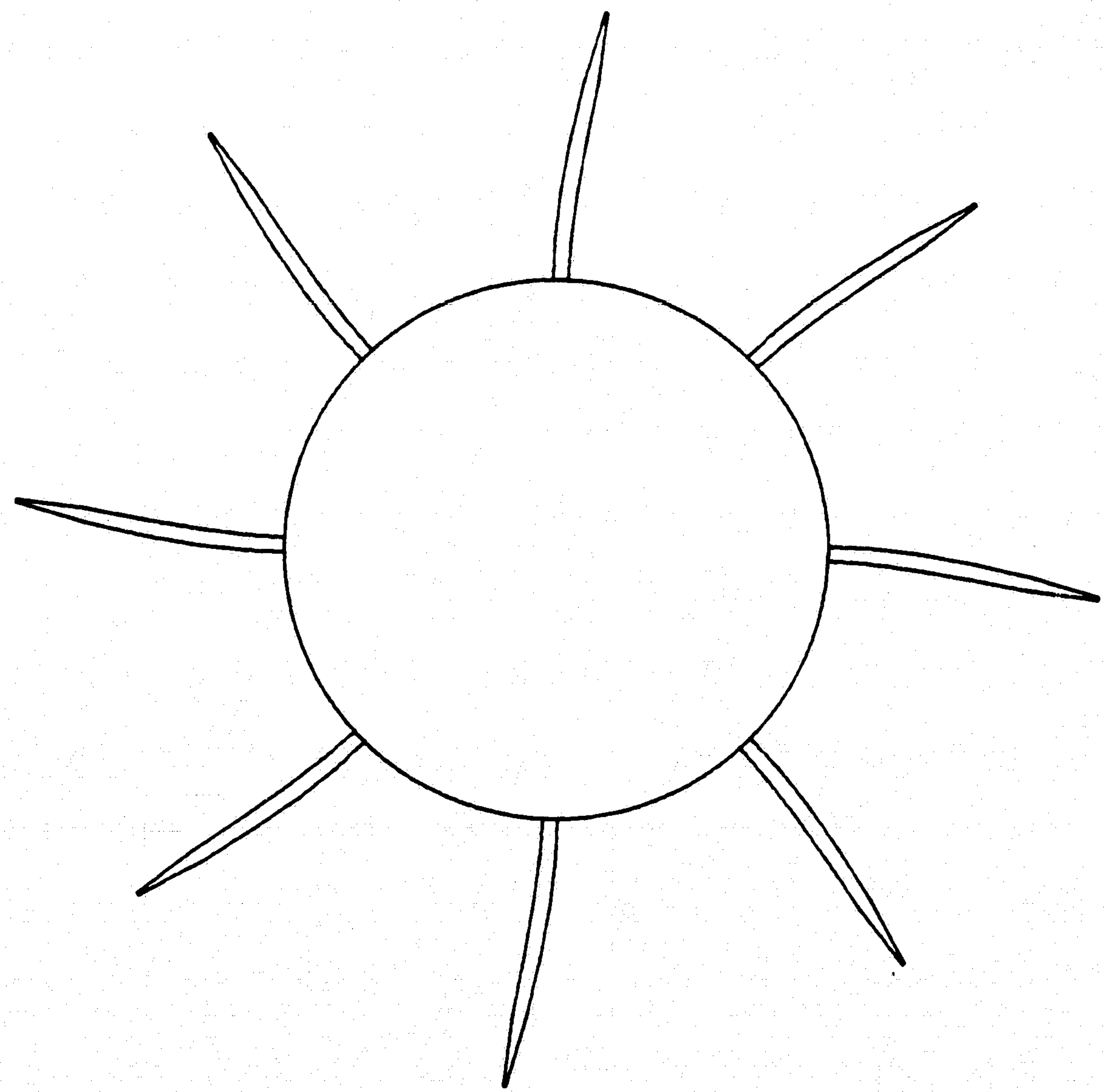


Figure 18

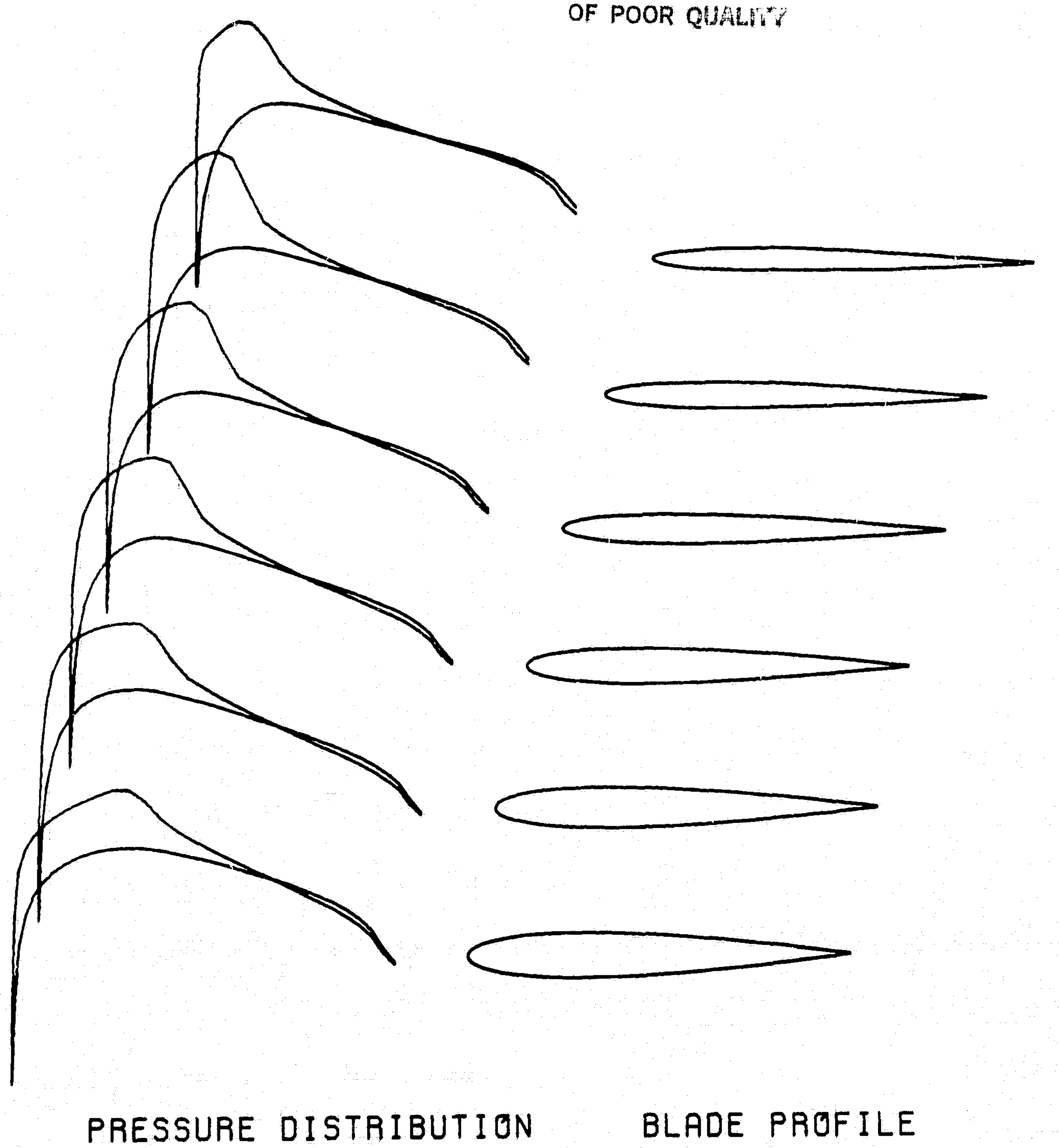

PRESSURE DISTRIBUTION
BLADE PROFILE
M1 = .75, M2 = .75, DEV = 0.0, ALP = 0.0

Figure 19

VI. HOW TO USE THE CODE

This section is intended to serve as a guide for users of our computer code. We explain the different code options and give a listing of the input parameters.

Our computer code includes two options which permit the solution of either the compressor problem or the propeller problem, and it treats them either in a realistic axial flow configuration or in a three-dimensional cascade representation. In the first option there is a choice of two boundary conditions. In the propeller case the reduced velocity potential is set to zero at infinity; in the compressor case the boundary condition is given by a Neumann condition at the cowling. The first option determines the first mapping to be performed. For the axial flow configuration the whole space is mapped into a part of the space consisting of as many periodic strips as there are blades around the hub. For the cascade configuration this first mapping is not required, since the computation starts directly from a periodic strip. In both cases the periodic strip is mapped conformally into a quadrilateral. For the compressor case the blade shape and the vortex sheet define the bottom of the quadrilateral; the hub and the cowling determine the sides; and the upper edge is defined by the flow at infinity. For the propeller case the side previously determined by the cowling is defined by the flow at infinity.

The shape of the blade is defined by giving the coordinates at different span sections from the hub to the cowling. As many as six sections can be defined by giving the Cartesian coordinates at each section with z constant. If two sections are similar, only the coordinates of the first section are read in.

We shall now describe the input data cards and explain the parameters which occur in the input. All the data cards are read in format (8F10.6). A title card is inserted before each data card or each series of data cards.

Title Card 1:

NX, NY, NZ, FPLOT, XSCAL, PSCAL, GRIDX, GRIDY.

NX, NY, NZ — are the number of mesh cells in each direction of the transformed space. Thus NZ is the number of mesh cells in the radial direction, NX the number of mesh cells along the blade and the vortex sheet, and NY the number of mesh cells in the third direction of the computational grid. If NX = 0, the program stops. The dimensions in the code are NX+1, NY+2, NZ+3.

FPLOT — is the parameter controlling the generation of plots. IF FPLOT = 0, a printer plot but no Calcomp plot is obtained at each span station. If FPLOT = 1, both a printer plot and a Calcomp plot are generated; and if FPLOT = 2, only a Calcomp plot is generated.

XSCAL, PSCAL — determine the scales of the Calcomp plot. The pressure scale is set to PSCAL per inch in each section plot. PSCAL = 0 is equivalent to PSCAL = 0.5. If PSCAL is positive, each section is scaled so that the span is 5. If PSCAL is negative, each section is scaled proportionately to the local chord and the maximum chord is XSCAL/2.

GRIDX, GRIDY — control the grid generation. The grid depends largely on the geometry of the compressor, so these two parameters must be chosen carefully so as to obtain an acceptable grid. GRIDX has to be greater than -1 and GRIDY is positive. In most of the cases we can use GRIDX = 0 and GRIDY = 1. Decreasing GRIDX gives more points near the nose and less at the trailing edge. Increasing GRIDX gives fewer points near the nose and more at the trailing edge. Increasing GRIDY increases the number of points near the blade and the vortex sheet.

Title Card 2:

MIT, COV, P1, P2, P3, BETA, FHALF, FCONT.

MIT — is the maximum number of iterations computed.

COV — is the desired accuracy. If the maximum correction is less than COV, the program goes to the next grid or terminates. The same will happen if the maximum correction is greater than 10.

P1,P2 — are the relaxation parameters for the subsonic and the supersonic points respectively P1 is between 1. and 2., P2 is less than or equal to 1.

P3 — is used for the boundary values. P3 is usually set to 1, but a smaller value improves the convergence for very cambered blades.

BETA — is the damping parameter controlling the amount of added ϕ_{st}. It is normally set between 0. and 0.5.

FHALF — indicates whether the mesh is to be refined. For FHALF = 0 the program stops after convergence or after the prescribed number of iterations have been completed. For FHALF = 1 the mesh will be refined and another input card must be read for the parameters of the refined mesh. The mesh can be refined only twice and the maximum grid is 128×24×14.

FCONT indicates how the computation starts. After each run the velocity potential is stored on tape 8. If FCONT = 0 the program begins by initializing the velocity potential. If FCONT = 1, we have a continuation run and the data stored on tape 8 from a previous run is used as input and is read in on tape 7.

Title Card 3:

FMACH, OM, AL, PA, ZONE, DC.

FMACH is the inlet Mach number.

OM is the speed of rotation. This means that the compressor is turning at the speed of OM revolutions per 60 units of time. The unit of time depends on the flow conditions. The unit of length is the length of the blade divided by CHORD and the unit of speed is the inlet speed.

AL is the inlet angle.

PA determines whether an axial flow configuration or a cascade is to be studied. PA = 0 for a cascade. PA = 1 for an axial flow configuration.

ZONE is the distance between the axis and the hub. In the cascade case it is set equal to 1.

DC is the gap-to-chord ratio for the cascade case. For an axial flow configuration it is the number of blades around the hub.

Title Card 4:

ZSYM, NC, SWEEP1, SWEEP2, SWEEP, DIHED1, DIHED2, DIHED

ZSYM indicates whether we have a compressor problem or a propeller problem. ZSYM = 0 for the compressor case and ZSYM = 1 for the propeller case. Thus, if ZSYM = 1, there is no cowling, the outlet speed is the same as the inlet speed and the radial direction is stretched to infinity.

NC is the number of span stations at which the blade section is defined. The blade section is interpolated linearly between two span stations. If NC < 3 the geometry is given by the last case studied and the program will run for this ZSYM and for the parameters given by data cards 1 through 3 without the need to read any more cards.

SWEEP1,SWEEP2,SWEEP are the angles of sweep of the singular line at the hub, at the cowling (or the tip) and at the far field respectively. For a compressor blade SWEEP is not used.

DIHED1,DIHED2,DIHED are the dihedral angles of the singular line, at the hub, at the cowling (or the tip) and at the far field respectively. In the compressor case DIHED is not used.

Title Card 5:

Z, XLE, YLE, CHORD, THICK, ALPHA, FSEC

Z	is the location of the span section.
XLE,YLE	are the coordinates of the leading edge.
CHORD	is the chord of this section, used to scale the profile.
THICK	multiplies each y coordinate; thus the thickness of the section is multiplied by THICK.
ALPHA	is the angle of twist through which the section is rotated. Changing the value of ALPHA by the same amount at each span station introduces a stagger angle.
FSEC	indicates whether section coordinates are to be read in from data cards. FSEC = 0 means no further cards are to be read for this section. FSEC = 1 means the coordinates of the section will be determined by the next input cards.

Title Card 6:

YSYM, NU, NL

YSYM	indicates whether the profile is symmetric or not. YSYM = 1 for a symmetric profile and YSYM = 0 otherwise. For YSYM = 1 only the coordinates of the upper surface are read.
NU,NL	are the number of points on the upper and lower surfaces respectively.

Title Card 7:

TRAIL, SLOPT, XSING, YSING

TRAIL — is the included trailing edge angle in degrees, or the angle between the upper and lower surfaces if the profile is open.

SLOPT — is the slope of the mean camber line at the trailing edge. SLOPT is used not only for the profile but also to determine the vortex sheet behind the profile. If SLOPT is too big and the gap-to-chord ratio is small the program may abort.

XSING,YSING — are the coordinates of the singular point inside the profile about which the profile is unwrapped. This point must be chosen in such a way that the mapped coordinates are as smooth as possible without a pronounced bump near the nose. If such a bump occurs, we first move the bump nearer the nose by positioning the singular point closer to the upper or lower surface. Then the bump is smoothed by placing the singular point at the right distance from the nose. Moreover, the singular point must be redefined not only for any new profile but also for any new configuration, since the geometry changes with the parameter DC.

Title Card 8:

X, Y

X,Y — are the coordinates of the upper surface. There are NU cards and the coordinates are read from the leading edge to the trailing edge.

Title Card 9:

X, Y

X,Y — are the coordinates of the lower surface. There are NL cards and the coordinates are read again from the leading edge to the trailing edge. The first data card repeats the first data card for the upper surface which defines the coordinates of the leading edge.

These cards are followed by (NC-1) series of cards beginning with Title Card 5.

The program provides both graphical and printed output. The Calcomp plots were shown in the previous section.

