NASA Contractor Report 3930

Numerical Studies of Two-Dimensional Flows

Gino Moretti

CONTRACT NAS1-17262
SEPTEMBER 1985

NASA

NASA Contractor Report 3930

Numerical Studies of Two-Dimensional Flows

Gino Moretti
GMAF, Inc.
Freeport, New York

Prepared for
Langley Research Center
under Contract NAS1-17262

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1985

1. Introduction

In the present Report, I discuss a technique for the analysis of two-dimensional, inviscid, compressible, transonic flows. Emphasis is put on the treatment of shocks. I will support my arguments with some numerical experiments on two-dimensional flows in ducts.

Whatever the range of Mach numbers, indeed, such flows provide a fertile ground for testing of numerical techniques and for the analysis of not-so-well-known features of steady and unsteady gasdynamical patterns. Topics of theoretical interest are: Formation and evolution of shocks, regular and Mach reflections, stability of shock patterns and (not considered in the present Report) interaction of unsteady internal shock patterns with the impinging flow. From a numerical standpoint, the relative merits and disadvantages of flux-difference splitting methods and of the λ-scheme with shock tracking can be tested.

The numerical scheme must:

a) be robust in unsteady, transonic regions,

b) properly represent the chosen inlet and outlet boundary models,

c) properly satisfy the boundary conditions on rigid walls,

d) provide accurate predictions of shock formation, regardless of the number and location of shocks,

e) provide an accurate analysis of shocks, their motion and interactions.

It is desired to satisfy the above requirements with a code, simple to understand, easily extendable to other problems, and as fast as possible. For a time-dependent code, where the advance in time is controlled by the CFL rule, the latter condition requires the use of coarse meshes, the reduction of calculations at any point to what is strictly necessary (elimination of redundancies), the calculation itself streamlined to a minimal number of operations (particularly multiplications and divisions), and the absence of logical or arithmetic branches (IF-statements).

In the present paper, I intend to outline a scheme which

seems to satisfy the above conditions.

2. The problem

Let an inlet be schematized as a duct, running from $x=-x_0$ to $x=x_0$. The shape of the lower wall is defined by $y=b(x)$. The shape of the upper wall is defined by $y=c(x)$. To simplify the analysis, we exclude any possibility of transonic interaction of the inlet flow with an infinite external flow; consequently, we can use a simple model to represent the external flow at the inflow boundary of the duct. In the same spirit of simplicity, we assume that the inlet itself is preceded by a straight channel and followed by another straight channel (Fig. 1).

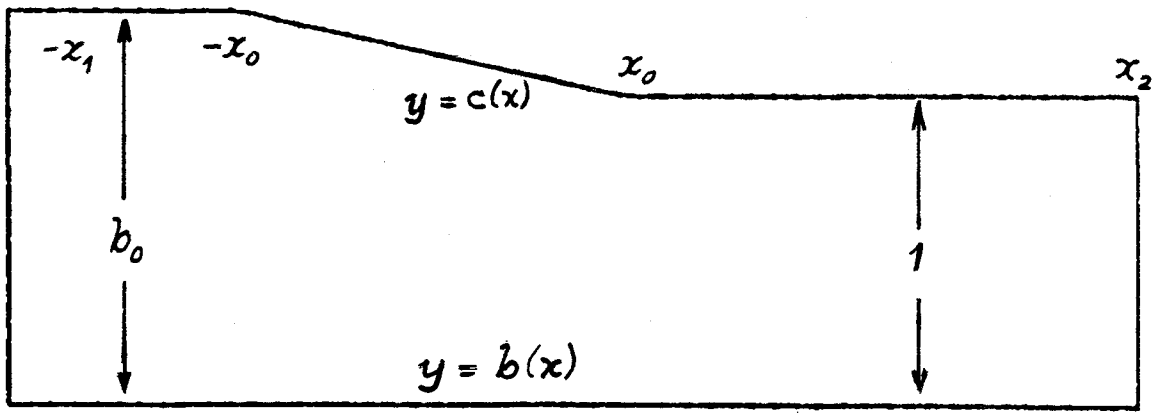

Figure 1.- Duct geometry for example problem.

In the numerical analysis, the flow will be computed between two sections, located at $x=-x_1$ and $x=x_2$. The flow problem can be modeled in three different ways:

1) The duct is permanently at rest. The gas in the duct is initially at rest. A diaphragm is abruptly broken at $x=x_2$, at time $t=0$, letting the gas in the duct communicate with an infinite cavity where the pressure is prescribed and constant. If the latter is lower then the initial pressure of the gas in the duct, a flow is generated and it will eventually reach a steady state.

2) The duct is permanently at rest. The gas in the duct is initially at rest. A normal shock wave reaches the duct from the left. We can think of it as generated by the rupture of a diaphragm located far to the left of $-x_1$, or by the push of a piston, started from rest with a constant speed at a time $t<0$. In either case, the shock is followed by a constant state. The shock itself travels from left to right within the duct and eventually disappears to the right of x_2.

3) The duct is initially at rest. The gas in the duct is also at rest. At $t=0$, the duct is impulsively set into motion

from right to left, in a direction parallel to the x-axis, with a velocity which is then maintained constant. The gas flow will be described by an observer moving with the duct, who will use the same frame of reference shown in Fig. 1. This, somewhat more artificial, way of starting the calculation may be interpreted as the limit of the more physical procedure of accelerating the duct from rest, the entire accelerating process taking place in one single computational step.

3. Equations of motion

As usual, we assume a pressure, a density and a length as reference values (p_{ref}, ρ_{ref}, x_{ref}, respectively). The reference velocity, temperature and time are thus:

$$u_{ref} = \sqrt{p_{ref}/\rho_{ref}},\quad \Theta_{ref} = p_{ref}/(R\rho_{ref}),\quad t_{ref} = x_{ref}/u_{ref} \tag{1}$$

where R is the gas constant. A nondimensional entropy, s, is defined as

$$s = \frac{1}{\gamma(\gamma-1)}\,[\ln p - \gamma \ln \rho] \tag{2}$$

Consequently,

$$\Theta = \exp[(\gamma-1)(P/\gamma + s)] \tag{3}$$

$$a^2 = \gamma\Theta \tag{4}$$

where γ is the ratio of specific heats.