In the printed output we can read the listing of all the coordinates given in the input. Then the mapped surface coordinates are printed for the hub and the tip. This allows us to ascertain whether the coordinates of the singular points are well chosen. After the iteration history, with the maximum correction and the mean correction to the velocity potential and with the maximum residual and the mean residual for the difference equations at each iteration, the program gives us a printed plot of the coefficient of pressure on the surface. This is followed by a Mach number chart. These last two are given for each span station. Finally the characteristics of the blade are printed.

If the mesh has to be refined, a new series of output is printed for the new mesh. If not, the program terminates or restarts by reading a new data deck.

VII. BIBLIOGRAPHY

1. F. Bauer, P. Garabedian, D. Korn, Supercritical Wing Sections, Lecture Notes in Economics and Mathematical Systems, Vol. 66, Springer-Verlag, New York, 1972.
2. F. Bauer, P. Garabedian, A. Jameson, D. Korn, Supercritical Wing Sections II, Lecture Notes in Economics and Mathematical Systems, Vol. 108, Springer-Verlag, New York, 1975.
3. F. Bauer, P. Garabedian, D. Korn, Supercritical Wing Sections III, Lecture Notes in Economics and Mathematical Systems, Vol. 150, Springer-Verlag, New York, 1977.
4. F. Bauer, P. Garabedian, G. McFadden, The NYU Inverse Swept Wing Code, to be published.
5. J. Brochet, J. P. Veuillot, J. P. Boisseau, M. Enselme, Calcul de l'écoulement tridimensionnel dans une roue mobile de souffante, 18^{e} colloque d'aéreodynamique appliquée de l'aaaF, November 1981.
6. R. Courant, K. O. Friedrichs, Supersonic Flow and Shock Waves, Interscience-Wiley, New York, 1948.
7. C. Farrell, Quasi-Three-Dimensional Full Potential Transonic Blade-to-Blade Code, NASA Lewis Research Report, October 1981.
8. C. Farrell, P. Beauchamp, A System of Codes for Quasi-Three-Dimensional Shockless Blade Row Design, NASA Lewis Research Report, October 1981.
9. P. Garabedian, Estimation of the Relaxation Factor for Small Mesh Size, Math. Tables Aid. Comp., Vol. 10, pp. 183-185, 1956.
10. P. Garabedian, Partial Differential Equations, Wiley, New York, 1964.

11. P. Garabedian, G. McFadden, On the Design of Supercritical Swept Wings, AIAA Journal, Vol. 20, pp. 289-291, 1982.

12. H. Howard, A. Hansen, Three-Dimensional Compressor Flow Theory and Real flow Effects, in Aerodynamic Design of Axial Flow Compressors (NASA-SP-36), pp. 365-384.

13. D. Ives, J. Liutermoza, Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques, AIAA Journal, Vol. 15, pp. 647-652, 1977.

14. A. Jameson, Iterative Solutions of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1, CPAM, Vol., 27, pp. 283-309, 1974.

15. A. Jameson, D. Caughey, Numerical Calculation of the Flow Past a Swept Wing, ERDA Research and Development Report No. COO-3077-140, June 1977.

16. W. H. Jou, Finite Volume Calculation of Three-Dimensional Flow around a Propeller, Oral Communication.

17. E. McIntyre, Design of Transonic Cascades by Conformal Transformation of the Complex Characteristics, ERDA Research and Development Report, No. COO-3077-136, November 1976.

18. C. S. Morawetz, On the Nonexistence of Continuous Transonic Flow Past Profiles, CPAM, Vol. 9, pp. 45-68, 1956 and CPAM, Vol. 10, pp. 107-131, 1957.

19. Ph. Morice, H. Viviand, Equations de conservation et condition d'irréversibilité pour les écoulements transsoniques potentiels, Compte Rendus de l'Académie des Sciences, Paris, t 291, série B, pp. 235-238, 1980.

20. E. Murman, J. Cole, Calculation of Plane Steady Transonic Flows, AIAA Journal, Vol. 9, pp. 114-121, 1971.

21. M. H. Vavra, Aero-Thermodynamics and Flow in Turbomachines, Wiley, New York, 1960.

22. J P. Veuillot, Calcul de l'écoulement moyen dans une roue de turbomachine axiale, Publication ONERA No. 155, 1973.

23. J. P. Veuillot, Méthodes de calcul de l'écoulement tridimensionnel dans une roue de turbomachine, Revue francaise de Mécanique no. 55-56, pp. 19-29, 1975.

24. J. P. Veuillot, Calculation of the Quasi-Three-Dimensional Flow in a Turbomachine Blade Row, Engineering for Power, Vol. 99, No. 1, pp. 53-62, 1977.