The equations of motion can be written in the form:

$$\begin{aligned} P_t + \mathbf{q}.\nabla P + \gamma \nabla.\mathbf{q} &= 0 \\ \mathbf{q}_t + (\mathbf{q}.\nabla)\mathbf{q} + \frac{a^2}{\gamma}\nabla P &= 0 \\ s_t + \mathbf{q}.\nabla s &= 0 \end{aligned} \tag{5}$$

where $P = \ln p$, $\mathbf{q} = U\mathbf{i} + V\mathbf{j}$, and $\mathbf{i}$, $\mathbf{j}$ are the unit vectors of the x and y axes, respectively. From (3) and (4) it follows that

$$dP = \frac{\gamma}{\delta}\frac{da}{a} - \gamma\, ds \tag{6}$$

where

$$\delta = \frac{\gamma-1}{2} \tag{7}$$

Therefore, (5) may be replaced by

$$\frac{a_t}{\delta} + \mathbf{q}.\nabla \frac{a}{\delta} + a\,\nabla.\mathbf{q} - a\,s_t - a\,\mathbf{q}.\nabla s = 0$$

$$\mathbf{q}_t + (\mathbf{q}.\nabla)\mathbf{q} + a\,\nabla \frac{a}{\delta} - a^2\nabla s = 0 \qquad (8)$$

$$s_t + \mathbf{q}.\nabla s = 0$$

which are the obvious extension of Eqs. (4) of Ref. 1 to non-isentropic flow.

One can use the same procedure of Ref. 1, defining two arbitrary, but orthogonal, unit vectors $\mathbf{n}$ and $\boldsymbol{\tau}$ at any point, and calling α_0 the angle formed by $\mathbf{n}$ and $\mathbf{i}$. Using the same notations as in Ref. 1, in particular

$$\beta = \mathbf{q}.\nabla\alpha_0 \qquad (9)$$

$$F = a\,\mathbf{k}\times\mathbf{q}.\nabla\alpha_0 \qquad (10)$$

where $\mathbf{k} = \mathbf{i}\times\mathbf{j}$, and also

$$C = \cos\alpha_0\ , \quad S = \sin\alpha_0 \qquad (11)$$

$$\begin{aligned}
\rho_1 &= \frac{a}{\delta} + \mathbf{n}.\mathbf{q}, & \boldsymbol{\Lambda}_1 &= \mathbf{q} + a\mathbf{n} \\
\rho_2 &= \frac{a}{\delta} - \mathbf{n}.\mathbf{q}, & \boldsymbol{\Lambda}_2 &= \mathbf{q} - a\mathbf{n} \\
\rho_3 &= \frac{a}{\delta} + \boldsymbol{\tau}.\mathbf{q}, & \boldsymbol{\Lambda}_3 &= \mathbf{q} + a\boldsymbol{\tau} \\
\rho_4 &= \frac{a}{\delta} - \boldsymbol{\tau}.\mathbf{q}, & \boldsymbol{\Lambda}_4 &= \mathbf{q} - a\boldsymbol{\tau}
\end{aligned} \qquad (12)$$

the equations of motion take on the form:

$$\begin{aligned}
&\frac{2}{\delta}\,a_t + \sum \boldsymbol{\Lambda}_i.(\nabla\rho_i - a\,\nabla s) - 2\mathbf{q}.\nabla\frac{a}{\delta} + 4a\mathbf{q}.\nabla s + 2F = 0 \\
&2\mathbf{n}.\mathbf{q}_t + \boldsymbol{\Lambda}_1.(\nabla\rho_1 - a\,\nabla s) - \boldsymbol{\Lambda}_2.(\nabla\rho_2 - a\,\nabla s) - 2\beta\mathbf{q}.\boldsymbol{\tau} = 0 \\
&2\boldsymbol{\tau}.\mathbf{q}_t + \boldsymbol{\Lambda}_3.(\nabla\rho_3 - a\,\nabla s) - \boldsymbol{\Lambda}_4.(\nabla\rho_4 - a\,\nabla s) + 2\beta\mathbf{q}.\mathbf{n} = 0 \\
&s_t + \mathbf{q}.\nabla s = 0
\end{aligned} \qquad (13)$$

We will also introduce the velocity components:

$$\begin{aligned}
u &= CU + SV \\
v &= -SU + CV
\end{aligned} \qquad (14)$$

along the $\mathbf{n}$- and $\boldsymbol{\tau}$-directions, respectively.

3.1. Computational grid

We will choose a computational grid by defining two computational variables, X and Y, as follows:

$$X = x$$
$$Y = \frac{y-b(x)}{c(x)-b(x)} \tag{15}$$

from which:

$$Y_y = \frac{1}{c-b}, \qquad Y_x = Y_y[b'(Y-1)-c'Y] \tag{16}$$

(primes denote differentiations with respect to x). The grid will result as shown in Fig. 2. It is obviously not orthogonal between $-x_0$ and x_0. We will assume **n** to be the direction of the Y = constant lines in the grid. Consequently,

$$C = Y_y/\nu, \quad S = -Y_x/\nu, \quad \nu = (Y_x^2+Y_y^2)^{1/2} \tag{17}$$

$$\alpha_{ox} = -C^2[b''(Y-1)-c''Y-Y_x(c'-b')] \tag{18}$$
$$\alpha_{oy} = \nu^2/Y_y\,(c'-b')$$

$$\beta = U\alpha_{ox} + V\alpha_{oy} \tag{19}$$

$$F = a\,(U\alpha_{oy} - V\alpha_{ox}) \tag{20}$$

Also, note that

$$CY_y - SY_x = \nu, \quad CY_x + SY_y = 0 \tag{21}$$

Therefore,

$$VY_y + UY_x = v\nu \tag{22}$$

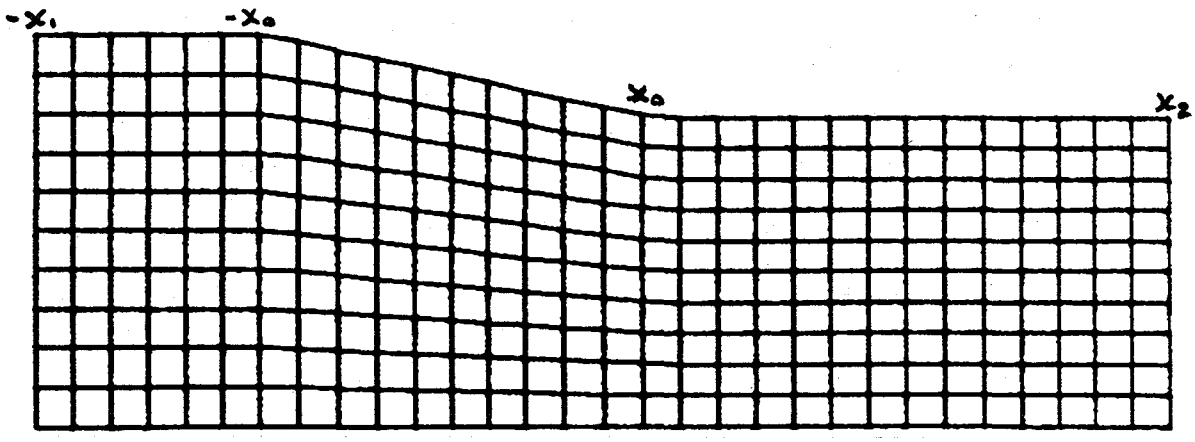

Figure 2.- Computational grid for example problem.