ORIGINAL PAGE IS
OF POOR QUALITY

VIII. LISTING OF THE CODE

```
      PROGRAM CSCDF22(INPUT=512,OUTPUT=512,TAPE5=INPUT,TAPE6=OUTPUT,TAPE
     17=512,TAPE8=512)
C     MAIN ROUTINE WHICH CONTROLS THE COMPUTATIONAL PROCEDURE
C     G IS THE REDUCED VELOCITY POTENTIAL
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,AO(129),A1(129),A2(129),A3(129),BO(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      COMMON /CAL/ P1,P2,P3,BETA,FR,KM,DG,GM,NS,U1,V1,W1,IR,JR,KR,IG,JG,
     1KG
      DIMENSION XS(200,6), YS(200,6), ZS(6), XLE(6), YLE(6), SLOPT(6), T
     1RAIL(6), NP(6), E1(6), E2(6), E3(6), E4(6), E5(6), XP(200), YP(200
     2), D1(200), D2(200), D3(200), X(129), Y(129), SV(129), SM(129), CP
     3(129), CHORD(17), SCL(17), SCD(17), SCM(17), FIT(3), COVO(3), P10(
     43), P20(3), P30(3), BETAO(3), FHALF(3), RES(501), COUNT(501), DESC
     5(10), TITLE(10)
      ND=200
      NE=129
      IND=1
      U1=0.
      V1=0.
      W1=0.
      IREAD=5
      IWRIT=6
      KPLOT=0
      IPLOT=1
      JIT=0
      REWIND 7
      RAD=57.29578
   10 WRITE (IWRIT,430)
      WRITE (IWRIT,210)
      READ (IREAD,420) TITLE
      WRITE (IWRIT,460) TITLE
      READ (IREAD,420) DESC
      READ (IREAD,410) FNX,FNY,FNZ,FPLOT,XSCAL,PSCAL,F,TY
      IF (FNX.LT.1.) GO TO 200
      WRITE (IWRIT,520) FNX,FNY,FNZ,FPLOT,XSCAL,PSCAL,F,TY
      NX=FNX
      NY=FNY
      NZ=FNZ
      KPLOT=ABS(FPLOT)
      READ (IREAD,420) DESC
      WRITE (IWRIT,530)
      NM=0
   20 NM=NM+1
```

```
      READ (IREAD,410) FIT(NM),COVO(NM),P1O(NM),P2O(NM),P3O(NM),BETAO(NM
     1),FHALF(NM),FCONT
      WRITE (IWRIT,450) FIT(NM),COVO(NM),P1O(NM),P2O(NM),P3O(NM),BETAO(N
     1M),FHALF(NM)
      IF (FHALF(NM).NE.0..AND.NM.LT.3) GO TO 20
      FHALF(3)=0.
      READ (IREAD,420) DESC
      READ (IREAD,410) FMACH,OM,AL,PA,ZONE,DC
      WRITE (IWRIT,540) FMACH,OM,AL,PA,DC
      DC=(1.-PA)*DC*RAD/360.+PA/DC
      XK=1./DC
      ZONE=1.+PA*(ZONE-1.)
      OMEGA=PA*OM/RAD*6./ZONE
      ALPHA=AL/RAD
      CALL GEOM (ND,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,SWEEP1,SWEE
     1P2,SWEEP,DIHED1,DIHED2,DIHED,KSYM,XTEO,CHORDO,ZTIP,ISYMO,XK,PA,IND
     2)
      ISYM=ISYMO
      IF (ALPHA.NE.0.) ISYM=0
      CA=COS(ALPHA)
      SA=SIN(ALPHA)
      IF (FCONT.LT.1) GO TO 40
      READ (7) NX,NY,NZ,NM,K1,K2,JIT,U1,V1,W1
      MX=NX+1
      MY=NY+2
      MZ=NZ+3
      DO 30 K=1,MZ
      READ (7) ((G(I,J,K),I=1,MX),J=1,MY)
   30 CONTINUE
      READ (7) (EO(K),K=K1,K2)
   40 CONTINUE
      CALL COORD (NX,NY,NZ,XTEO,ZTIP,XMAX,ZMAX,KSYM,PA,SY,SCAL,SCALZ,AX,
     1AY,AZ,TY,F,A0,A1,A2,A3,B0,B1,B2,B3,Z,C1,C2,C3,C4,C5)
      CALL SINGL (NC,NZ,KTE1,KTE2,CHORDO,PA,SWEEP1,SWEEP2,SWEEP,DIHED1,D
     1IHED2,DIHED,ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,Z,E1,E2,E3,E4,E5,IND,ZO
     2NE)
      CALL SURF (ND,NE,NC,NX,NZ,ISYM,KTE1,KTE2,SCAL,A0,Z,ZS,XC,YC,SLOPT,
     1TRAIL,XS,YS,NP,ITE1,ITE2,IV,SO,XP,YP,D1,D2,D3,X,Y,IND,XK,PA,XZ,YZ,
     2A1,C1,KSYM)
      IF (IND.EQ.0) GO TO 190
      IF (FCONT.GE.1.) GO TO 50
      NM=1
      CALL ESTIM
   50 CONTINUE
      NIT=JIT
      T=2./SCAL
   60 WRITE (IWRIT,430)
      FCONT=0.
      MIT=FIT(NM)+JIT
      COV=COVO(NM)
      BETA=BETAO(NM)
      MX=NX+1
```

```
      MY=NY+2
      MZ=NZ+3
      KY=NY+1
      K1=3
      K2=NZ+2
      WRITE (IWRIT,220)
      DO 70 I=2,NX
      WRITE (IWRIT,480) (IV(I,K),K=K1,K2)
   70 CONTINUE
      WRITE (IWRIT,430)
      WRITE (IWRIT,230)
      DO 80 I=2,NX
      WRITE (IWRIT,440) AO(I),SO(I,K1),SO(I,KTE2)
   80 CONTINUE
      WRITE (IWRIT,240)
      WRITE (IWRIT,440) XMAX,AX
      WRITE (IWRIT,430)
      WRITE (IWRIT,250)
      DO 90 J=2,KY
      WRITE (IWRIT,440) BO(J)
   90 CONTINUE
      WRITE (IWRIT,260)
      WRITE (IWRIT,440) SY,AY
      WRITE (IWRIT,430)
      WRITE (IWRIT,270)
      DO 100 K=K1,K2
      WRITE (IWRIT,440)
      WRITE (IWRIT,440) Z(K),XC(K),YC(K),XZ(K),YZ(K),XZZ(K),YZZ(K)
  100 CONTINUE
      WRITE (IWRIT,280)
      WRITE (IWRIT,440) ZMAX,AZ
      WRITE (IWRIT,430)
      WRITE (IWRIT,290)
      WRITE (IWRIT,300)
      WRITE (IWRIT,470) NX,NY,NZ
      CALL SECOND (TIME)
      WRITE (IWRIT,510) TIME
      WRITE (IWRIT,310)
      WRITE (IWRIT,440) FMACH,OM,AL
      WRITE (IWRIT,320)
      LX=NX/2+1
      CL=0.
      DO 110 K=K1,K2
      I1=ITE1(K)
      X(I1)=1.+.5*SCAL*(AO(I1)*AO(I1)-SO(I1,K)*SO(I1,K))
      Y1=SCAL*AO(I1)*SO(I1,K)
      X(I1)=XC(K)+ALOG(X(I1)**2+Y1**2)/XK/2.
      X(LX)=1.+.5*SCAL*(AO(LX)*AO(LX)-SO(LX,K)*SO(LX,K))
      Y1=SCAL*AO(LX)*SO(LX,K)
      X(LX)=XC(K)+ALOG(X(LX)**2+Y1**2)/XK/2.
      CHORD(K)=X(I1)-X(LX)
  110 CONTINUE
```

```
      KZDUM=KTE2-1
      S=0.
      DO 120 K=KTE1,KZDUM
      DZO=.5*(Z(K+1)-Z(K))
      S=S+DZO*(CHORD(K+1)+CHORD(K))/C5(K)
  120 CONTINUE
      ABLADE=S
  130 NIT=NIT+1
      P1=P10(NM)
      P2=P20(NM)
      P3=P30(NM)
      CALL BOUND
      U2=U1+CA
      V2=V1+SA
      WRITE (IWRIT,490) NIT,DG,IG,JG,KG,FR,IR,JR,KR,GM,RM,P1,U2,V2,NS
      COUNT(NIT)=NIT-1-JIT
      RES(NIT)=FR
      IF (NIT.LT.MIT.AND.ABS(DG).GT.COV.AND.ABS(DG).LT.10.) GO TO 130
      RATE=0.
      IF (NIT.GT.JIT+1) RATE=(ABS(RES(NIT)/RES(JIT+1)))**(1./(COUNT(NIT)
     1-COUNT(JIT+1)))
      WRITE (IWRIT,330)
      WRITE (IWRIT,500) RES(JIT+1),RES(NIT),COUNT(NIT),RATE
      CALL SECOND (TIME)
      WRITE (IWRIT,510) TIME
      LX=NX/2+1
      DO 150 K=K1,MZ
      IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 150
      I1=ITE1(K)
      I2=ITE2(K)
      CALL VELO (K,K,SV,SM,CP,X,Y)
      CHORD(K)=X(I1)-X(LX)
      CALL FORCF (I1,I2,X,Y,CP,AL,CHORD(K),XC(K),SCL(K),SCD(K),SCM(K),A2
     1,A3,C5(K),PA)
      IF (KPLOT.GT.1.AND.K.GT.KTE1) GO TO 140
      WRITE (IWRIT,430)
      WRITE (IWRIT,340)
      WRITE (IWRIT,440) FMACH,OM,AL
      WRITE (IWRIT,350)
  140 WRITE (IWRIT,440) Z(K),SCL(K),SCD(K),SCM(K),CHORD(K)
      IF (KPLOT.LE.1.) CALL CPLOT (I1,I2,SM,A0,SO(1,K),FMACH)
      CALL SPEED (K)
  150 CONTINUE
      CALL TOTFOR (KTE1,KTE2,CHORD,SCL,SCD,SCM,Z,XC,C5,CL,CD,CMP,CMR,CMY
     1,PA,ABLADE)
      VLD=0.
      IF (ABS(CD).GT.1.E-6) VLD=CL/CD
      WRITE (IWRIT,430)
      WRITE (IWRIT,360)
      IF (KSYM.EQ.1) U2=CA
      IF (KSYM.EQ.1) V2=SA
      V=ATAN2(V2,U2)
```

```
      V=V*RAD-AL
      Q=U2*U2+V2*V2+W1*W1
      Q1=1./FMACH**2+0.2
      U=SQRT(Q/(Q1-0.2*Q))
      WRITE (IWRIT,440) FMACH,DM,AL,U,V
      WRITE (IWRIT,370)
      WRITE (IWRIT,440) CL,CD,VLD
      WRITE (IWRIT,380)
      WRITE (IWRIT,440) CMP,CMR,CMY,ABLADE
      LPLOT=FPLOT*(1.-FHALF(NM))
      IF (LPLOT.LT.1) GO TO 160
      DC=360./XK/RAD*ZONE/CHORDO
      CALL THREED (IPLOT,SV,SM,CP,X,Y,TITLE,DC,AL,ZONE,U,V,CHORDO,XSCAL,
     1PSCAL)
  160 CONTINUE
      IF (FHALF(NM).EQ.0.) GO TO 170
      NX=NX+NX
      NY=NY+NY
      NZ=NZ+NZ
      CALL COORD (NX,NY,NZ,XTEO,ZTIP,XMAX,ZMAX,KSYM,PA,SY,SCAL,SCALZ,AX,
     1AY,AZ,TY,F,A0,A1,A2,A3,B0,B1,B2,B3,Z,C1,C2,C3,C4,C5)
      CALL SINGL (NC,NZ,KTE1,KTE2,CHORDO,PA,SWEEP1,SWEEP2,SWEEP,DIHED1,D
     1IHED2,DIHED,ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,Z,E1,E2,E3,E4,E5,IND,ZO
     2NE)
      CALL SURF (ND,NE,NC,NX,NZ,ISYM,KTE1,KTE2,SCAL,A0,Z,ZS,XC,YC,SLOPT,
     1TRAIL,XS,YS,NP,ITE1,ITE2,IV,SO,XP,YP,D1,D2,D3,X,Y,IND,XK,PA,XZ,YZ,
     2A1,C1,KSYM)
      IF (IND.EQ.0) GO TO 190
      CALL REFIN
      NM=NM+1
      NIT=0
      GO TO 60
  170 CONTINUE
      WRITE (8) NX,NY,NZ,NM,K1,K2,NIT,U1,V1,W1
      WRITE (6,390)
      DO 180 K=1,MZ
      WRITE (8) ((G(I,J,K),I=1,MX),J=1,MY)
  180 CONTINUE
      WRITE (8) (EO(K),K=K1,K2)
      END FILE 8
      REWIND 8
      GO TO 10
  190 WRITE (IWRIT,430)
      WRITE (IWRIT,400)
      STOP 0102
  200 CONTINUE
      CALL PLOT (0.,0.,999)
      STOP 0101
C
  210 FORMAT (20HONYU COMPRESSOR CODE)
  220 FORMAT (49HOINDICATION OF LOCATION OF BLADE AND VORTEX SHEET,27H I
     1N COORDINATE PLANE Y = 0./27HO((IV(I,K),K=K1,K2),I=2,NX))
```

```
230 FORMAT (49HOCHORDWISE CELL DISTRIBUTION IN TRANSFORMED PLANE,46H A
   1ND MAPPED SURFACE COORDINATES AT HUB AND TIP/15HO       X        ,15
   2H    HUB PROFILE,15H   TIP PROFILE )
240 FORMAT (15HO  TE LOCATION ,15H    POWER LAW   )
250 FORMAT (46HONORMAL CELL DISTRIBUTION IN TRANSFORMED PLANE/15HO
   1   Y        )
260 FORMAT (15HO  SCALE FACTOR,15H    POWER LAW   )
270 FORMAT (45HOSPANWISE CELL DISTRIBUTION AND SINGULAR LINE/15HO
   1  Z        ,15H       X SING   ,15H       Y SING   ,15H          XZ
   2,15H          YZ     ,15H        XZZ     ,15H          YZZ     )
280 FORMAT (15HO  TIP LOCATION,15H    POWER LAW   )
290 FORMAT (19HOITERATIVE SOLUTION)
300 FORMAT (15HO         NX      ,15H          NY      ,15H          NZ     )
310 FORMAT (15HO      MACH NO    ,15H       OMEGA     ,15H  ANG OF ATTACK)
320 FORMAT (10HOITERATION,15H    CORRECTION ,4H   I ,4H  J ,4H  K ,15H
   1     RESIDUAL  ,4H  I ,4H  J ,4H  K ,10H MEAN COR.,10H MEAN RES.,10
   2H REL FCT 1,10H   XSPEED ,10H  YSPEED  ,10H SONIC PTS)
330 FORMAT (15HO MAX RESIDAL 1,15H  MAX RESIDAL 2,15H         WORK     ,1
   15H  REDUCTN/CYCLE)
340 FORMAT (24HOSECTION CHARACTERISTICS/15HO     MACH NO    ,15H       OM
   1EGA    ,15H  ANG OF ATTACK)
350 FORMAT (/13H SPAN STATION,12X2HCL,13X2HCD,13X2HCM,10X5HCHORD)
360 FORMAT (22HOBLADE CHARACTERISTICS/15HO     MACH NO 1 ,15H       OMEG
   1A    ,15H  ANG OF ATTACK,15H      MACH NO 2 ,15H  DEV. ANGLE    )
370 FORMAT (15HO          CL      ,15H        CD        ,15H          L/D       )
380 FORMAT (/2X8HCM PITCH,6X7HCM ROLL,9X6HCM YAW,9X6HABLADE)
390 FORMAT (1X,14HWRITE ON TAPE8)
400 FORMAT (24HOBAD DATA,SPLINE FAILURE)
410 FORMAT (8F10.6)
420 FORMAT (10A8)
430 FORMAT (1H1)
440 FORMAT (F12.5,7F15.5)
450 FORMAT (1X,8E15.5)
460 FORMAT (1HO,10A8)
470 FORMAT (I8,7I15)