3.2. Generalized Riemann variables

In the spirit of Ref. 1, we introduce generalized Riemann variables in the form:

$$R_1^X = \frac{a}{\delta} + u, \quad R_2^X = \frac{a}{\delta} - u, \quad R_1^Y = \frac{a}{\delta} + v, \quad R_2^Y = \frac{a}{\delta} - v$$

and corresponding expressions for eight new coefficients:

$$\lambda_1^X = U+Ca, \quad \lambda_2^X = U-Ca, \quad \lambda_3^X = U, \quad \lambda_4^X = U-Sa, \quad \lambda_5^X = U+Sa$$

$$\lambda_1^Y = (v+a)Y_y/C, \quad \lambda_2^Y = (v-a)Y_y/C, \quad \lambda_3^Y = vY_y/C$$

Finally, the following terms are defined:

$$f_1^X = -\lambda_1^X(R_{1X}^X - as_X), \quad f_2^X = -\lambda_2^X(R_{2X}^X - as_X), \quad f_3^X = -\lambda_3^X(\frac{a_X}{\delta} - as_x)$$

$$f_4^X = -\lambda_4^X(R_{1X}^Y - as_X), \quad f_5^X = -\lambda_5^X(R_{2X}^Y - as_X), \quad f_6^X = -\lambda_3^X s_X$$

$$f_1^Y = -\lambda_1^Y(R_{1Y}^Y - as_Y), \quad f_2^Y = -\lambda_2^Y(R_{2Y}^Y - as_Y), \quad f_3^Y = -\lambda_3^Y u_Y$$

$$f_4^Y = -\lambda_3^Y s_Y$$

$$f_1^L = -2F, \quad f_2^L = 2\beta v, \quad f_3^L = -2\beta u$$

A new form of the equations of motion results:

$$s_t = f_6^X + f_4^Y \tag{23}$$

$$a_t = \frac{\delta}{2}(f_1^X + f_2^X + f_1^Y + f_2^Y + f_4^X + f_5^X - 2f_3^X + f_1^L) - \delta a s_t \tag{24}$$

$$u_t = \frac{1}{2}(f_1^X - f_2^X + 2f_3^Y + f_2^L) \tag{25}$$

$$v_t = \frac{1}{2}(f_1^Y - f_2^Y + f_4^X - f_5^X + f_3^L) \tag{26}$$

As pointed out in Ref. 1, in the new formulation there are terms (denoted by f_i^X) which contain only X-derivatives, terms (denoted by f_i^Y) which contain only Y-derivatives, and local terms (denoted by f_i^L). Such a splitting provides a clear definition of the points to be taken into account in the discretization of derivatives. According to the arguments exposed in Ref. 1, each X- or Y-derivative is to be approximated using only two points. One is the point to be updated, the other is the point on the X- or Y-axis, respectively, near the point to be updated, on the side from which the line $X=\lambda_i^X t$ or $Y=\lambda_i^X t$ proceeds. Specifically, for the X-terms, the point to be used lies to the left of the point to be updated if $\lambda_i^X>0$, and to the right of it if $\lambda_i^X<0$. For the Y-terms, the point to be used lies below the point to be updated if $\lambda_i^Y>0$, and above it if $\lambda_i^Y<0$.

4. Two-level integration scheme

To avoid unnecessary branching options (which are time consuming on the computer), we will extend the computational region by one additional row of points all around the boundary. This will allow certain derivatives to be approximated by differences taken from outside, without violating any physical principle since such terms are computed but never used, as explained in the next Section.

At all points, the values of a, u, v and s are updated using a two-level scheme, formulated as follows:

4.1. Predictor level.

All values of f_i^X, f_i^Y, and f_i^L are computed, approximating any X-derivative according to the formula:

$$\phi_X \approx \frac{\phi_{j+1,m} - \phi_{j,m}}{\Delta X}$$

where

$$j = n \quad \text{if} \quad \lambda_i^X < 0$$
$$j = n-1 \quad \text{if} \quad \lambda_i^X > 0$$

and any Y-derivative according to the formula:

$$\phi_Y \approx \frac{\phi_{n,j+1} - \phi_{n,j}}{\Delta Y}$$

where

$$j = m \quad \text{if} \quad \lambda_i^Y < 0$$
$$j = m-1 \quad \text{if} \quad \lambda_i^Y > 0$$

All coefficients are approximated accordingly as

$$\lambda^X \approx \frac{1}{2}\left[\lambda^X_{j,m} + \lambda^X_{j+1,m}\right]$$
$$\lambda^Y \approx \frac{1}{2}\left[\lambda^Y_{n,j} + \lambda^Y_{n,j+1}\right]$$

Note that all terms are defined according to their physical domains of dependence at all points.

At nodes bracketing a transition from $\lambda_2^X<0$ to $\lambda_2^X>0$, λ_2^X is made equal to the local value, $\lambda^X_{n,m}$. This device (which can be

applied without using IF-statements) prevents the formation of expansion shocks. In the presence of shock, similar devices must be used in their vicinity. They will be explained in Section 6.3.

All values of f_i^X, f_i^Y and f_i^L so defined at the beginning of the computational step will be denoted by $f^{(k)}$ for brevity. They will be used to advance half a step using (23),(24),(25) and (26) in conjunction with

$$a^{k+1/2} = a^k + \sum f^{(k)} \Delta t/2 \qquad (27)$$

and similar formulas.