480 FORMAT (1X,32I4)
490 FORMAT (I10,E15.5,3I4,E15.5,3I4,2E10.3,3F10.5,I10)
500 FORMAT (2E15.4,2F15.4)
510 FORMAT (15HOCOMPUTING TIME,F10.3,10H   SECONDS)
520 FORMAT (/5X3HFNX,11X3HFNY,11X3HFNZ,10X5HFPLOT,9X5HXSCAL,9X5HPSCAL,
   19X5HGRIDX,9X5HGRIDY/1X,8E14.5)
530 FORMAT (/4X7HFIT(NM),8X8HCOVO(NM),8X7HP10(NM),8X7HP20(NM),8X7HP30(
   1NM),6X9HBETAO(NM),6X9HFHALF(NM))
540 FORMAT (/5X5HFMACH,11X2HOM,14X2HAL,12X3H PA,12X4H DC /1X,5E15.5)
    END
```

```
      SUBROUTINE GEOM (ND,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,SWEEP
     11,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,KSYM,XTEO,CHORDO,ZTIP,ISYMO,XK,
     2PA,IND)
C      GEOMETRIC DEFINITION OF BLADE
      DIMENSION XS(ND,1), YS(ND,1), ZS(1), XLE(1), YLE(1), SLOPT(1), ZT(
     16), TRAIL(1), XP(1), YP(1), NP(1), X(6), Y(6), Z(6), XC(6), YC(6),
     2 C(6)
      DIMENSION DESC(10), A(6), B(6), D(6), E(6)
      IREAD=5
      IWRIT=6
      RAD=57.29578
      READ (IREAD,230) DESC
      READ (IREAD,220) ZSYM,FNC,SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
      WRITE (IWRIT,270) ZSYM,FNC,SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
      KSYM=ZSYM
      IF (FNC.LT.3.) RETURN
      NC=FNC
      SWEEP1=SWEEP1/RAD
      SWEEP2=SWEEP2/RAD
      SWEEP=SWEEP/RAD
      DIHED1=DIHED1/RAD
      DIHED2=DIHED2/RAD
      DIHED=DIHED/RAD
      S1=TAN(SWEEP1)
      S2=TAN(SWEEP2)
      T1=TAN(DIHED1)
      T2=TAN(DIHED2)
      ISYMO=1
      XTEO=0.
      CHORDO=0.
      ZONE=1.
      K=1
   10 READ (IREAD,230) DESC
      READ (IREAD,220) ZT(K),XL,YL,CHORD,THICK,AL,FSEC
      WRITE (IWRIT,280) ZT(K),XL,YL,CHORD,THICK,AL,FSEC
      ALPHA=AL/RAD
      ZT(K)=ZT(K)/ZONE
      ZS(K)=ALOG(ZT(K)+(1.-PA))*PA+(1.-PA)*ZT(K)
      IF (FSEC.EQ.0.) GO TO 80
      IF (K.GT.1) GO TO 20
      ZONE=(1.-PA)+PA*ZT(1)
      ZS(1)=(1.-PA)*ZT(1)
      ZT(1)=1.
   20 CONTINUE
      READ (IREAD,230) DESC
      READ (IREAD,220) YSYM,FNU,FNL
      WRITE (IWRIT,290) YSYM,FNU,FNL,ZONE
      NU=FNU
      NL=FNL
      IF (YSYM.EQ.1.) NL=NU
      N=NU+NL-1
      READ (IREAD,230) DESC
```

```
      READ (IREAD,220) TRL,SLT,XSING,YSING
      WRITE (IWRIT,300) TRL,SLT,XSING,YSING
      READ (IREAD,230) DESC
      WRITE (IWRIT,310)
      DO 30 I=NL,N
      READ (IREAD,220) XP(I),YP(I)
      WRITE (IWRIT,260) XP(I),YP(I)
   30 CONTINUE
      L=NL+1
      IF (YSYM.GT.0.) GO TO 50
      READ (IREAD,230) DESC
      WRITE (IWRIT,320)
      DO 40 I=1,NL
      READ (IREAD,220) VAL,DUM
      WRITE (IWRIT,260) VAL,DUM
      J=L-I
      XP(J)=VAL
      YP(J)=DUM
   40 CONTINUE
      GO TO 70
   50 J=L
      DO 60 I=NL,N
      J=J-1
      XP(J)=XP(I)
   60 YP(J)=-YP(I)
   70 CONTINUE
      WRITE (IWRIT,200) ZS(K)
      WRITE (IWRIT,250) TRL,SLT,XSING,YSING
      WRITE (IWRIT,240)
   80 CONTINUE
      SCALE=CHORD/(XP(1)-XP(NL))/ZONE
      C(K)=EXP(-PA*ZS(K))
      XLE(K)=XL/ZONE+(XSING-XP(NL))*C(K)*THICK*SCALE
      YLE(K)=YL/ZONE+(YSING-YP(NL))*C(K)*THICK*SCALE
      XX=XP(NL)+(XSING-XP(NL))*C(K)*THICK
      YY=YP(NL)+(YSING-YP(NL))*C(K)*THICK
      CA=COS(ALPHA)
      SA=SIN(ALPHA)
      XC(K)=SCALE*(XX*CA+YY*THICK*SA)
      YC(K)=SCALE*(YY*THICK*CA-XX*SA)
      DO 90 I=1,N
      XS(I,K)=SCALE*(XP(I)*CA+THICK*YP(I)*SA)
      YS(I,K)=SCALE*(THICK*YP(I)*CA-XP(I)*SA)
   90 CONTINUE
      SLOPT(K)=TAN(ATAN(THICK*SLT)-ALPHA)*C(K)
      TRAIL(K)=ATAN(THICK*TAN(TRL/RAD)*C(K))
      NP(K)=N
      CHORDO=AMAX1(CHORDO,CHORD)
      IF (YSYM.LE.0..OR.ALPHA.NE.0.) ISYMO=0
      WRITE (IWRIT,210) ZS(K)
      WRITE (IWRIT,250) XL,YL,CHORD,THICK,AL
      K=K+1
```

```
      IF (K.LE.NC) GO TO 10
      XK=XK*((1.-PA)/CHORDO+PA)
      IF (PA.NE.1.0) GO TO 150
      DO 120 I=1,N
      DO 100 K=1,NC
      X(K)=XS(I,K)
      Y(K)=ATAN2(YS(I,K),ZT(K))
      Z(K)=ALOG(YS(I,K)**2+ZT(K)**2)/2.
  100 CONTINUE
      E1=1./(COS(Y(1))+T1*SIN(Y(1)))
      E2=1./(COS(Y(NC))+T2*SIN(Y(NC)))
      F1=(COS(Y(1))*T1-SIN(Y(1)))*E1
      F2=(COS(Y(NC))*T2-SIN(Y(NC)))*E2
      E1=S1*E1
      E2=S2*E2*ZT(NC)
      CALL SPLIF (1,NC,Z,X,A,B,C,1,E1,1,E2,0,0.,IND)
      CALL INTPL (1,NC,ZS,D,1,NC,Z,X,A,B,C,0)
      CALL SPLIF (1,NC,Z,Y,A,B,C,1,F1,1,F2,0,0.,IND)
      CALL INTPL (1,NC,ZS,E,1,NC,Z,Y,A,B,C,0)
      DO 110 K=1,NC
      XS(I,K)=D(K)
      YS(I,K)=E(K)
  110 CONTINUE
  120 CONTINUE
      DO 130 K=1,NC
      X(K)=XC(K)
      Y(K)=ATAN2(YC(K),ZT(K))
      Z(K)=ALOG(YC(K)**2+ZT(K)**2)/2.
  130 CONTINUE
      EA=COS(Y(1))
      EB=SIN(Y(1))
      EC=COS(Y(NC))
      ED=SIN(Y(NC))
      E1=S1/(EA+T1*EB)
      E2=S2*ZT(NC)/(EC+T2*ED)
      F1=(EA*T1-EB)/(EA+T1*EB)
      F2=(EC*T2-ED)/(EC+T2*ED)
      CALL SPLIF (1,NC,Z,X,A,B,D,1,E1,1,E2,0,0.,IND)
      CALL INTPL (1,NC,ZS,XC,1,NC,Z,X,A,B,D,0)
      CALL SPLIF (1,NC,Z,Y,A,B,D,1,F1,1,F2,0,0.,IND)
      CALL INTPL (1,NC,ZS,YC,1,NC,Z,Y,A,B,D,0)
      DO 140 K=1,NC
      X(K)=XLE(K)
      Y(K)=ATAN2(YLE(K),ZT(K))
      Z(K)=ALOG(YLE(K)**2+ZT(K)**2)/2.
  140 CONTINUE
      EA=COS(Y(1))
      EB=SIN(Y(1))
      EC=COS(Y(NC))
      ED=SIN(Y(NC))
      E1=S1/(EA+EB*T1)
      E2=S2*ZT(NC)/(EC+ED*T2)
```

```
      F1=(EA*T1-EB)/(EA+EB*T1)
      F2=(EC*T2-ED)/(EC+ED*T2)
      CALL SPLIF (1,NC,Z,X,A,B,D,1,E1,1,E2,0,0.,IND)
      CALL INTPL (1,NC,ZS,XLE,1,NC,Z,X,A,B,D,0)
      CALL SPLIF (1,NC,Z,Y,A,B,D,1,F1,1,F2,0,0.,IND)
      CALL INTPL (1,NC,ZS,YLE,1,NC,Z,Y,A,B,D,0)
      GO TO 170
  150 CONTINUE
      Z0=ZS(1)
      DO 160 K=1,NC
      ZS(K)=ZS(K)-Z0
  160 CONTINUE
  170 CONTINUE
      DO 190 K=1,NC
      DO 180 I=1,N
      XS(I,K)=XS(I,K)-XC(K)
      YS(I,K)=YS(I,K)-YC(K)
  180 CONTINUE
      XXS=EXP(XK*XS(1,K))*COS(XK*YS(1,K))-1.
      XX=SQRT(EXP(2.*XK*XS(1,K))-2.*XXS-1.)
      XXS=(XX+XXS)/2.
      XTEO=AMAX1(XTEO,XXS)
  190 CONTINUE
      ZTIP=ZS(NC)
      RETURN
C
  200 FORMAT (16HOPROFILE AT Z = ,F10.5/15HO     TE ANGLE   ,15H      TE SL
     1OPE   ,15H       X SING   ,15H       Y SING    )
  210 FORMAT (27HOSECTION DEFINITION AT Z = ,F10.5/15HO        XLE      ,15
     1H        YLE      ,15H      CHORD     ,15HTHICKNESS RATIO,15H       AL
     2PHA      )
  220 FORMAT (8F10.6)
  230 FORMAT (10A8)
  240 FORMAT (1H1)
  250 FORMAT (F12.4,7F15.4)
  260 FORMAT (8E15.5)
  270 FORMAT (/5X4HZSYM,12X3HFNC,10X6HSWEEP1,9X6HSWEEP2,9X5HSWEEP,10X5HD
     1IHED1,9X6HDIHED2,10X5HDIHED/1X,8E15.5)
  280 FORMAT (/5X5HZS(K),12X2HXL,13X2HYL,11X5HCHORD,10X5HTHICK,12X2HAL,1
     12X4HFSEC/1X,7E15.5)
  290 FORMAT (/6X4HYSYM,11X3HFNU,12X3HFNL/1X,4E15.5)
  300 FORMAT (/6X3HTRL,12X3HSLT,11X5HXSING,10X5HYSING/1X,4E15.5)
  310 FORMAT (/5X5HXP(I),10X5HYP(I))
  320 FORMAT (/6X3HVAL,12X3HDUM)
      END
```

```
      SUBROUTINE COORD (NX,NY,NZ,XTEO,ZTIP,XMAX,ZMAX,KSYM,MA,SY,SCAL,SCA
     1LZ,AX,AY,AZ,TY,F,A0,A1,A2,A3,B0,B1,B2,B3,Z,C1,C2,C3,C4,C5)
C     SETS UP STRETCHED AND SPANWISE COORDINATES
      DIMENSION A0(1), A1(1), A2(1), A3(1), B0(1), B1(1), B2(1), B3(1),
     1Z(1), C1(1), C2(1), C3(1), C4(1), C5(1)
      DX=2./NX
      DY=1./NY
      KY=NY+1
      Z0=0.0
      K2=NZ+2
      DZ=1./NZ
      K1=3
      AB=1.
      AX=.5
      AY=1.
      AZ=.5
      BZ=0.
      XMAX=.75
      IF (KSYM.EQ.1.) XMAX=0.625
      S73=SQRT(73.)
      ZMAX=.625
      IF (KSYM.EQ.0) ZMAX=1.-1./NZ
      SCAL=XTEO/(.50001*XMAX*XMAX)
      SCALZ=ZTIP/(1.000001*ZMAX)
      T=2./SCAL
      SY=SQRT(T)/TY
      BX=AMIN1(0.,250.*(T-0.07)*ABS(T-0.07))
      BX=(F*(S73-11.)/2.+BX)/(F+1.)
      V2=(DX/DY)**2
      W1=1./SCALZ
      W2=(W1*DX/DZ)**2
      BBX=-BX*SQRT(3.*(7.+S73))/((1.+S73)*XMAX**3)
      ABX=1.-BBX*SQRT((7.+S73)/12.)*XMAX**3
      CBX=(19.+S73)*XMAX*XMAX/12.
      ABBX=ABX+BBX*(3.*CBX-4.*XMAX*XMAX)*XMAX*XMAX/SQRT(CBX-XMAX*XMAX)
      MX=NX+1
      LX=NX/2+1
      DO 30 I=1,MX
      DD=(I-LX)*DX
      IF (I.EQ.1) DD=-1.+1./NX
      IF (I.EQ.MX) DD=1.-1./NX
      B=1.
      IF (ABS(DD).GT.XMAX) GO TO 10
      A=CBX-DD*DD
      AS=SQRT(A)
      C=ABX*AS+BBX*(3.*CBX-4.*DD*DD)*DD*DD
      D0=ABX*DD+BBX*AS*DD**3
      D1=AS/C
      D2=BBX*(CBX*(-6.*CBX+19.*DD*DD)-12.*DD**4)*DD/(A*C)
      GO TO 20
   10 IF (DD.LT.0.) B=-1.
      A=AB-((DD-B*XMAX)/(1.-XMAX))**2
```

```
      C=A**AY
      D=(AY+AY-1.)*(A[illegible]-A)
      [illegible]
      D1=A*C/((A[illegible]+D)*A[illegible])
      [illegible]
   20 A0(I)=.0
      A1(I)=.5[illegible]
      A2(I)= 1*.1
   30 A3(I)=.[illegible]
      DO 40 J=2,NY
      DD=(NY-J)*DY
      A=AH-DD*D1
      C=A**AY
      D=(AY+AY-1.)*(AH-A)
      D1=A*C/((AH+D)*DY)
      W1(J)=DY*DD/C
      E1(J)=.5*D1/DY
      E2(J)=D1*D1*W2
   40 E3(J)=-AY*DD*DY*(3.+A[illegible]+D)/((A3+D)*A)
      BBZ=-3.*SQRT(3.*(7.+S73))/((1.+S73)*ZMAX**3)
      ABZ=1.-3./8.*SQRT((7.+S73)/12.)*ZMAX**2
      CBZ=(1[illegible].+S73)*ZMAX*ZMAX/12.
      ABBZ=ABZ+BBZ*(3.*CBZ-4.*ZMAX*ZMAX)*ZMAX*ZMAX/SQRT(CBZ-ZMAX*ZMAX)
      K2=K2+1
      DO 70 K=2,K2
      DD=(K-K1)*DZ-Z0
      IF (K.EQ.K2) DD=DD-2.*DZ
      B=1.
      IF (ABS(DD).GT.ZMAX) GO TO 50
      A=CBZ-DD*DD
      AS=SQRT(A)
      C=ABZ*AS+BBZ*(3.*CBZ-4.*DD*DD)*DD*DD
      DD=ABZ*DD+BBZ*AS*DD**3
      D1=AS/C
      D2=BBZ*(CBZ*(-6.*CBZ+17.*DD*DD)-12.*DD**4)*DD/(A*C)
      GO TO 60
   50 IF (DD.LT.0.) B=-1.
      A=1.-((DD-B*ZMAX)/(1.-ZMAX))**2
      C=A**AZ
      D=(AZ+AZ-1.)*(1.-A)
      DD=B*ZMAX+ABBZ*(DD-B*ZMAX)/C
      D1=A*C/((1.+D)*ABBZ)
      D2=-(AZ+AZ)*(DD-B*ZMAX)*(3.+D)/((1.+D)*A*(1.-ZMAX)**2)
   60 Z(K)=SCALZ*DD
      C1(K)=.5*D1*W1/DZ
      C2(K)=D1*D1*W2
      C3(K)=.5*D2*D2
      C5(K)=EXP(-PA*Z(K))
   70 C4(K)=C5(K)*C5(K)
      Z(K2+1)=Z(K2)+Z(K2)-Z(K2-1)
      RETURN
      END
```

```
      SUBROUTINE SINGL (NC,NZ,KTE1,KTE2,CHORDO,PA,SWEEP1,SWEEP2,SWEEP,DI
     1HED1,DIHED2,DIHED,ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,Z,E1,E2,E3,E4,E5,
     2IND,ZONE)