4.2. Corrector level.

All f's are recomputed using the values updated by (27) at half step. Then, in order to maintain the proper domain of dependence at the corrector level, we denote the new f's by $\tilde{f}$'s and correct them using the formulas:

$$f^{X(k+1/2)} = \tilde{f}^X - \frac{1}{2} f^{X(k)}_{2j-n+1,m}$$

$$f^{Y(k+1/2)} = \tilde{f}^Y - \frac{1}{2} f^{Y(k)}_{n,2j-m+1}$$

$$f^{L(k+1/2)} = \tilde{f}^L - \frac{1}{2} f^{L(k)}_{n,m}$$

Finally, formulas similar to (27) are applied:

$$a^{k+1} = a^{k+1/2} + \sum f^{(k+1/2)} \Delta t \qquad (28)$$

A remark is in order as regards f_4^X and f_5^X (typical contributions due to the non-orthogonality of the grid). They appear in two combinations only: $g=f_4^X+f_5^X-2f_3^X$ and $h=f_4^X-f_5^X$. With U>0 and U>Sa (a most likely occurrence), all derivatives in g and h are approximated by upstream differences. at the corrector level, only g and h (both computed at n-1) are needed in the updating of a and v, respectively. If U<Sa, f_4^X is approximated by downstream differences. We will make use of these remarks in Sections 5 and 6.1.

5. Boundary conditions

Boundary conditions must be applied at four boundaries: 1) lower wall, 2) upper wall, 3) entrance, and 4) exit. On the walls, no special provision has to be made for the f^X terms or for the f^L terms. As regards the f^Y terms, note that v=0 on both lower and upper wall. Therefore, $f_3^Y = 0$. All local terms also

vanish. On the upper wall, f_1^Y is computed using information from inside the flow; therefore, it is correct; f_2^Y, instead, must be defined using the condition, $v_t=0$. On the lower wall, the opposite is true. Consequently, f_2^Y will be determined on the upper wall by the equation

$$f_2^Y = f_1^Y + f_4^X - f_5^X \tag{29}$$

which descends from (26), and f_1^Y will be determined on the lower wall by

$$f_1^Y = f_2^Y \tag{30}$$

which also descends from (26), with the additional, obvious consideration that $f_4^X - f_5^X = 0$ (since the wall lies on the x-axis and $S=0$, $\lambda_4^X = \lambda_5^X$, $R_1^Y = R_2^Y$).

At the entry and exit boundaries, the local terms vanish, because the walls are parallel. On the entry boundary the flow may be supersonic or subsonic. If it is supersonic, all values can be prescribed, and all f-s vanish identically. If the flow is subsonic, we assume that $v=0$ and, consequently, $f_3^Y=0$ at the left boundary. The entropy is constant in the impinging flow (to the left of $-x_1$); therefore, $f_6^X = f_4^Y = 0$. Moreover, $\lambda_3^X = \lambda_4^X = \lambda_5^X$; consequently,

$$g = f_4^X + f_5^X - 2f_3^X = 0, \qquad h = f_4^X - f_5^X = 0 \tag{31}$$

The impinging flow has a clearly defined total temperature. It is convenient, thus, to impose that the total temperature is also constant at $x=-x_1$ all across the boundary. From the definition of total speed of sound,

$$a_c^2 = a^2 + \delta q^2 \tag{32}$$

written in the form:

$$a\, a_t + \delta\, q\, q_t = 0 \tag{33}$$

and the assumption that $v=v_t=0$ at $x=-x_1$, (24) and (25) with the first of (31) yield:

$$(a+u)f_1^X + (a-u)f_2^X + a(f_1^Y + f_2^Y) = 0 \tag{34}$$

from which f_1^X can be evaluated. Note that any value can be used for f_4^X, provided that f_3^X and f_5^X are defined to satisfy (31). Such a procedure is correct because only the values of g and h computed at the boundary are used, at the corrector level, to correct the

predicted values on the line next to the boundary.

On the right boundary, if the flow is supersonic, no special provision is to be taken (all X-derivatives are automatically approximated by upstream differences). If the flow is subsonic, we will simply impose that the pressure, p_{ex}, is constant in time. Using (6), the condition is found:

$$a_t/\delta - a\, s_t = 0 \tag{35}$$

Note that g=0 at the exit, where S=0. Therefore, we can apply (35) after setting g equal to zero, and, from (35), we find the condition,

$$f_1^X + f_2^X + f_1^Y + f_2^Y = 0 \tag{36}$$

which allows f_2^X to be evaluated from known quantities.

5.1. Resetting.

Terms, such as f_2^Y in (30), re-evaluated at the boundaries, are used in (27) and (28) to update the boundary points themselves. In principle, v, originally equal to zero on the walls, should remain equal to zero because (30) assures the vanishing of v_t. Similarly, a_c and p_{ex} should remain constant in force of (34) and (35). In practice, it may not be so because the updating of v, a_c and p_{ex} is affected by almost imperceptible truncation errors in time. After a number of steps, however, one can observe a departure from the original values, producing an increase or decrease in total energy and/or a non-vanishing v (expressing an addition or loss of mass through the wall). It is necessary, therefore, to reset certain quantities to maintain a_c, p_{ex} and v constant at the entry, exit and wall boundary, respectively. The task is accomplished, at $x=-x_1$, by computing

$$R_2^X = a/\delta - u \tag{37}$$

from the available values of a and u and using it, together with (32), to obtain:

$$u = \frac{1}{\delta+1}\left[-\delta R_2^X + \left(\frac{\delta+1}{\delta}\, a_c^2 - \delta (R_2^X)^2\right)^{1/2}\right] \tag{38}$$

Then, a follows from (37). At $x=x_2$,

$$a = \sqrt{\gamma \exp\left[\delta(\ln p/\gamma + s)\right]} \tag{39}$$

and u is recomputed from the definition of R_1^X:

$$u = R_1^X - a/\delta \tag{40}$$

Finally, at the upper wall,

$$R_1^X = a/\delta + v \tag{41}$$

is recomputed from the available values of a and v and the condition, v=0 is used to obtain a new value for a:

$$a = \delta R_1^Y \tag{42}$$

U and V are obtained from (14), with v=0 and u accepted as computed.

6. Shocks

The analysis of a flow containing shocks cannot be made using the technique described in Sections 3 through 5, as is. Indeed, the equations of motion written in the form (8) assume that the entropy is carried by the moving flow elements. Therefore, such equations cannot account for the rise in entropy occurring at shocks. Oblique shocks with low normal Mach numbers can be properly captured by the technique. If normal shocks or shocks followed by a subsonic region occur, additional care must be taken to handle them correctly. Other techniques [2,3,4], seem to reach the goal of capturing all kinds of shocks. This is done by using at every point the same computational scheme which would be needed at shock points only. In the present approach, the technique for all points remains the same as described above. All shock points, however, must be tracked and the Rankine-Hugoniot conditions used to update the grid points downstream of a shock.