C     GENERATES SINGULAR LINE FOR TRANSFORMATION
      DIMENSION ZS(1), XLE(1), YLE(1), XC(1), XZ(1), XZZ(1), YC(1), YZ(
     11), YZZ(1), Z(1), E1(1), E2(1), E3(1), E4(1), E5(1)
      DO 10 K=1,NC
      E4(K)=0.
      E5(K)=0.
   10 CONTINUE
      K1=3
      K2=NZ+2
      KTE1=K1
      DO 20 K=K1,K2
      IF (Z(K).LT.ZS(1)) KTE1=K+1
      IF (Z(K).LE.ZS(NC)) KTE2=K
   20 CONTINUE
      B=CHORDO/ZONE
      S1=TAN(SWEEP1)
      S2=TAN(SWEEP2)
      T1=TAN(DIHED1)
      T2=TAN(DIHED2)
      IF (PA.NE.1.0) GO TO 30
      F1=1./(COS(YLE(1))+T1*SIN(YLE(1)))
      F2=1./(COS(YLE(NC))+T2*SIN(YLE(NC)))
      T1=(COS(YLE(1))*T1-SIN(YLE(1)))*F1
      T2=(COS(YLE(NC))*T2-SIN(YLE(NC)))*F2
      S1=S1*F1
      S2=S2*F2*EXP(ZS(NC))
   30 CONTINUE
      CALL SPLIF (1,NC,ZS,XLE,E1,E2,E3,1,S1,1,S2,0,0.,IND)
      CALL INTPL (KTE1,KTE2,Z,XC,1,NC,ZS,XLE,E1,E2,E3,0)
      CALL INTPL (KTE1,KTE2,Z,XZ,1,NC,ZS,E1,E2,E3,E4,0)
      CALL INTPL (KTE1,KTE2,Z,XZZ,1,NC,ZS,E2,E3,E4,E5,0)
      CALL SPLIF (1,NC,ZS,YLE,E1,E2,E3,1,T1,1,T2,0,0.,IND)
      CALL INTPL (KTE1,KTE2,Z,YC,1,NC,ZS,YLE,E1,E2,E3,0)
      CALL INTPL (KTE1,KTE2,Z,YZ,1,NC,ZS,E1,E2,E3,E4,0)
      CALL INTPL (KTE1,KTE2,Z,YZZ,1,NC,ZS,E2,E3,E4,E5,0)
      S=B*TAN(SWEEP)
      S1=B*S1
      S2=B*S2
      T=B*TAN(DIHED)
      T1=B*T1
      T2=B*T2
      XC(2)=3.*(XC(3)-XC(4))+XC(5)
      YC(2)=3.*(YC(3)-YC(4))+YC(5)
      N=KTE2+1
```

```
      IF (N.GT.K2) GO TO 50
      DO 40 K=N,K2
      ZZ=(Z(K)-Z(KTE2))/B
      A=EXP(-ZZ)
      XC(K)=XC(KTE2)+S*ZZ+(S2-S)*(1.-A)
      YC(K)=YC(KTE2)+T*ZZ+(T2-T)*(1.-A)
      XZ(K)=(S+(S2-S)*A)/B
      YZ(K)=(T+(T2-T)*A)/B
      XZZ(K)=-(S2-S)*A/(B*B)
      YZZ(K)=-(T2-T)*A/(B*B)
   40 CONTINUE
   50 CONTINUE
      YC(K1-1)=YC(K1+1)
      YC(K2+1)=YC(K2-1)
      RETURN
      END


      SUBROUTINE SURF (ND,NE,NC,NX,NZ,ISYM,KTE1,KTE2,SCAL,AO,Z,ZS,XC,YC,
     1SLOPT,TRAIL,XS,YS,NP,ITE1,ITE2,IV,SO,XP,YP,D1,D2,D3,X,Y,IND,XK,PA,
     2XZ,YZ,A1,C1,KSYM)
C     INTERPOLATES MAPPED BLADE SURFACE AT MESH POINTS
C     INTERPOLATION IS LINEAR FOR CYLINDRICAL COORDINATES
      DIMENSION SO(NE,1), XS(ND,1), YS(ND,1), ZS(1), SLOPT(1), TRAIL(1),
     1 XC(1), YC(1), AO(1), Z(1), X(1), Y(1), XP(1), YP(1), D1(1), D2(1)
     2, D3(1), IV(NE,1), NP(1), ITE1(1), ITE2(1), XZ(1), YZ(1), A1(1), C
     31(1)
      PI=3.14159265
      S1=.5*SCAL
      T=1./S1
      DX=2./NX
      LX=NX/2+1
      MX=NX+1
      MZ=NZ+3
      IV0=1-ISYM-ISYM-ISYM
      IV1=-1-ISYM
      DO 10 K=1,MZ
      ITE1(K)=MX
      ITE2(K)=MX
      DO 10 I=1,MX
      IV(I,K)=-2
      SO(I,K)=0.
   10 CONTINUE
      K=KTE1
      K2=1
   20 K2=K2+1
```

```
      K1=K2-1
      R2=1.
      IF (ZS(K2)-Z(K)) 20,40,30
   30 R2=(EXP(Z(K))-EXP(ZS(K1)))/(EXP(ZS(K2))-EXP(ZS(K1)))
      R2=PA*R2+(1.-PA)*(Z(K)-ZS(K1))/(ZS(K2)-ZS(K1))
   40 R1=1.-R2
      C=R1*XS(1,K1)+R2*XS(1,K2)
      D=R1*YS(1,K1)+R2*YS(1,K2)
      CX=EXP(XK*C)*COS(XK*D)-1.
      C=SQRT(EXP(2.*XK*C)-2.*CX-1.)
      C=(C+CX)/2.
      CC=SQRT((C+C)/SCAL)
      DO 50 I=2,NX
      IF ((AO(I)+.5*DX).LT.-CC) I1=I+1
      IF ((AO(I)-.5*DX).LT.CC) I2=I
   50 CONTINUE
      ITE1(K)=I1
      ITE2(K)=I2
      CC=AO(I2)/CC
      KK=K1
      P=R1
   60 N=NP(KK)
      XA=EXP(XK*XS(1,KK))*COS(XK*YS(1,KK))-1.
      YA=EXP(XK*XS(1,KK))*SIN(XK*YS(1,KK))
      XB=EXP(XK*XS(N,KK))*COS(XK*YS(N,KK))-1.
      YB=EXP(XK*XS(N,KK))*SIN(XK*YS(N,KK))
      CX=SQRT(XA*XA+YA*YA)
      CX=(XA+CX)/2.
      Q=SQRT(CX/C)/CC
      DO 70 I=1,MX
      X(I)=Q*AO(I)
   70 CONTINUE
      ANGL=PI+PI
      U=1.
      V=0.
      DO 90 I=1,N
      XI=EXP(XK*XS(I,KK))*COS(XK*YS(I,KK))-1.
      YI=EXP(XK*XS(I,KK))*SIN(XK*YS(I,KK))
      R=SQRT(XI**2+YI**2)
      IF (R.EQ.0.) GO TO 80
      ANGL=ANGL+ATAN2((U*YI-V*XI),(U*XI+V*YI))
      U=XI
      V=YI
      R=SQRT((R+R)/SCAL)
      XP(I)=R*COS(.5*ANGL)
      YP(I)=R*SIN(.5*ANGL)
      GO TO 90
   80 ANGL=PI
      U=-1.
      V=0.
      XP(I)=0.
      YP(I)=0.
```

```
   90 CONTINUE
      ANGL=ATAN(SLOPT(KK))
      ANGL1=ATAN(YA/XA)
      ANGL2=ATAN(YB/XB)
      ANGL1=ANGL-.5*(ANGL1-TRAIL(KK))
      ANGL2=ANGL-.5*(ANGL2+TRAIL(KK))
      ANGL1=ANGL1+XK*YS(1,KK)
      ANGL2=ANGL2+XK*YS(N,KK)
      T1=TAN(ANGL1)
      T2=TAN(ANGL2)
      CALL SPLIF (1,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0.,IND)
      CALL INTPL (I1,I2,X,Y,1,N,XP,YP,D1,D2,D3,0)
C
C     DEFINITION OF THE VORTEX SHEET
      XA=XS(1,KK)
      X1=.25*XA
      A=SLOPT(KK)*(XA-X1)
      B=1./(XA-X1)
      ANGL=PI+PI
      U=1.
      V=0.
      M=I1-1
      XXS=1.+SCAL*(X(I1)**2-Y(I1)**2)/2./Q/Q
      YYS=SCAL*X(I1)*Y(I1)
      YYS=ATAN2(YYS,XXS)/XK
      DO 110 I=1,M
      E=1./1.05
  100 CONTINUE
      E=1.05*E
      XX1=ALOG(1.+.5*E*SCAL*X(I)**2/Q/Q)/XK
      D=B*(XX1-X1)
      YY1=YYS+A*(D-1.)/D
      XX=EXP(XK*XX1)*COS(XK*YY1)-1.
      YY=EXP(XK*XX1)*SIN(XK*YY1)
      R=SQRT(XX**2+YY**2)
      ANGL=ANGL+ATAN2((U*YY-V*XX),(U*XX+V*YY))
      U=XX
      V=YY
      R=SQRT((R+R)/SCAL)*Q
      XP(I)=R*COS(.5*ANGL)
      IF (XP(I).GT.X(I)) GO TO 100
      Y(I)=R*SIN(.5*ANGL)
  110 CONTINUE
      XB=XS(N,KK)
      A=SLOPT(KK)*(XB-X1)
      B=1./(XB-X1)
      ANGL=0.
      U=1.
      V=0.
      M=I2+1
      XXS=1.+SCAL*(X(I2)**2-Y(I2)**2)/2./Q/Q
      YYS=SCAL*X(I2)*Y(I2)
```

```
      YYS=ATAN2(YYS,XXS)/XK
      DO 130 I=M,MX
      E=1./1.05
  120 CONTINUE
      E=1.05*E
      XX1=ALOG(1.+.5*E*SCAL*X(I)**2/Q/Q)/XK
      D=B*(XX1-X1)
      YY1=YYS+A*(D-1.)/D
      XX=EXP(XK*XX1)*COS(XK*YY1)-1.
      YY=EXP(XK*XX1)*SIN(XK*YY1)
      R=SQRT(XX**2+YY**2)
      ANGL=ANGL+ATAN2((U*YY-V*XX),(U*XX+V*YY))
      U=XX
      V=YY
      R=SQRT((R+R)/SCAL)*Q
      XP(I)=R*COS(.5*ANGL)
      IF (XP(I).LT.X(I)) GO TO 120
      Y(I)=R*SIN(.5*ANGL)
  130 CONTINUE
      C=P*CC
      DO 140 I=1,MX
      SO(I,K)=SO(I,K)+Q*Y(I)
  140 CONTINUE
      IF (KK.EQ.K2) GO TO 150
      KK=K2
      P=R2
      GO TO 60
  150 DO 160 I=I1,I2
      IV(I,K)=2
  160 CONTINUE
      SO(1,K)=SO(2,K)+SO(2,K)-SO(3,K)
      SO(MX,K)=SO(NX,K)+SO(NX,K)-SO(NX-1,K)
      M=I1-1
      DO 170 I=2,M
      ZZ=Z(K)
      IF (ZZ.GE.Z(KTE1)) IV(I,K)=IVO
  170 CONTINUE
      M=I2+1
      DO 180 I=M,NX
      ZZ=Z(K)
      IF (ZZ.GE.Z(KTE1)) IV(I,K)=IVO
  180 CONTINUE
      K2=K2-1
      K=K+1
      IF (K.LE.KTE2) GO TO 20
      K1=3
      K2=NZ+2
  190 DO 200 I=2,NX
      ZZ=Z(K)
      IF (ZZ.LE.ZS(NC).AND.ZZ.GE.Z(KTE1)) IV(I,K)=IVO
  200 CONTINUE
      K=K+1
```

```
      IF (K.LE.K2) GO TO 190
      N=KTE2
      I=ITE1(KTE1)
      DO 220 K=K1,K2
      DO 210 I=2,NX
      IF (IV(I,K).GT.0) GO TO 210
      IF (IV(I+1,K+1).GT.0.OR.IV(I-1,K+1).GT.0) IV(I,K)=IV1
      IF (IV(I+1,K-1).GT.0.OR.IV(I-1,K-1).GT.0) IV(I,K)=IV1
  210 CONTINUE
      S=SO(LX,K)
      IF (2./XK*S/(T-S*S).LT.1.E-05) IV(LX,K)=0
  220 CONTINUE
      DO 230 I=1,MX
      SO(I,MZ)=3.*(SO(I,MZ-1)-SO(I,MZ-2))+SO(I,NZ)
      SO(I,MZ)=KSYM*SO(I,MZ)+(1-KSYM)*SO(I,MZ-2)
      SO(I,K1-1)=SO(I,K1+1)
  230 CONTINUE
      ITE1(MZ)=KSYM*ITE1(MZ)+(1-KSYM)*ITE1(MZ-2)
      ITE1(K1-1)=ITE1(K1+1)
      ITE2(MZ)=NX+2-ITE1(MZ)
      ITE2(K1-1)=ITE2(K1+1)
      RETURN
      END



      SUBROUTINE ESTIM
C     INITIAL ESTIMATE OF REDUCED POTENTIAL
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),B0(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      T=2./SCAL
      KY=NY+1
      MZ=NZ+3
      DO 10 I=1,129
      DO 10 J=1,26
      DO 10 K=1,17
      G(I,J,K)=0.
   10 CONTINUE
      KZ=3
      DO 30 I=2,NX
      DO 20 K=KZ,MZ
      IF (IV(I,K).LT.2) GO TO 20
      DSI=SO(I+1,K)-SO(I-1,K)
      DSK=SO(I,K+1)-SO(I,K-1)
```

```
      SX=A1(I)*DSI
      SZ=C1(K)*DSK
      X=A0(I)
      Y=S0(I,K)
      XX=2./XK*X*(T+X**2+Y**2)/((T+X**2-Y**2)**2+(2.*X*Y)**2)
      YX=2./XK*Y*(T-X**2-Y**2)/((T+X**2-Y**2)**2+(2.*X*Y)**2)
      FH=XX*XX+YX*YX
      H=1./FH
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      A=C5(K)
      Y1=YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK
      P=A*COS(PA*Y1)
      Q=A*SIN(PA*Y1)
      FZZ=A*A
      FYZ=H*FZZ*BZ
      FXZ=H*FZZ*AZ
      FYY=H*H*(YX*YX+FZZ*(BZ*BZ+XX*XX))
      FXY=H*H*(-XX*YX+FZZ*(AZ*BZ+XX*YX))
      FXX=H*H*(XX*XX+FZZ*(AZ*AZ+YX*YX))
      AX=FXY-SX*FXX-SZ*FXZ
      AY=FYY-SX*FXY-SZ*FYZ
      AZ=FYZ-SX*FXZ-SZ*FZZ
      BY=AY-SX*AX-SZ*AZ
      V=SA*XX*P/FZZ-CA*YX-OMEGA*XX/C4(K)
      U=CA*XX+SA*YX*P/FZZ-OMEGA*YX/C4(K)
      W=CA*XZ(K)+SA*(Q+P*YZ(K))/FZZ-OMEGA*YZ(K)/C4(K)
      G(I,KY+1,K)=G(I,KY-1,K)+(AX*U+AY*V+AZ*W)/(BY*H1(KY))
   20 CONTINUE
   30 CONTINUE
      K1=KTE1
      K2=KTE2
      DO 40 K=K1,K2
      E0(K)=0.
   40 CONTINUE
      RETURN
      END



      SUBROUTINE BOUND
      DEFINES THE BOUNDARY VALUES OF THE VELOCITY POTENTIAL G
      COMMON G(129,26,17),S0(129,17),E0(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),B0(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
```

```
      COMMON /CAL/ P1,P2,P3,BETA,FR,RM,DG,GM,NS,U1,V1,W1,IR,JR,KR,IG,JG,
     1KG
      COMMON /SWP/ G1(129,26),G2(129,26),SX(129),SZ(129),SXX(129),SXZ(12
     19),SZZ(129),RO(129),R1(129),C(129),D(129),I1,I2,LX,MX,KY,MY,T1,AAO
     2,Q1,Q2,NM
      LX=NX/2+1
      MX=NX+1
      KY=NY+1
      MY=NY+2
      MZ=NZ+3
      DX=2./NX
      T1=DX*DX
      AAO=1./FMACH**2+.2
      ZSYM=1-KSYM
      Q1=2./P1
      Q2=1./P2
      RM=0.
      GM=0.
      NM=0
      FR=0.
      IR=0
      JR=0
      KR=0
      DG=0.
      IG=0
      JG=0
      KG=0
      NS=0
      I1=2
      I2=NX
      DO 10 J=1,MY
      DO 10 I=1,MX
      G1(I,J)=G(I,J,1)
      G2(I,J)=G(I,J,1)
   10 CONTINUE
      JT=1
   20 CONTINUE
      KL=(NZ+5-JT*(NZ+1))/2
      KM=KL+JT
      KN=KM+JT
      I=LX
      G(I,1,KL)=G(I,1,KN)+ZSYM*(G(I,2,KL)-G(I,2,KN))
      DSI=SO(I+1,KM)-SO(I-1,KM)
      DSK=SO(I,KN)-SO(I,KL)
      SX(I)=A1(I)*DSI
      SZ(I)=C1(KM)*DSK
      R=1.0
      DO 30 J=2,KY
      G(1,J,KL)=G(1,J,KN)+ZSYM*(G(2,J,KL)-G(2,J,KN))
      G(MX,J,KL)=G(MX,J,KN)+ZSYM*(G(NX,J,KL)-G(NX,J,KN))
      YP=BO(J)+SO(I,KM)
      IF (J.EQ.KY) R=AMINO(1,IV(I,KM))
```

```
      H=R*(1-YP**2)**2/(1.-R+(2./XK*YP)**2)
      AZ=-YP*YZ(KM)*2./XK/(1-YP**2)
      BZ=YP*XZ(KM)*2./XK/(1-YP**2)
      A=H*AZ*A1(I)*JT
      B=(H*(BZ-AZ*SX(I))-JT*SZ(I))*B1(J)*JT
      DGI=G(I+1,J,KM)-G(I-1,J,KM)
      DGJ=G(I,J+1,KM)-G(I,J-1,KM)
      G(I,J,KL)=G(I,J,KM)+(A*DGI-B*DGJ)/C1(KM)
      G1(I,J)=G(I,J,KL)
      G2(I,J)=G(I,J,KM)+3.*(G(I,J,KL)-G(I,J,KM))
      G(I,J,1)=G2(I,J)+ZSYM*(G(I,J,1)-G2(I,J))
   20 CONTINUE
      J=KY+1
      G(I,J,KL)=G(I,J,KM)+(A*DGI-B*DGJ)/C1(KM)
      G1(I,J)=G(I,J,KL)
      G2(I,J)=G(I,J,KM)+3.*(G(I,J,KL)-G(I,J,KM))
      G(I,J,1)=G2(I,J)+ZSYM*(G(I,J,1)-G2(I,J))
      N=NX/2-1
      DO 70 II=1,N
      I=LX-II
      GO TO 50
   40 CONTINUE
      I=LX+II
   50 CONTINUE
      DSI=SC(I+1,KM)-SC(I-1,KM)
      DSK=SC(I,KM)-SC(I,KL)
      SX(I)=A1(I)*DSI
      SZ(I)=C1(KM)*DSK
      G(I,1,KL)=G(I,1,KN)+ZSYM*(G(I,2,KL)-G(I,2,KN))
      DO 60 J=2,KY
      X=XC(I)
      Y=YC(J)+SC(I,KM)
      XX=2./XK*X*(1+X**2+Y**2)/((1+X**2-Y**2)**2+(2.*X*Y)**2)
      YX=2./XK*Y*(1-X**2-Y**2)/((1+X**2-Y**2)**2+(2.*X*Y)**2)
      H=1./(XX*XX+YX*YX)
      AZ=-XX*XZ(KM)-YX*YZ(KM)
      BZ=-XX*YZ(KM)+YX*XZ(KM)
      S=SIGN(1.,AZ)*JT
      A=H*ABS(AZ)*A1(I)
      B=(H*(BZ-AZ*SX(I))-JT*SZ(I))*B1(J)*JT
      IP=I+IFIX(S)
      IM=I-IFIX(S)
      DGI=G(IP,J,KN)-G(IM,J,KN)
      DGJ=G(I,J+1,KM)-G(I,J-1,KM)
      G(I,J,KL)=(C1(KM)*G(I,J,KN)+A*(G(IP,J,KL)+DGI)-B*DGJ)/(C1(KM)+A)
      G1(I,J)=G(I,J,KL)
      G2(I,J)=G(I,J,KN)+3.*(G(I,J,KL)-G(I,J,KM))
      G(I,J,1)=G2(I,J)+ZSYM*(G(I,J,1)-G2(I,J))
   60 CONTINUE
      J=KY+1
      G(I,J,KL)=(C1(KM)*G(I,J,KN)+A*(G(IP,J,KL)+DGI)-B*DGJ)/(C1(KM)+A)
      C1(I,J)=G(I,J,KL)
```

```
      IF (I.LT.LX) GO TO 40
   70 CONTINUE
      IF (JT.EQ.-1) GO TO 110
      KK=2
      L=3
      I=5
      J=NY
      K=L
      X=AO(I)
      Y=SO(I,K)+BO(J)
      Y1=PA*(YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      H=1./(XX*XX+YX*YX)
      P=C5(K)*COS(Y1)
      Q=C5(K)*SIN(Y1)
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      SX(I)=A1(I)*(SO(I+1,K)-SO(I-1,K))
      SZ(I)=C1(K)*(SO(I,K+1)-SO(I,K-1))
      GX=A1(I)*(G(I+1,J,K)-G(I-1,J,K))
      GY=B1(J)*(G(I,J-1,K)-G(I,J+1,K))
      GZ=C1(K)*(G(I,J,K+1)-G(I,J,K-1))
      U=GX-SX(I)*GY
      V=GY
      W=GZ-SZ(I)*GY
      U0=H*(XX*U-YX*V)
      V0=H*((P*YX+Q*AZ)*U+(P*XX+Q*BZ)*V)+Q*W
      W0=H*((P*AZ-Q*YX)*U+(P*BZ-Q*XX)*V)+P*W
      U1=.9*U1+.1*U0
      V1=.9*V1+.1*V0
      W1=.9*W1+.1*W0
      DO 100 K=KK,MZ
      IS=ISIGN(1,2*(MZ-K)-1)
      IT=ISIGN(1,2*(K-KK)-1)
      DO 80 I=2,NX
      X=AO(I)
      Y=SO(I,K)+BO(2)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      Y1=PA*(YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK)
      G2(I,1)=G1(I,1)
      G1(I,1)=G(I,1,K)
      G(I,1,K)=G(I,3,K)-(YX*U1-XX*(V1*COS(Y1)-W1*SIN(Y1))/C5(K))/B1(2)
      G(I,1,K)=ZSYM*G(I,1,K)
      SX(I)=A1(I)*(SO(I+1,K)-SO(I-1,K))
      SZ(I)=C1(K)*(SO(I,K+IS)-SO(I,K-IT))
      DSII=SO(I+1,K)-SO(I,K)-SO(I,K)+SO(I-1,K)+A3(I)*DSI
      DSKK=SO(I,K+IS)-SO(I,K)-SO(I,K)+SO(I,K-IT)+C3(K)*DSK
      DSIK=SO(I+1,K+IS)-SO(I-1,K+IS)-SO(I+1,K-IT)+SO(I-1,K-IT)
```

```
      SXX(I)=A2(I)*DSII
      SZZ(I)=C2(K)*DSKK
      SYZ(I)=T1*A1(I)*C1(K)*DSIK
   80 CONTINUE
      DO 90 J=2,KY
      X=A0(2)
      Y=S0(2,K)+B0(J)
      Y1=PA*(YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      P=C5(K)*COS(Y1)
      Q=C5(K)*SIN(Y1)
      GY=B1(J)*(G(2,J-1,K)-G(2,J+1,K))
      G2(1,J)=G1(1,J)
      G1(1,J)=G(1,J,K)
      G(1,J,K)=G(3,J,K)-(XX*U1+YX/C4(K)*(P*V1-Q*W1)+SX(2)*GY)/A1(2)
      G(1,J,K)=ZSYM*G(1,J,K)
      NL=NX-1
      X=A0(NX)
      Y=S0(NX,K)+B0(J)
      Y1=PA*(YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      P=C5(K)*COS(Y1)
      Q=C5(K)*SIN(Y1)
      GY=B1(J)*(G(NX,J-1,K)-G(NX,J+1,K))
      G2(MX,J)=G1(MX,J)
      G1(MX,J)=G(MX,J,K)
      G(MX,J,K)=G(NL,J,K)+(XX*U1+YX/C4(K)*(P*V1-Q*W1)+SX(NX)*GY)/A1(NX)
      G(MX,J,K)=ZSYM*G(MX,J,K)
   90 CONTINUE
      G(1,1,K)=G(1,2,K)+G(2,1,K)-G(2,2,K)*ZSYM
      G(MX,1,K)=G(MX,2,K)+G(NX,1,K)-G(NX,2,K)*ZSYM
      E=G(ITE2(K),KY,K)-G(ITE1(K),KY,K)
      G(1,KY+1,K)=G(MX,KY-1,K)-E*ZSYM
      G(MX,KY+1,K)=G(1,KY-1,K)+E*ZSYM
      G(MX,KY,K)=G(1,KY,K)+E*ZSYM
      IF (IS*IT.GT.0) CALL YSWEEP (K)
  100 CONTINUE
      IF (KSYM.NE.0) GO TO 110
      JT=-1
      GO TO 20
  110 CONTINUE
      FR=1.2*FR/AA0
      RM=1.2*RM/NM/AA0
      GM=GM/NM
      RETURN
      END
```

```
      SUBROUTINE YSWEEP (K)
C     THE EQUATIONS FOR G ARE SOLVED HERE ,FOR MIXED SUBSONIC
C     AND SUPERSONIC FLOW ,BY ROW RELAXATION ,AND BY USING A
C     ROTATED DIFFERENCE SCHEME
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),BO(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      COMMON /CAL/ P1,P2,P3,BETA,FF,FM,DG,GM,NS,U1,V1,W1,IR,JR,KR,IG,JG,
     1KG
      COMMON /SWP/ G1(129,26),G2(129,26),SX(129),SZ(129),SXX(129),SXZ(12
     19),SZZ(129),R0(129),R1(129),C(129),D(129),I1,I2,LX,MX,KY,MY,T1,AA0
     2,Q1,Q2,NM
      L=K
      J1=2
      IF (FMACH.GE.1.) J1=3
      C(I1-1)=0.
      D(I1-1)=0.
      DO 10 I=I1,I2
      R0(I)=1.
      R1(I)=1.
      G1(I,1)=G(I,1,L)
      G1(I,J1-1)=G(I,J1-1,L)
   10 CONTINUE
      J=J1
      I3=I2
   20 CONTINUE
      BC=-T1*B1(J)*C1(K)
      DO 60 I=I1,I3
      AB=-T1*A1(I)*B1(J)
      AC=T1*A1(I)*C1(K)
      X=AO(I)
      Y=SO(I,K)+BO(J)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      FH=1.-R0(I)+XX*XX+YX*YX
      H=R0(I)/FH
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      E=H*AZ
      F=H*BZ
      XXX=2./XK*((T+3.*X*X+Y*Y)*DEN-4.*X*X*(T+X*X+Y*Y)**2)/DEN**2
      YYY=4./XK*X*Y*(DEN+2.*(T+X*X+Y*Y)*(T-X*X-Y*Y))/DEN**2
      AA=R0(I)*C5(K)
      Y1=YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK
```

```
C     IN THE FOLLOWING LINES ,THE COEFFICIENTS OF THE JACOBIAN
C     MATRIX ARE DEFINED ,AS WELL AS THEIR DERIVATIVES WITH
C     RESPECT TO THE NEW COORDINATES
      A=H*XX
      B=H*YX
      AX=H**2*((YX**2-XX**2)*XXX+2.*XX*YX*YYY)
      AY=H**2*(-2.*XX*YX*XXX)+(YX**2-XX**2)*YYY)
      P=AA*COS(Y1*PA)
      Q=AA*SIN(Y1*PA)
      PX=-Q*YX
      PY=-Q*XX
      PZ=-P*PA-Q*F
      QX=P*PA*YX
      QY=P*PA*XX
      QZ=-Q+P*PA*F
      EX=-(AX*XZ(K)+AY*YZ(K))
      EY=-(AY*XZ(K)-AX*YZ(K))
      EZ=-(A*XZZ(K)+B*YZZ(K))
      FZ=-A*YZZ(K)+B*XZZ(K)
      YXJ=P*B+Q*E
      YYJ=P*A+Q*F
      ZXJ=P*E-Q*B
      ZYJ=P*F-Q*A
      YXU=PX*B+QX*E+P*AY+Q*EX
      YXV=PY*B+QY*E-P*AX+Q*EY
      YXW=PZ*B+QZ*E+Q*EZ
      YYU=PX*A+QX*F+P*AX-Q*FY
      YYV=PY*A+QY*F+P*AY+Q*EX
      YYW=PZ*A+QZ*F+Q*FZ
      ZXU=PX*E-QX*B+P*EX-Q*AY
      ZXV=PY*E-QY*B+P*EY+Q*AX
      ZXW=PZ*E-QZ*B+P*EZ
      ZYU=PX*F-QX*A-P*EY-Q*AX
      ZYV=PY*F-QY*A+P*EX-Q*AY
      ZYW=PZ*F-QZ*A+P*FZ
      DGI=G(I+1,J,L)-G(I-1,J,L)
      DGJ=G(I,J+1,L)-G1(I,J-1)
      DGK=G(I,J,L+1)-G1(I,J)
      DGII=G(I+1,J,L)-G(I,J,L)-G(I,J,L)+G(I-1,J,L)+A3(I)*DGI
      DGJJ=G(I,J+1,L)-G(I,J,L)-G(I,J,L)+G(I,J-1,L)-B3(J)*DGJ
      DGKK=G(I,J,L+1)-G(I,J,L)-G(I,J,L)+G(I,J,L-1)+C3(K)*DGK
      DGIJ=G(I+1,J+1,L)-G(I-1,J+1,L)-G(I+1,J-1,L)+G(I-1,J-1,L)
      DGIK=G(I+1,J,L+1)-G(I+1,J,L-1)-G(I-1,J,L+1)+G(I-1,J,L-1)
      DGJK=G(I,J+1,L+1)-G(I,J-1,L+1)-G(I,J+1,L-1)+G(I,J-1,L-1)
      GX=A1(I)*DGI
      GY=-B1(J)*DGJ
      GZ=C1(K)*DGK
      UB=GX-SX(I)*GY
      VB=GY
      WB=GZ-SZ(I)*GY
      U=UB+CA*XX+SA*YX*P/C4(K)
      V=VB+SA*XX*P/C4(K)-CA*YX
```

```
      W=WB+CA*XZ(K)+SA*(Q+P*YZ(K))/C4(K)
      UO=A*U-B*V
      VO=YXJ*U+YYJ*V+Q*W
      WO=ZXJ*U+ZYJ*V+P*W
      QQ=UO**2+VO**2+WO**2
      VO=VO-OMEGA*P/C4(K)
      WO=WO+OMEGA*Q/C4(K)
      AA=DIM(AAO,.2*(QQ-2.*OMEGA*(B*U+A*V)))
      CC=UO*UO+VO*VO+WO*WO
      FXX=A*A+YXJ*YXJ+ZXJ*ZXJ
      FYY=B*B+YYJ*YYJ+ZYJ*ZYJ
      FZZ=Q*Q+P*P
      FXY=-A*B+YXJ*YYJ+ZXJ*ZYJ
      FXZ=Q*YXJ+P*ZXJ
      FYZ=Q*YYJ+P*ZYJ
      BU=A*UO+YXJ*VO+ZXJ*WO
      BV=-B*UO+YYJ*VO+ZYJ*WO
      BW=Q*VO+P*WO
      DU=A*AX-B*AY+YXJ*YXU+YYJ*YXV+Q*YXW+ZXJ*ZXU+ZYJ*ZXV+P*ZXW
      DV=-A*AY-B*AX+YXJ*YYU+YYJ*YYV+Q*YYW+ZXJ*ZYU+ZYJ*ZYV+P*ZYW
      DW=YXJ*QX+YYJ*QY+Q*QZ+ZXJ*PX+ZYJ*PY+P*PZ
      CU=UO*(AX*BU+AY*BV)+VO*(YXU*BU+YXV*BV+YXW*BW)+WO*(ZXU*BU+ZXV*BV+ZX
     1W*BW)
      CV=UO*(-AY*BU+AX*BV)+VO*(YYU*BU+YYV*BV+YYW*BW)+WO*(ZYU*BU+ZYV*BV+Z
     1YW*BW)
      CW=VO*(QX*BU+QY*BV+QZ*BW)+WO*(PX*BU+PY*BV+PZ*BW)
      FXX=FXX+R1(I)-RO(I)
      FYY=FYY+1.-RO(I)
      FYY=SX(I)**2*FXX+FYY+SZ(I)**2*FZZ-2.*SX(I)*FXY-2.*SZ(I)*FYZ+2.*SX(
     1I)*SZ(I)*FXZ
      FXY=FXY-SZ(I)*FXZ-SX(I)*FXX
      FYZ=FYZ-SX(I)*FXZ-SZ(I)*FZZ
      AV=BV-SX(I)*BU-SZ(I)*BW
      UU=BU*BU
      UV=BU*AV
      UW=BU*BW
      VV=AV*AV
      VW=AV*BW
      WW=BW*BW
      AXX=FXX*AA-UU
      AZZ=FZZ*AA-WW
      AXZ=2.*(FXZ*AA-UW)
      R=-(AXX*SXX(I)+AZZ*SZZ(I)+AXZ*SXZ(I))*GY+T1*((AA*DU-CU)*UB+(AA*DV-
     1CV)*VB+(AA*DW-CW)*WB)
      AXT=ABS(BU*A1(I))
      AYT=ABS(AV*B1(J))
      AZT=ABS(BW*C1(K))
      A=RO(I)*BETA*AA/AMAX1(AXT,AYT,AZT,1.-RO(I))
      AXT=A*AXT
      AYT=A*AYT
      AZT=A*AZT
C
```

```
C     SUPERSONIC POINTS AND SUBSONIC POINTS ARE SEPARATED
      IF (QQ.GE.AA) GO TO 30
      AXX=AXX*A2(I)
      AYY=(FYY*AA-VV)*B2(J)
      AZZ=AZZ*C2(K)
      AXY=(FXY*AA-UV)*(AB+AB)
      AXZ=AXZ*AC
      AYZ=(FYZ*AA-VW)*(BC+BC)
      BP=AXX
      BM=AXX
      B=-AXX-AXX-Q1*(AYY+AZZ)
      R=AXX*DGII+AYY*DGJJ+AZZ*DGKK+AXY*DGIJ+AYZ*DGJK+AXZ*DGIK+R
      GO TO 40
   30 CONTINUE
      NS=NS+1
      SI=SIGN(1.,U)
      IM=I-IFIX(SI)
      IMM=IM-IFIX(SI)