The updating of a shock point is very simple and straightforward, once the direction of the normal to the shock is known because, along the normal, the one-dimensional technique [5] can be used. Finding the normal to a shock at any shock point, however, requires a tracking of all the shocks, which involves a certain amount of logic and bookkeeping. We will begin with discussing the updating at any shock point, assuming that the direction of the normal thereon is already known.

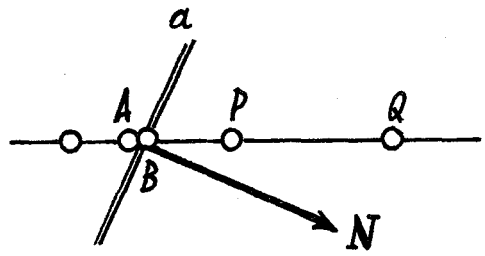

Figure 3.- Shock wave passing through Y=constant line.

6.1. Updating a shock point.

Let a (Fig. 3) be a shock wave, separating a low-pressure region from a high-pressure region. Let A and B be two points, on a Y=constant line, on either side of the shock. We know their (common) position, as well as the direction of the normal, defined by the unit vector, **N**. Both at the predictor and at the corrector level, the calculation of the shock follows the updating of all grid points, made following the procedure detailed in Section 4 (with a few provisoes to be explained in Section 6.3).

From now on, we follow the analysis exposed in Ref. 5, Section 9 for normal shocks. Let N_1 and N_2 be the cosine and sine of the angle between **N** and **n**, and

$$\tilde{u} = q \cdot \mathbf{N} = uN_1 + vN_2 \tag{43}$$

the component of the velocity normal to the shock; let us call $\tilde{v}$ the tangential component,

$$\tilde{v} = - uN_2 + vN_1 \tag{44}$$

The parameter, Σ, defined by (30) in Ref. 5, is redefined here as

$$\Sigma = \delta\,(R^X_{2B} + u_A)/a_A = [a] + (\delta/a_A)[u] \tag{45}$$

where [] denotes a jump across the shock. Note that

$$[u] = [\tilde{u}]\, N_1 \tag{46}$$

since $[u]=[\tilde{u}N_1-\tilde{v}N_2]$ from (43) and (44), but $[\tilde{v}]=0$. The normal relative Mach number of the shock is

$$M = \frac{\tilde{u}_A - W}{a_A} \tag{47}$$

where **WN** is the velocity of the shock as a locus of discontinuities. From the Rankine-Hugoniot conditions we obtain:

$$\tilde{u}_B = \tilde{u}_A + a_A \frac{1-M^2}{(1+\delta)M} \tag{48}$$

$$a_B = a_A \frac{\sqrt{(\gamma M^2-\delta)(1+\delta M^2)}}{(1+\delta)|M|} \tag{49}$$

$$s_B = s_A + \frac{1}{2\gamma\delta}\left[\ln \frac{\gamma M^2-\delta}{1+\delta} + \gamma \ln \frac{1+\delta M^2}{(1+\delta)M^2}\right] \tag{50}$$

$$\tilde{v}_B = \tilde{v}_A \tag{51}$$

From the last term in (45), using (48) and (49), we obtain

$$\Sigma = \frac{1}{(1+\delta)M}\left[\sqrt{(\gamma M^2-\delta)(1+\delta M^2)} + \delta(M^2-1)N_1\right] \qquad (52)$$

From a given Σ, M is thus obtainable. Since (52) cannot be inverted, it is convenient to proceed as follows. With

$$g_1 = \sqrt{(16\gamma-\delta)(1+16\delta)}]/[4(1+\delta)] \ , \quad g_2 = (15\delta)/[4(1+\delta)] \qquad (53)$$

$$\Sigma_0 = g_1 + g_2 N_1 \qquad (54)$$

a first guess for M is written in the form:

$$M = G_2+G_1\Sigma, \quad G_1 = 3/(\Sigma_0-1), G_2 = 1-G_1 \qquad (55)$$

and the approximation is improved iteratively, until Σ gets stabilized to within a given tolerance (0.00001 on an IBM 370, 0.000001 on a CRAY 1). Three to five iterations are sufficient. The shock velocity is then obtained from (47), and the Rankine-Hugoniot conditions can be applied to update the values at B. The abscissa of the shock point is obtained by adding

$$\Delta x_s = W\,\Delta t\,\frac{\cos\alpha_0}{N_1} \qquad (56)$$

to its current value.

With different degrees of sophistication, we can assume that A and B coincide with the grid nodes bracketing the shock point (an acceptable approximation if the grid is very fine) or extrapolate upstream values on the same Y=constant line to define values at A and downstream values to define initial values at B. Similarly, we can interpolate from B and the second node downstream of the shock (Q in Fig. 3) to get final values at the first node downstream (P in Fig. 3).

The shock calculation is now completed. At the predictor level, however, we must re-evaluate f_1^X, f_3^X, f_5^X and f_6^X at P, because these quantities are needed at the corrector level for point Q (f_4^X can be left unchanged, since only g and h, as defined by (31), are used). The re-evaluation is possible using the actual increments of a, u and s in time and (23), (25), (26) and (24) to determine f_6^X, f_1^X, f_5^X and f_3^X, in that order.

6.2. Formation and tracking of shocks.

We opted for not attempting any tracking of oblique shocks with transitions from supersonic to supersonic flow. In the present class of problems, such shocks have a normal Mach number very close to 1. Therefore, the entropy increase across the shock is minimal, and it can be neglected. Moreover, the shock is not influenced by downstream conditions. Therefore, we may assume that it finds its location correctly. In conclusion, only the formation of shocks with

transition from supersonic to subsonic flow is to be monitored.

Testing on shock formation begins only when there is at least one grid cell across which λ_z^X changes from positive to negative. Each Y=constant line is swept from left to right, searching for a possible cell where λ^X_z changes from positive to negative (and which has not been marked as a shock already). The node to the left of the cell is marked as a possible A-point of a shock. An approximate initial location of the shock is assumed in the cell, where $\lambda_z^X=0$. New shock points are not generated if old shock points already exist along the same Y=constant line within two cells on either side. Once a shock point has been generated, the testing proceeds along the same Y=constant line until the entire line has been monitored.