      AXX=UU*A2(I)
      AYY=VV*B2(J)
      AZZ=WW*C2(K)
      AXY=8.*SI*UV*AB
      AXZ=8.*SI*UW*AC
      AYZ=8.*VW*BC
      BXX=(FXX*QQ-UU)*A2(I)
      BYY=(FYY*QQ-VV)*B2(J)
      BZZ=(FZZ*QQ-WW)*C2(K)
      BXY=(FXY*QQ-UV)*(AB+AB)
      BXZ=(FXZ*QQ-UW)*(AC+AC)
      BYZ=(FYZ*QQ-VW)*(BC+BC)
      AQ=AA/QQ
      DELTAG=BXX*DGII+BYY*DGJJ+BZZ*DGKK+BXY*DGIJ+BYZ*DGJK+BXZ*DGIK
      DGII=G(I,J,L)-G(IM,J,L)-G(IM,J,L)+G(IMM,J,L)+A3(I)*DGI
      DGJJ=G(I,J,L)-G(I,J-1,L)-G(I,J-1,L)+G1(I,J-2)-B3(J)*DGJ
      DGKK=G(I,J,L)-G(I,J,L-1)-G(I,J,L-1)+G2(I,J)+C3(K)*DGK
      DGIJ=G(I,J,L)-G(IM,J,L)-G(I,J-1,L)+G(IM,J-1,L)
      DGIK=G(I,J,L)-G(I,J,L-1)-G(IM,J,L)+G(IM,J,L-1)
      DGJK=G(I,J,L)-G(I,J,L-1)-G(I,J-1,L)+G(I,J-1,L-1)
      GSS=AXX*DGII+AYY*DGJJ+AZZ*DGKK+AXY*DGIJ+AYZ*DGJK+AXZ*DGIK
      B=.5*(AQ-1.)*(AXX+AXX+AXY+AXZ)
      BP=AQ*BXX-(1.-SI)*B
      BM=AQ*BXX-(1.+SI)*B
      B=-AQ*(BXX+BXX+Q2*(BYY+BZZ))+(AQ-1.)*(2.*(AXX+AYY+AZZ)+AXY+AYZ+AXZ
     1)
      R=(AQ-1.)*GSS+AQ*DELTAG+R
   40 CONTINUE
      IF (ABS(R).LE.ABS(FR)) GO TO 50
      FR=R
      IR=I
      JR=J
      KR=K
   50 CONTINUE
```

```
      R=R-AYT*(G1(I,J-1)-G(I,J-1,L))-AZT*(G1(I,J)-G(I,J,L-1))
      RM=RM+ABS(R)
      NM=NM+1
      B=B -AXT-AYT-AZT
      BM=BM+AXT
      B=1./(B-BM*C(I-1))
      C(I)=B*BP
      D(I)=B*(R-BM*D(I-1))
 60   CONTINUE
      CG=0.
      I=I3
      DO 80 M=I1,I3
      CG=D(I)-C(I)*CG
      GM=GM+ABS(CG)
      IF (ABS(CG).LE.ABS(DG)) GO TO 70
      DG=CG
      IG=I
      JG=J
      KG=K
 70   CONTINUE
      G2(I,J)=G1(I,J)
      G1(I,J)=G(I,J,L)
      G(I,J,L)=G(I,J,L)-CG
      I=I-1
 80   CONTINUE
      J=J+1
      IF (J-KY) 20,90,110
 90   CONTINUE
      IF (I2.GT.ITE2(K)) I3=ITE2(K)
      IF (ITE2(K).EQ.MX) I3=LX
      DO 100 I=I1,I3
      LV=IABS(1-IABS(IV(I,K)))
      RO(I)=AMIN0(LV,IABS(IV(I,K)))
      R1(I)=LV
100   CONTINUE
      GO TO 20
110   CONTINUE
      I=LX+1
      IO=NX+2-I3
      IF (K.GT.KTE2) GO TO 130
      DO 120 I=IO,I3
      X=AO(I)
      Y=SO(I,K)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      A=1.-RO(I)+XX*XX+YX*YX
      H=RO(I)/A
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      A=RO(I)*C5(K)
      Y1=YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK
```

```
      P=A*COS(PA*Y1)
      Q=A*SIN(PA*Y1)
      FZZ=A*A
      FYY=H*H*(YX*YX+FZZ*(BZ*BZ+XX*XX))+1.-RO(I)
      FXX=H*H*(XX*XX+FZZ*(AZ*AZ+YX*YX))+1.-RO(I)
      FXY=H*H*(-XX*YX+FZZ*(AZ*BZ+XX*YX))
      FXZ=H*FZZ*AZ
      FYZ=H*FZZ*BZ
      AX=FXY-SX(I)*FXX-SZ(I)*FXZ
      AY=FYY-SX(I)*FXY-SZ(I)*FYZ
      AZ=FYZ-SX(I)*FXZ-SZ(I)*FZZ
      BY=AY-SX(I)*AX-SZ(I)*AZ
      DGI=G(I+1,KY,L)-G(I-1,KY,L)
      DGK=G(I,KY,L+1)-G2(I,KY)
      V=SA*XX*P/C4(K)-CA*YX-OMEGA*XX/C4(K)
      U=A1(I)*DGI+CA*XX+SA*YX*P/C4(K)-OMEGA*YX/C4(K)
      W=C1(K)*DGK+CA*XZ(K)+SA*(Q+P*YZ(K))/C4(K)-OMEGA*YZ(K)/C4(K)
      G(I,KY+1,L)=G(I,KY-1,L)+(AX*U+AY*V+AZ*W)/(BY*H1(KY))
  120 CONTINUE
      I=IO
      IF (IO.NE.ITE1(K)) GO TO 130
      E=G(I3,KY,L)-G(IO,KY,L)
      EO(L)=EO(L)+P3*(E-EO(L))
  130 CONTINUE
      IF (I.LE.I1) RETURN
      I=I-1
      E=0.
      IF (IV(I,K).NE.1) GO TO 140
      E=EO(L)
  140 CONTINUE
      M=NX+2-I
      G(I,KY+1,L)=G(M,KY-1,L)-E
      G(M,KY+1,L)=G(I,KY-1,L)+E
      G2(M,KY)=G1(M,KY)
      G1(M,KY)=G(M,KY,L)
      G(M,KY,L)=G(I,KY,L)+E
      GO TO 130
      END


      SUBROUTINE VELO (K,L,SV,SM,CP,X,Y)
C     CALCULATES SURFACE VELOCITY AND PRESSURE COEFFICIENT
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,AO(129),A1(129),A2(129),A3(129),BO(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
```

```
     4HA,CA,SA,FMACH,T,PA,ABLADE
      DIMENSION SV(1), SM(1), CP(1), X(1), Y(1)
      I10=ITE1(K)
      I20=ITE2(K)
      J=NY+1
      Q1=.2*FMACH**2
      T1=1./(.7*FMACH**2)
      B=C5(K)
      DO 10 I=I10,I20
      A=A0(I)
      D=S0(I,K)
      DEN=(T+A**2-D**2)**2+(2.*A*D)**2
      XX=2./XK*A*(T+A**2+D**2)/DEN
      YX=2./XK*D*(T-A**2-D**2)/DEN
      FH=XX*XX+YX*YX
      H=0.
      IF (IV(I,K).NE.0) H=1./FH
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      DSI=S0(I+1,K)-S0(I-1,K)
      DSK=S0(I,K+1)-S0(I,K-1)
      SX=A1(I)*DSI
      SZ=C1(K)*DSK
      Y1=YC(K)+ATAN2(2.*A*D,T+A*A-D*D)/XK
      P=B*COS(PA*Y1)
      Q=B*SIN(PA*Y1)
      DGI=G(I+1,J,L)-G(I-1,J,L)
      DGJ=G(I,J+1,L)-G(I,J-1,L)
      DGK=G(I,J,L+1)-G(I,J,L-1)
      U=A1(I)*DGI+SX*B1(J)*DGJ+CA*XX+SA*YX*P/C4(K)
      V=-B1(J)*DGJ+SA*XX*P/C4(K)-CA*YX
      W=C1(K)*DGK+SZ*B1(J)*DGJ+CA*XZ(K)+SA*(Q+P*YZ(K))/C4(K)
      U0=H*(XX*U-YX*V)
      V0=H*((P*YX+Q*AZ)*U+(P*XX+Q*BZ)*V)+Q*W
      W0=H*((P*AZ-Q*YX)*U+(P*BZ-Q*XX)*V)+P*W
      QQ=U0*U0+V0*V0+W0*W0
      SV(I)=SIGN(SQRT(QQ),U)
      IF (IV(I,K).EQ.0) SV(I)=SV(I-1)+SV(I-1)-SV(I-2)
      QQ=1.+Q1*(1.-QQ+2.*OMEGA*H*(YX*U+XX*V))
      SM(I)=FMACH*SV(I)/SQRT(QQ)
      CP(I)=T1*(QQ**3.5-1.)
      XI=1.+.5*SCAL*(A*A-D*D)
      YI=SCAL*A*D
      X(I)=XC(K)+ALOG(SQRT(XI**2+YI**2))/XK
      Y(I)=YC(K)+ATAN2(YI,XI)/XK
      A2(I)=XZ(K)-YX*SZ
      A3(I)=YZ(K)+XX*SZ
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CPLOT (I1,I2,X,B,D,FMACH)
C     PLOTS CP AT EQUAL INTERVALS IN THE MAPPED PLANE
      DIMENSION KODE(3), LINE(100), X(1), B(1), D(1)
      DATA KODE/1H ,1H+,1HO/
      IWRIT=6
      WRITE (IWRIT,60)
      DO 10 I=1,100
   10 LINE(I)=KODE(1)
      FDEN=1./B(I2)
      AMAX=0.
      CMMAX=0.
      CPMAX=0.
      DO 20 I=I1,I2
      AMAX=AMAX1(AMAX,ABS(X(I)))
      CMMAX=AMIN1(CMMAX,D(I))
   20 CPMAX=AMAX1(CPMAX,D(I))
      CMAX=CPMAX-CMMAX
      DO 50 I=I1,I2
      XFRAC=FDEN*B(I)
      K1=(59./AMAX)*ABS(X(I))+41.
      K1=MIN0(K1,100)
      LINE(K1)=KODE(2)
      K2=41
      K2=MIN0(K2,100)
      LINE(K2)=KODE(3)
      K3=(36./CMAX)*(D(I)-CMMAX)+1.
      K3=MIN0(K3,40)
      IF (K3.GE.1) GO TO 30
      K3=1.
      GO TO 40
   30 LINE(K3)=KODE(2)
   40 K4=1
      LINE(K4)=KODE(3)
      JJ=0
      WRITE (IWRIT,70) XFRAC,X(I),JJ,LINE
      LINE(K1)=KODE(1)
      LINE(K2)=KODE(1)
      LINE(K3)=KODE(1)
      LINE(K4)=KODE(1)
   50 CONTINUE
      RETURN
C
   60 FORMAT (1X,/,3X,3HX/C,4X,5HMCOMP,4X,5HMDSGN,3H ON)
   70 FORMAT (1X,F5.2,F9.3,I3,2X,100A1)
      END
```

```
      SUBROUTINE SPEED (K)
C     THE SPEED AND THE  MACH NUMBER ARE COMPUTED AT EACH GRID
C     POINT , THEN THE MACH NUMBERS ARE WRITTEN ON THE OUTPUT
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),B0(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      DIMENSION IS(24)
      C1=1./FMACH**2+0.2
      A=C5(K)
      I1=2
      I2=NX
      KY=NY+1
      WRITE (6,30)
      WRITE (6,40) FMACH
      DO 20 I=I1,I2
      DSI=SO(I+1,K)-SO(I-1,K)
      DSK=SO(I,K+1)-SO(I,K-1)
      SX=A1(I)*DSI
      SZ=C1(K)*DSK
      DO 10 J=2,KY
      X=A0(I)
      Y=SO(I,K)+B0(J)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      FH=XX**2+YX**2
      H=0.
      IF (FH.GT..1E-09) H=1./FH
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      Y1=YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK
      P=A*COS(PA*Y1)
      Q=A*SIN(PA*Y1)
      DGI=G(I+1,J,K)-G(I-1,J,K)
      DGJ=G(I,J+1,K)-G(I,J-1,K)
      DGK=G(I,J,K+1)-G(I,J,K-1)
      GX=A1(I)*DGI
      GY=-B1(J)*DGJ
      U=GX-SX*GY+CA*XX+SA*YX*P/C4(K)
      V=GY+SA*XX*P/C4(K)-CA*YX
      W=C1(K)*DGK-SZ*GY+CA*XZ(K)+SA*(Q+P*YZ(K))/C4(K)
      U0=H*(XX*U-YX*V)
      V0=H*((P*YX+Q*AZ)*U+(P*XX+Q*BZ)*V)+Q*W
      W0=H*((P*AZ-Q*YX)*U+(P*BZ-Q*XX)*V)+P*W
      Q=U0*U0+V0*V0+W0*W0
```

```
      IS(J)=SQRT(Q/(Q1-0.2*(Q-2.*OMEGA*H*(YX*U+XX*V))))*1000.
   10 CONTINUE
      WRITE (6,50) (IS(J),J=2,KY)
   20 CONTINUE
      RETURN
C
   30 FORMAT (1H1)
   40 FORMAT (18H PRINTOUT OF SPEED,/,13H    FMACH    = ,F4.2,/)
   50 FORMAT (24I5)
      END



      SUBROUTINE FORCF (I1,I2,X,Y,CP,AL,CHORD,XM,CL,CD,CM,XK,YK,ZI,PA)
C     CALCULATES SECTION FORCE COEFFICIENTS
      DIMENSION X(1), Y(1), XK(1), YK(1), CP(1)
      RAD=57.29578
      ALPHA=AL/RAD
      CL=0.
      CD=0.
      CM=0.
      N=I2-1
      DO 10 I=I1,N
      DX=(X(I+1)-X(I))/CHORD
      DY=(Y(I+1)-Y(I))/CHORD/ZI
      XA=(.5*(X(I+1)+X(I))-XM)/CHORD
      YA=.5*(Y(I+1)+Y(I))
      CPA=.5*(CP(I+1)+CP(I))
      DX=DX*COS(PA*YA)+XK(I)*SIN(PA*YA)*DY
      DY=DY*(1.+YK(I)*SIN(PA*YA)**2/ZI)
      YA=((1.-PA)*YA+SIN(PA*YA)/ZI)/CHORD
      DCL=-CPA*DX
      DCD=CPA*DY
      CL=CL+DCL
      CD=CD+DCD
   10 CM=CM+DCD*YA-DCL*XA
      DCL=CL*COS(ALPHA)-CD*SIN(ALPHA)
      CD=CL*SIN(ALPHA)+CD*COS(ALPHA)
      CL=DCL
      RETURN
      END
```

```
      SUBROUTINE TOTFOR (KTE1,KTE2,CHORD,SCL,SCD,SCM,Z,XC,C5,CL,CD,CMP,C
     1MR,CMY,PA,ABLADE)
C     CALCULATES TOTAL FORCE COEFFICIENTS
      DIMENSION CHORD(1), SCL(1), SCD(1), SCM(1), Z(1), XC(1), C5(1)
      SPAN=Z(KTE2)-Z(KTE1)
      CL=0.
      CD=0.
      CMP=0.
      CMR=0.
      CMY=0.
      S=0.
      N=KTE2-1
      DO 10 K=KTE1,N
      DZ=.5*(1./C5(K+1)-1./C5(K))
      AZ=.5*(1./C5(K+1)+1./C5(K))
      DZ=DZ+(1.-PA)*(Z(K+1)-Z(K))/2.
      AZ=AZ+(1.-PA)*(Z(K+1)+Z(K))/2.
      CL=CL+DZ*(SCL(K+1)*CHORD(K+1)+SCL(K)*CHORD(K))
      CD=CD+DZ*(SCD(K+1)*CHORD(K+1)+SCD(K)*CHORD(K))
      CMP=CMP+DZ*(CHORD(K+1)*(SCM(K+1)*CHORD(K+1)-SCL(K+1)*XC(K+1))+CHOR
     1D(K)*(SCM(K)*CHORD(K)-SCL(K)*XC(K)))
      CMR=CMR+AZ*DZ*(SCL(K+1)*CHORD(K+1)+SCL(K)*CHORD(K))
      CMY=CMY+AZ*DZ*(SCD(K+1)*CHORD(K+1)+SCD(K)*CHORD(K))
   10 S=S+DZ*(CHORD(K+1)+CHORD(K))
      ABLADE=S
      CL=CL/S
      CD=CD/S
      CMP=CMP*SPAN/S**2
      CMR=(CMR+CMR)/(S*SPAN)
      CMY=(CMY+CMY)/(S*SPAN)
      RETURN
      END


      SUBROUTINE REFIN
C     HALVES MESH SIZE
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),B0(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      MX=NX+1
      KY=NY+1
      MY=NY+2
      MZ=NZ+3
      MXO=NX/2+1
```