All shock points so generated over the computational grid are stored in a one-dimensional array. The next step consists of grouping the shock points in successive links, so that each link contains shock points arranged in the order of increasing Y and the distance between two successive points of the same link is not greater than one cell.

Once there is more than one point in one link, the direction of the normal at a point is easily found. At a wall, the normal to the shock is forced to be parallel to the wall. Otherwise, the normal is evaluated by using centered differences along the shock.

If a shock point moves to an adjacent cell downstream, the values upstream are extended to the new upstream grid node; similarly, if the shock point moves to an adjacent cell upstream, the new downstream grid node is given the most recent values at the B-point. Such updatings are made at the end of the corrector level only; consequently, there is no need for re-evaluation of f-terms.

6.3. Accuracy of the shock calculation.

There is no doubt that the present technique of shock fitting is both fast and very accurate. Speed is achieved because the calculation at any node remains the same as for a shockless flow, plus a minimal amount of additional work, as described in Section 6.1, performed at the shock points only. Even more important is the accuracy of the results.

The entire calculation of the shock depends on two parameters, the shock angle and Σ. The angle is conveniently defined by centered differences, based on a shock geometry which is clearly described, regardless of the number of shocks and their relative positions. The other parameter, Σ, as defined by (45), contain two variables computed at point A, and one computed at point B. Let us consider first the case of Fig. 4, where three consecutive shock points remain within the same grid column. Since the flow is supersonic at A, all the f^X-s are computed from upstream, correctly. All the f^Y-s are

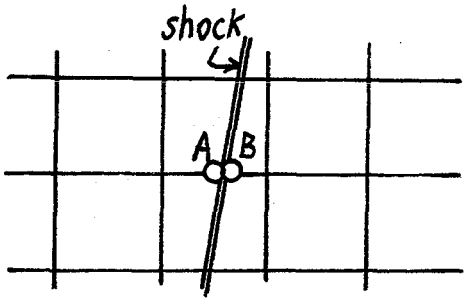

Figure 4.- Nearly vertical shock wave.

computed as in any ordinary point, correctly. Therefore, the values of u_A and a_A are correct. If we single out a_B and u_B, which concur to form R^X_{2B}, we see that neither a nor u are correct, because they depend on f_1^X which is improperly computed across the shock; nevertheless, the combination R^X_{2B} is correct, because the contribution of f_1^X cancels out (the contribution of $f_4^X+f_5^X-2f_3^X$ is minimal and can be set equal to zero).

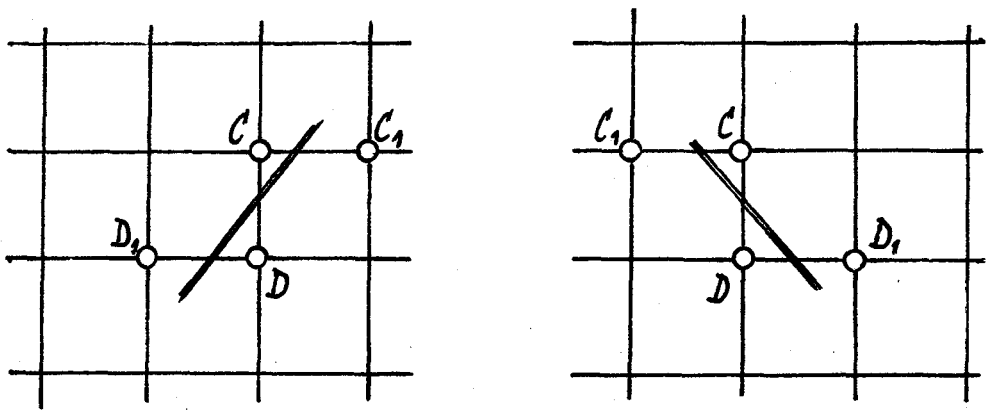

Figure 5.- Shock waves passing through X=constant lines.

In the case of Fig. 5, the term, f_1^Y is evaluated at C using a local value of λ_1^Y and R_1^Y at D_1 instead of at D, and the term, f_2^Y is evaluated at D using a local value of λ_2^Y and R_2^Y at C_1 instead of at C. Such devices can be loosely interpreted by saying that the derivatives are taken after subtracting the jumps across the shock. It is interesting to note that none of these devices requires the use

of IF-statements.

7. A numerical example.

From the several cases analyzed to test the present technique, we chose one which has some interesting features and the results of which can be compared to calculations by other authors. The geometry of the duct is shown in Fig. 1. The values of x_1, x_2, x_0 and b_0 are

$$x_1 = 1.5$$
$$x_2 = 2.5$$
$$x_0 = 0.66937$$
$$b_0 = 1.26775$$

Therefore, the wedge has a 20% slope. The Mach number at the entry section is 1.6. The calculation has been made starting from rest impulsively (this corresponds to the third model of Section 2, with an infinite acceleration). Two runs have been made, one with a 50 by 20 mesh, and another with a 100 by 30 mesh. In both runs, the CFL number has been taken as 1. In these runs, lengths have been scaled to the exit width, and speeds to the speed of sound at the entrance, divided by $\sqrt{\gamma}$. Therefore, times are scaled to the ratio between the width and the speed of sound at the entrance, multiplied by $\sqrt{\gamma}/b_0$.

The figures presenting results for these runs are self-explanatory. In Fig. 6, four sets of isomachs are shown, at increasing times, together with the shape of the shocks which have been fitted. Note the appearance of a second shock of very low strength, in the latter stages of the flow evolution. Note also the good resolution and precision of details, despite the use of a mesh which is considered rather coarse in the existing literature. Better results, but not so much better as to justify the effort, are obtained by using a finer mesh, as shown in the second set of plots, Fig. 7. In Fig. 8, a typical plot of entropy is shown, at the same step as the final plot of Fig. 7. Finally, in Fig. 9 we reproduce some figures from Ref. 4. Such figures have been obtained using a 150 by 50 mesh. We believe that the comparison shows that the present technique is more efficient.

The code used so far has not been optimized; no attempt has been made yet to fully exploit the advantages of vectorization. The computational time required on a CRAY 1 machine is of the order of 20 seconds for 400 steps, using a 50 by 20 mesh.

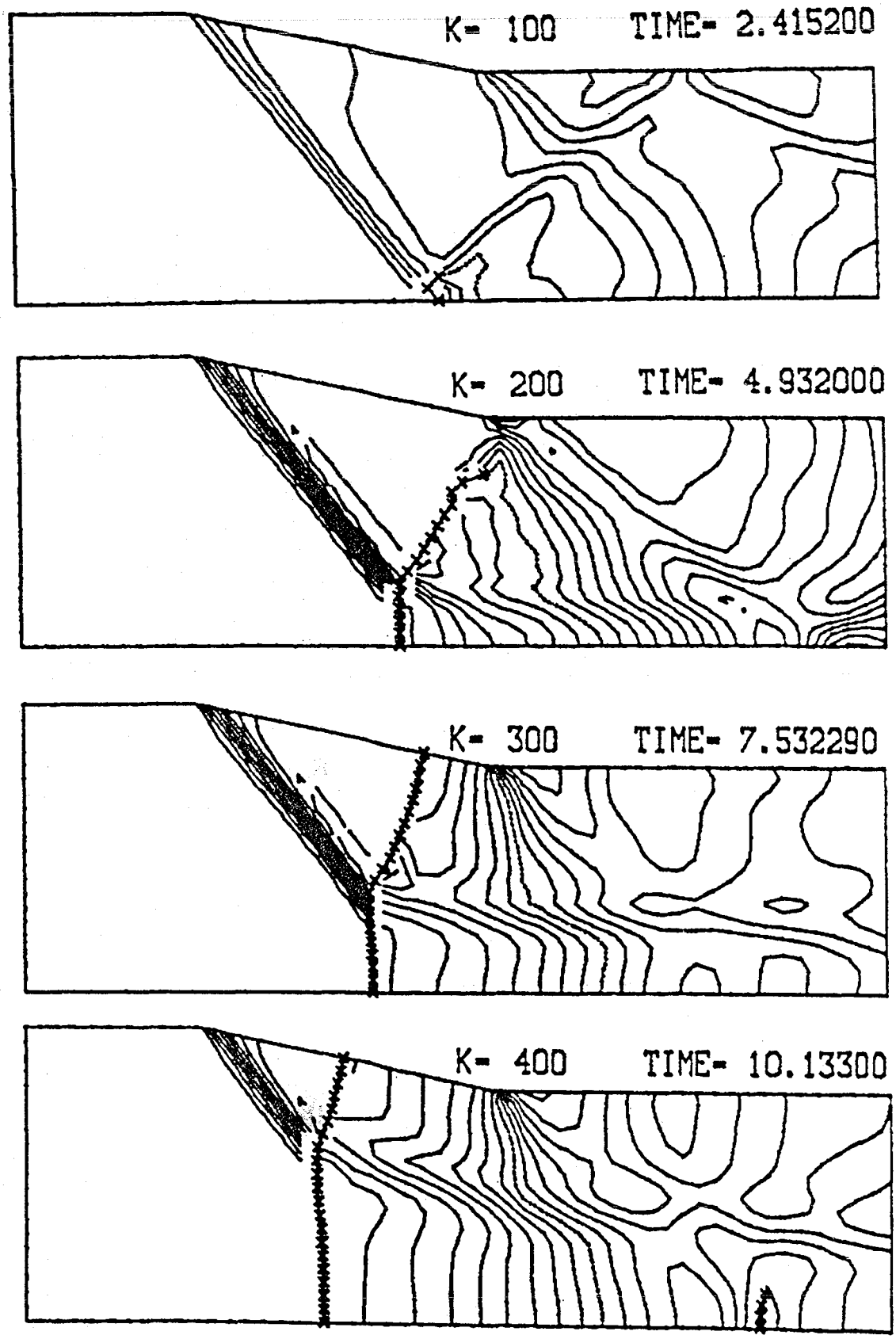

Figure 6.- Isomachs at four times for 50 by 20 mesh.