C-2

```
      MZ0=NZ/2+3
      KK=2
      DO 60 K=KK,MZ0
      J=NY/2+1
      JJ=KY
   10 I=MX0
      II=MX
   20 G(II,JJ,K)=G(I,J,K)
      I=I-1
      II=II-2
      IF (I.GT.0) GO TO 20
      J=J-1
      JJ=JJ-2
      IF (J.GT.0) GO TO 10
      DO 30 J=1,KY,2
      DO 30 I=2,NX,2
   30 G(I,J,K)=.5*(G(I+1,J,K)+G(I-1,J,K))
      DO 50 I=1,MX
      DO 40 J=2,NY,2
   40 G(I,J,K)=.5*(G(I,J+1,K)+G(I,J-1,K))
   50 G(I,MY,K)=0.
   60 CONTINUE
      MZM=MZ0
      MZST=MZ
   70 CONTINUE
      DO 80 J=1,MY
      DO 80 I=1,MX
   80 G(I,J,MZST)=G(I,J,MZM)
      IF (MZST.EQ.1) GO TO 100
      MZST=MZST-1
      DO 90 J=1,MY
      DO 90 I=1,MX
   90 G(I,J,MZST)=0.5*(G(I,J,MZM)+G(I,J,MZM-1))
      MZM=MZM-1
      IF(MZST.EQ.1) GO TO 100
      MZST=MZST-1
      GO TO 70
  100 CONTINUE
      KK=3
      DO 150 K=KK,MZ
      I=MX0+1
      IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 120
      I1=ITE1(K)
      I2=ITE2(K)
      DO 110 I=I1,I2
      DSI=SO(I+1,K)-SO(I-1,K)
      DSK=SO(I,K+1)-SO(I,K-1)
      SX=A1(I)*DSI
      SZ=C1(K)*DSK
      DGI=G(I+1,KY,K)-G(I-1,KY,K)
      DGK=G(I,KY,K+1)-G(I,KY,K-1)
      R=AMIN0(1,IV(I,K))
```

```
      X=AC(I)
      Y=SC(I,K)
      DEN=(T+X*X-Y*Y)**2+(2.*X*Y)**2
      XX=2./XK*X*(T+X*X+Y*Y)/DEN
      YX=2./XK*Y*(T-X*X-Y*Y)/DEN
      A=1.0-F+XX*XX+YX*YX
      H=R/A
      AZ=-XX*XZ(K)-YX*YZ(K)
      BZ=-XX*YZ(K)+YX*XZ(K)
      A=R*C5(K)
      Y1=YC(K)+ATAN2(2.*X*Y,T+X*X-Y*Y)/XK
      P=A*COS(PA*Y1)
      Q=A*SIN(PA*Y1)
      FZZ=A*A
      FYY=H*H*(YX*YX+FZZ*(BZ*BZ+XX*XX))+1.-R
      FXX=H*H*(XX*XX+FZZ*(AZ*AZ+YX*YX))+1.-R
      FXY=H*H*(-XX*YX+FZZ*(AZ*BZ+XX*YX))
      FXZ=H*FZZ*AZ
      FYZ=H*FZZ*BZ
      AX=FXY-SX*FXX-SZ*FXZ
      AY=FYY-SX*FXY-SZ*FYZ
      AZ=FYZ-SX*FXZ-SZ*FZZ
      BY=AY-SX*AX-SZ*AZ
      U=A1(I)*DGI+CA*XX+SA*YX*P/C4(K)-OMEGA*YX/C4(K)
      W=C1(K)*DGK+CA*XZ(K)+SA*(Q+P*YZ(K))/C4(K)-OMEGA*YZ(K)/C4(K)
      V=SA*XX*P/C4(K)-CA*YX-OMEGA*XX/C4(K)
  110 G(I,KY+1,K)=G(I,KY-1,K)+(AX*U+AY*V+AZ*W)/(BY*B1(KY))
      EC(K)=G(I2,KY,K)-G(I1,KY,K)
      I=I1
  120 I=I-1
      E=0.
      IF (IV(I,K).NE.1) GO TO 130
      E=EC(K)
  130 M=NX+2-I
      G(I,KY+1,K)=G(M,KY-1,K)-E
      G(M,KY+1,K)=G(I,KY-1,K)+E
      IF (IV(I,K).NE.-1) GO TO 140
      G(I,KY,K)=.5*G(I,KY,K-1)+.25*(G(I,KY,K+1)+G(M,KY,K+1))
      IF (IV(I,K+1).LT.1) G(I,KY,K)=.5*G(I,KY,K+1)+.25*(G(I,KY,K-1)+G(M,
     1KY,K-1))
      G(M,KY,K)=G(I,KY,K)
      G(I,KY-1,K)=.5*(G(I,KY,K)+G(I,KY-2,K))
      G(M,KY-1,K)=.5*(G(M,KY,K)+G(M,KY-2,K))
  140 IF (I.GT.2) GO TO 120
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPLIF (M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FOM,IND)
C     CUBIC SPLINE FIT WITH PRESCRIBED END CONDITIONS
      DIMENSION S(1), F(1), FP(1), FPP(1), FPPP(1)
      IF (IND.EQ.0) GO TO 180
      IND=0
      K=IABS(N-M)
      IF (K-1) 180,180,10
   10 K=(N-M)/K
      I=M
      J=M+K
      DS=S(J)-S(I)
      D=DS
      IF (DS) 20,180,20
   20 DF=(F(J)-F(I))/DS
      IF (KM-2) 30,40,50
   30 U=.5
      V=3.*(DF-VM)/DS
      GO TO 80
   40 U=0.
      V=VM
      GO TO 80
   50 U=-1.
      V=-DS*VM
      GO TO 80
   60 I=J
      J=J+K
      DS=S(J)-S(I)
      IF (D*DS) 180,180,70
   70 DF=(F(J)-F(I))/DS
      B=1./(DS+DS+U)
      U=B*DS
      V=B*(6.*DF-V)
   80 FP(I)=U
      FPP(I)=V
      U=(2.-U)*DS
      V=6.*DF+DS*V
      IF (J-N) 60,90,60
   90 IF (KN-2) 100,110,120
  100 V=(6.*VN-V)/U
      GO TO 130
  110 V=VN
      GO TO 130
  120 V=(DS*VN+FPP(I))/(1.+FP(I))
  130 B=V
      D=DS
  140 DS=S(J)-S(I)
      U=FPP(I)-FP(I)*V
```

```
      FPPP(I)=(V-U)/DS
      FPP(I)=U
      FP(I)=(F(J)-F(I))/DS-DS*(V+U+U)/6.
      V=U
      J=I
      I=I-K
      IF (J-M) 140,150,140
  150 I=N-K
      FPPP(N)=FPPP(I)
      FPP(N)=B
      FP(N)=DF+D*(FPP(I)+B+B)/6.
      IND=1
      IF (MODE) 180,180,160
  160 FPPP(J)=FQM
      V=FPP(J)
  170 I=J
      J=J+K
      DS=S(J)-S(I)
      U=FPP(J)
      FPPP(J)=FPPP(I)+.5*DS*(F(I)+F(J)-DS*DS*(U+V)/12.)
      V=U
      IF (J-N) 170,180,170
  180 CONTINUE
      RETURN
      END


      SUBROUTINE INTPL (MI,NI,SI,FI,M,N,S,F,FP,FPP,FPPP,MODE)
C     INTERPOLATION OF CUBIC SPLINE BY TAYLOR SERIES
      DIMENSION SI(1), FI(1), S(1), F(1), FP(1), FPP(1), FPPP(1)
      K=IABS(N-M)
      K=(N-M)/K
      I=M
      MIN=MI
      NIN=NI
      D=S(N)-S(M)
      IF (D*(SI(NI)-SI(MI))) 10,20,20
   10 MIN=NI
      NIN=MI
   20 KI=IABS(NIN-MIN)
      IF (KI) 40,40,30
   30 KI=(NIN-MIN)/KI
   40 II=MIN-KI
      C=0.
      IF (MODE) 60,60,50
   50 C=1.
```

```
   60 II=II+KI
      SS=SI(II)
   70 I=I+K
      IF (I-N) 80,90,80
   80 IF (D*(S(I)-SS)) 70,70,90
   90 J=I
      I=I-K
      SS=SS-S(I)
      FPPPP=C*(FPPP(J)-FPPP(I))/(S(J)-S(I))
      FF=FPPP(I)+.25*SS*FPPPP
      FF=FPP(I)+SS*FF/3.
      FF=FP(I)+.5*SS*FF
      FI(II)=F(I)+SS*FF
      IF (II-NIN) 60,100,60
  100 CONTINUE
      RETURN
      END




      SUBROUTINE THREED (IPLOT,SV,SM,CP,X,Y,TITLE,DC,AL,ZONE,FM2,DEV,CHO
     1RDO,XSCAL,PSCAL)
C     GENERATES THREE DIMENSIONAL CALCOMP PLOTS ON CDC 6600
      COMMON G(129,26,17),SO(129,17),EO(17),IV(129,17),ITE1(17),ITE2(17)
     1,A0(129),A1(129),A2(129),A3(129),B0(26),B1(26),B2(26),B3(26),Z(17)
     2,C1(17),C2(17),C3(17),C4(17),C5(17),XC(17),XZ(17),XZZ(17),YC(17),Y
     3Z(17),YZZ(17),KSYM,NX,NY,NZ,KTE1,KTE2,ISYM,SCAL,SCALZ,XK,OMEGA,ALP
     4HA,CA,SA,FMACH,T,PA,ABLADE
      DIMENSION X(1), Y(1), SV(1), SM(1), CP(1), TITLE(10), R(20)
      M=1
      IF (XSCAL.NE.0.) SCALX=.5*ABS(XSCAL)/CHORDO*ZONE
      DZ=Z(KTE2)-Z(KTE1)
      IF (PA.EQ.1.) DZ=1./C5(KTE2)-1./C5(KTE1)
      IF (PSCAL.GE.0.) SCALX=5./DZ
      SCALP=-1.00
      IF (PSCAL.NE.0.) SCALP=-.5/ABS(PSCAL)
      TX=3.0
      SX=-SCALX*XC(KTE1)
      IF (IPLOT.NE.1) GO TO 10
      CALL PLOTSBL (1000,25HANTOINE BOURGEADE 4337WWH)
   10 CONTINUE
      IPLOT=0
      CALL FRAME
      CALL PLOT (1.25,1.,-3)
      ENCODE (65,190,R) FMACH,FM2,DEV,AL
      CALL SYMBOL (0.0,0.75,.14,R,0.,65)
      ENCODE (60,200,R)
```

```
      CALL SYMBOL (.50,1.25,.14,R,0.,60)
      CONTINUE
   20 CONTINUE
      K=1
   30 CONTINUE
      K=K+1
      IF (K.GT.KTE2) GO TO 70
      IF (K.LT.KTE1) GO TO 30
      I1=ITE1(K)
      I2=ITE2(K)
      CALL VELC (K,K,SV,SM,CP,X,Y)
      SY=5.*(Z(K)-Z(KTE1))/(Z(KTE2)-Z(KTE1))+2.45
      SCP=5.*(Z(K)-Z(KTE1))/(Z(KTE2)-Z(KTE1))+2.75
      DO 40 I=I1,I2
      X(I)=SCALX*X(I)+SX
      Y(I)=SCALX*Y(I)+SY
      CP(I)=SCALP*CP(I)+SCP
   40 CONTINUE
      IF (M.EQ.2) GO TO 50
      N=I2-I1+1
      CALL LINE (X(I1),CP(I1),N,1,0,2,0.,1.,0.,1.)
      GO TO 30
   50 CONTINUE
      N=I2-I1+1
      DO 60 I=I1,I2
      X(I)=X(I)+TX
   60 CONTINUE
      CALL LINE (X(I1),Y(I1),N,1,0,2,0.,1.,0.,1.)
      GO TO 30
   70 CONTINUE
      M=M+1
      IF (M.GT.2) GO TO 80
      GO TO 20
   80 CONTINUE
      CALL FRAME
      CALL VELC (KTE1,KTE1,SV,SM,CP,X,Y)
      I1=ITE1(KTE1)
      I2=ITE2(KTE1)
      SCALX=1.5*XK/3.14159265
      DO 90 I=I1,I2
      X(I)=2.*X(I)*SCALX+SX+1.
      Y(I)=2.*Y(I)*SCALX+.5
      CP(I)=Y(I)+6.
   90 CONTINUE
      N=I2-I1+1
      CALL LINE (X(I1),Y(I1),N,1,0,2,0.,1.,0.,1.)
      CALL LINE (X(I1),CP(I1),N,1,0,2,0.,1.,0.,1.)
      ENCODE (60,210,R)
      CALL SYMBOL (1.0,8.5,.14,R,0.,60)
      ENCODE (60,220,R) DC
      CALL SYMBOL (1.0,7.75,.14,R,0.,60)
      CALL PLOT (-1.25,-1.,-3)
```

```
      LX=NX/2+1
      KY=NY+1
      NA=PA
      KD=(KTE2-KTE1+1)/2*NA+(1-NA)*(KTE2-KTE1+2)
      KD=MAX0(KD,2)
      K=KTE1
  100 CONTINUE
      CALL FRAME
      CALL PLOT (3.,4.5,-3)
      DO 110 J=2,KY
      D=BO(J)+SO(LX,K)
      A=XC(K)+ALOG(ABS(1.-.5*SCAL*D*D))/XK
      IF (D*D.LT.T) GO TO 120
  110 CONTINUE
  120 CONTINUE
      D=BO(KY)+SO(2,K)
      B=1.+(AO(2)*AO(2)-D*D)/T
      D=SCAL*AO(2)*D
      B=XC(K)+ALOG(B*B+D*D)/XK/2.
      A=-2./A
      B=4./B
      IF (A.LE.0.) A=B
      SSX=1.2/ZONE*XK
      SSX=AMIN1(SSX,A,B)
      DO 130 I=2,NX
      A=AO(I)
      LPLOT=3
      DO 130 J=2,KY
      D=BO(J)+SO(I,K)
      X1=1.+.5*SCAL*(A*A-D*D)
      Y1=SCAL*A*D
      X2=XC(K)+ALOG(X1*X1+Y1*Y1)/XK/2.
      Y2=YC(K)+ATAN2(Y1,X1)/XK
      Y3=SIN(Y2)/C5(K)
      X2=SSX*X2
      Y3=SSX*Y3
      CALL PLOT (X2,Y3,LPLOT)
      LPLOT=2
  130 CONTINUE
      LPLOT=3
      DO 140 J=2,KY
      D=BO(J)+SO(LX,K)
      X1=1.-.5*SCAL*D*D
      X2=XC(K)+ALOG(ABS(X1))/XK
      Y2=YC(K)-ATAN2(0.,X1)/XK
      Y3=SIN(Y2)/C5(K)
      X2=SSX*X2
      Y3=SSX*Y3
      CALL PLOT (X2,Y3,LPLOT)
      LPLOT=2
  140 CONTINUE
      DO 160 J=2,KY
```

```
      LPLOT=3
      DO 160 I=2,NX
      A=AO(I)
      D=BO(J)+SO(I,K)
      X1=1.+.5*SCAL*(A*A-D*D)
      Y1=SCAL*A*D
      X2=XC(K)+ALOG(X1*X1+Y1*Y1)/XK/2.
      Y2=YC(K)+ATAN2(Y1,X1)/XK
      Y3=SIN(Y2)/C5(K)
      X2=SSX*X2
      Y3=SSX*Y3
      IF (I.NE.LX) GO TO 150
      Y2=2.*YC(K)-Y2
      Y4=SSX*SIN(Y2)/C5(K)
      CALL PLOT (X2,Y4,LPLOT)
      LPLOT=3
  150 CONTINUE
      CALL PLOT (X2,Y3,LPLOT)
      LPLOT=2
  160 CONTINUE
      Z1=PA*ZONE*EXP(Z(K))+(1.-PA)*Z(K)
      ENCODE (60,230,R) Z1
      CALL SYMBOL (-1.0,4.5,.14,R,0.,60)
      CALL PLOT (-3.,-4.5,-3)
      K=K+KD
      IF (K.LE.KTE2) GO TO 100
      K=K-1
      IF (K.EQ.KTE2) GO TO 100
      DC=1./XK
      IF (PA.NE.1.0) GO TO 180
      N=2
      DO 170 I=1,N
      S=FLOAT(I)/FLOAT(N)
      X0=0.5/ZONE*S
      CALL CUT (X0,DC,SCALP,SV,SM,CP,X,Y,Z,YC,ITE1,ITE2,KTE1,KTE2,KSYM)
  170 CONTINUE
  180 CONTINUE
      RETURN
C
  190 FORMAT (5HM1 = ,F4.2,1H,2X,5HM2 = ,F5.2,1H,2X,6HDEV = ,F5.1,1H,2X,
     16HALP = ,F4.1)
  200 FORMAT (21HPRESSURE DISTRIBUTION,5X,14HBLADE PROFILE )
  210 FORMAT (23H CASCADE REPRESENTATION)
  220 FORMAT (9H    G/C =,F5.2)
  230 FORMAT (25H GRID ON THE SURFACE  Z =,F5.2)
      END
```

```
      SUBROUTINE CUT (XO,DC,SCALP,SV,SM,CP,X,Y,Z,YC,ITE1,ITE2,KTE1,KTE2,
     1KSYM)
C     THIS SUBROUTINE PLOTS SECTIONS OF THE COMPRESSOR
      COMMON /SWP/ G1(129,26),G2(129,26),SX(129),SZ(129),SXX(129),SXZ(12
     19),SZZ(129),RO(129),R1(129),C(129),D(129),I1,I2,LX,MX,KY,MY,I.,AA
     2,Q1,Q2,NM
      DIMENSION SV(1), SM(1), CP(1), X(1), Y(1), Z(1), ITE1(1), ITE2(1),
     1YC(1)
      CALL FRAME
      NZ=KTE2-KTE1+1
      DO 50 K=KTE1,KTE2
      CALL VELL (K,K,SV,SM,CP,X,Y)
      I10=ITE1(K)
      I20=ITE2(K)
      B=1.
      DO 10 I=I10,I20
      IF (X(I).LT.XO) GO TO 20
   10 CONTINUE
      I=I-1
      B=0.
   20 CONTINUE
      F=(XO-X(I))/(X(I-1)-X(I))
      SV(K)=B*(Y(I)+F*(Y(I-1)-Y(I))-YC(K))+YC(K)
      C(K)=(CP(I)+F*(CP(I-1)-CP(I)))*B*SCALP-2.
      B=1.
      DO 30 I=I10,I20
      M=I10+I20-I
      IF (X(M).LT.XO) GO TO 40
   30 CONTINUE
      B=0.
   40 CONTINUE
      F=(XO-X(M))/(X(M+1)-X(M))
      SM(K)=B*(Y(M)+F*(Y(M+1)-Y(M))-YC(K))+YC(K)
      D(K)=(CP(M)+F*(CP(M+1)-CP(M)))*B*SCALP+2.
      RO(K)=EXP(Z(K)-Z(KTE2))
      R1(K)=2.*RO(K)-1.5
   50 CONTINUE
      CALL PLOT (1.,5.,-3)
      CALL LINE (R1(KTE1),C(KTE1),NZ,1,0,2,0.,1.,0.,1.)
      CALL LINE (R1(KTE1),D(KTE1),NZ,1,0,2,0.,1.,0.,1.)
      M=1./DC
      CALL PLOT (6.,0.,-3)
      DO 80 L=1,M
      XL=L
      DO 60 K=KTE1,KTE2
      X(K)=3.*RO(K)*COS(SV(K)+2.*3.14159265*XL*DC)
      Y(K)=3.*RO(K)*SIN(SV(K)+2.*3.14159265*XL*DC)
   60 CONTINUE
      CALL LINE (X(KTE1),Y(KTE1),NZ,1,0,2,0.,1.,0.,1.)
      DO 70 K=KTE1,KTE2
      A=2.*3.14159265*XL*DC
      X(K)=3.*RO(K)*COS(SM(K)+A)
```

```
      Y(K)=3.*RO(K)*SIN(SM(K)+A)
   70 CONTINUE
      CALL LINE (X(KTE1),Y(KTE1),NZ,1,0,2,C.,1.,0.,1.)
   80 CONTINUE
      A=3.*EXP(Z(KTE1)-Z(KTE2))
      DO 90 I=1,97
      T=2.*3.14159265*FLOAT(I-1)/96.
      C(I)=3.*COS(T)
      D(I)=3.*SIN(T)
      X(I)=A*COS(T)
      Y(I)=A*SIN(T)
   90 CONTINUE
      CALL LINE (X(1),Y(1),97,1,0,2,C.,1.,0.,1.)
      IF (KSYM.EQ.1.) GO TO 100
      CALL LINE (C(1),D(1),97,1,0,2,C.,1.,0.,1.)
  100 CONTINUE
      X0=3.*X0/A
      ENCODE (60,110,R1) X0
      CALL SYMBOL (-2.1,4.0,.14,R1,C.,60)
      CALL PLOT (-7.,-5.,-3)
      RETURN
C
  110 FORMAT (25H SECTION IN THE PLANE X =,F5.3)
      END
```