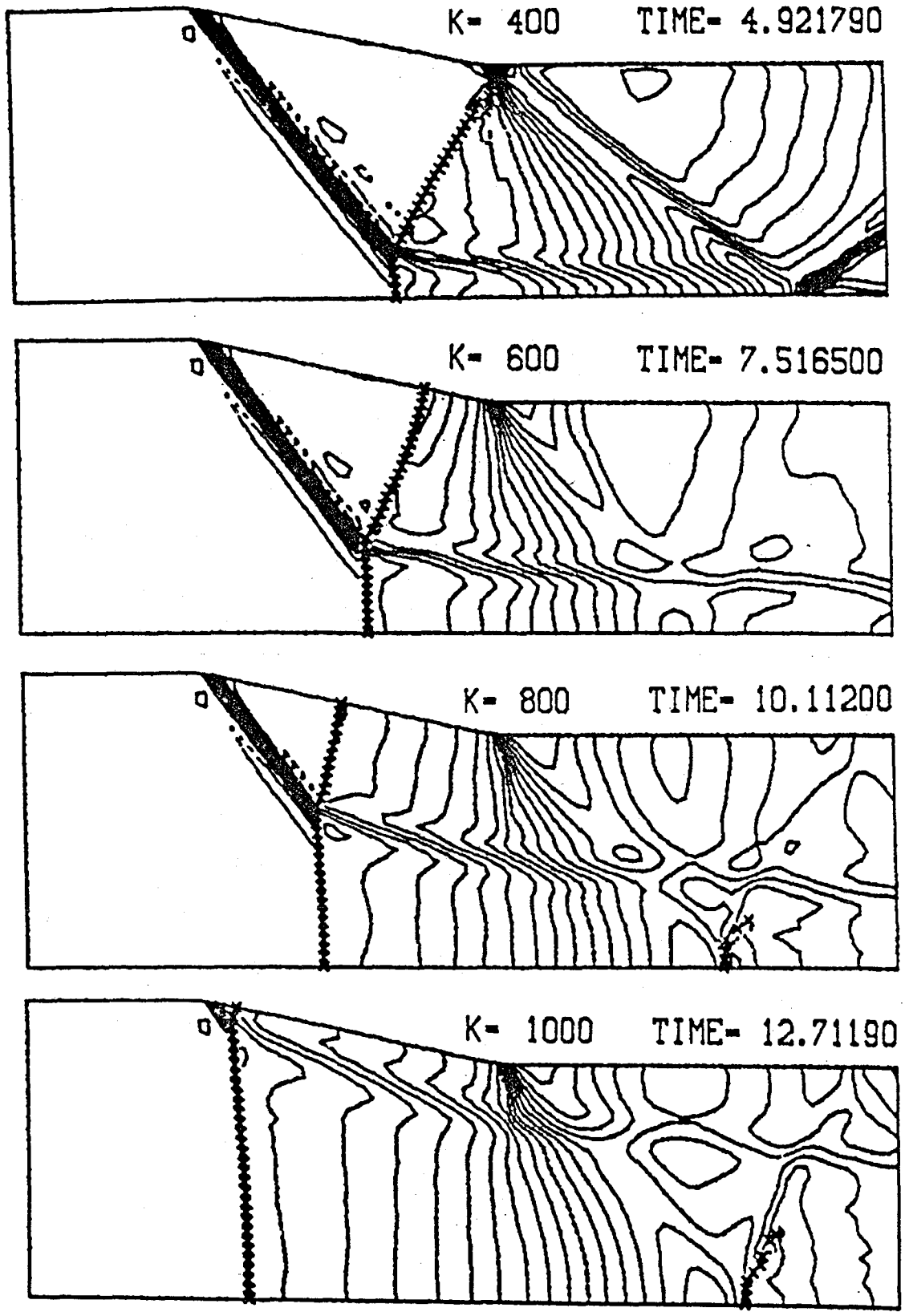

Figure 7.- Isomachs at four times for 100 by 30 mesh.

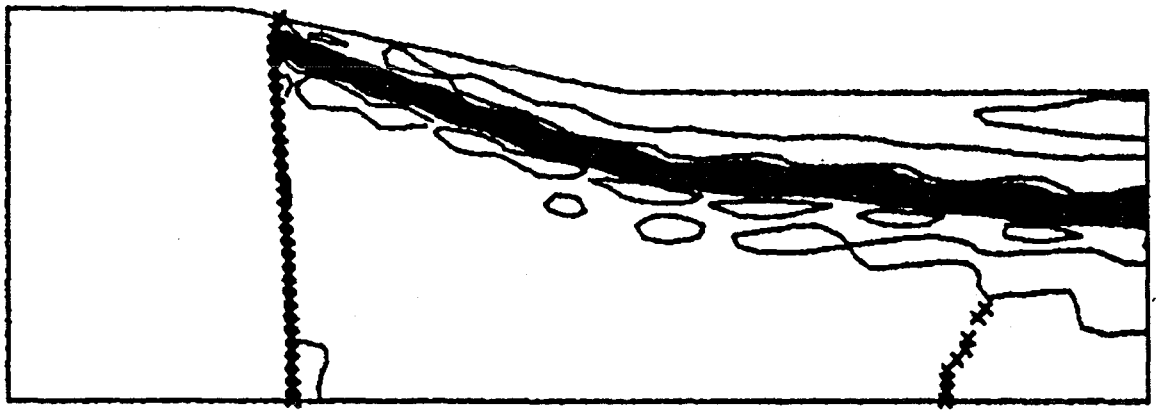

Figure 8.- Isentropes at time = 12.71190 for 100 by 30 mesh.

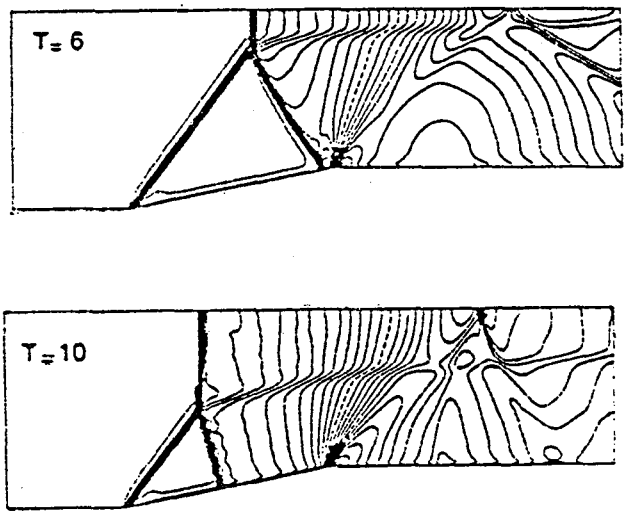

Figure 9.- Isomachs from reference 4 for same problem, 150 by 50 mesh.

REFERENCES

1. Moretti, G.; and Zannetti, L.: A New and Improved Computational Technique for Two-Dimensional, Unsteady, Compressible Flows. AIAA J., vol. 22, no. 6, pp. 758-765, 1984.

2. van Leer, B.: Towards the Ultimate Conservative Difference Scheme, IV. A New Approach to Numerical Convection. J. Comp. Phys., vol, 123, pp. 276-299, 1977.

3. Osher, S.; and Solomon, F.: Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws. Math. of Comp., vol. 38, pp. 339-377, 1982.

4. Montagne, J. L.: A Second-Order Accurate Flux Splitting Scheme in Two-Dimensional Gas Dynamics. Lecture Notes in Physics, vol. 218, pp. 406-411, 1985.

5. Moretti, G.; and Dipiano, M. T.: An Improved Lambda-Scheme for One-Dimensional Flows. NASA CR-3712, 1983.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3930		

4. Title and Subtitle	5. Report Date
Numerical Studies of Two-Dimensional Flows	September 1985
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Gino Moretti	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
GMAF, Inc. P.O. Box 184 Freeport, NY 11520	NAS1-17262
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code 505-31-03-02

15. Supplementary Notes

Langley Technical Monitor: J. C. Townsend
Final Report

16. Abstract

A formulation of the lambda-scheme for the analysis of two-dimensional inviscid, compressible, unsteady transonic flows is presented. The scheme uses generalized Riemann variables to determine the appropriate two-point, one-sided finite-difference approximation for each derivative in the unsteady Euler equations. These finite differences are applied at the predictor and corrector levels with shock updating at each level. The weaker oblique shocks are captured, but strong near-normal shocks are fitted into the flow using the Rankine-Hugoniot relations. This code is demonstrated with a numerical example of a duct-flow problem with developing normal and oblique shock waves. The technique is implemented in a code which has been made efficient by streamlining to a minimal number of operations and by eliminating branch statements. The scheme is shown to provide an accurate analysis of the flow, including formation, motions, and interactions of shocks; the results obtained on a relatively coarse mesh are comparable to those obtained by other methods on much finer meshes.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
computational fluid dynamics finite-difference methods Euler equations unsteady flow transonic flow two-dimensional flow	Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	22	A02

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1985

End of Document