Efficient Euler Solver with Many Applications

Gino Moretti*
G.M.A.F., Inc., Freeport, New York

A computational technique for two-dimensional unsteady Euler equations, based on a λ-formulation and explicit shock fitting, is presented. Any number of shocks, of any shape and type, and their interactions can be treated by this technique, which does not require complicated logic and does not perform redundant calculations. The code is fast and the results are very accurate. Examples (transonic airfoils, shocks in ducts, intake flows, multiple Mach reflections) are presented and discussed.

Generalities About Shock Fitting

THE superiority of shock fitting over shock capturing in one-dimensional computations is above controversy. For two- and three-dimensional flows, however, shock fitting has always been considered impractical. The object of the present paper is to show that two-dimensional calculations with shock fitting can be done, are easy to code, and produce excellent results. Shock-fitting codes are simpler, faster, more accurate, and require fewer mesh nodes than shock-capturing codes.

Riemann Variables

Let a, u, p, s, and γ mean speed of sound, velocity, pressure, entropy, and ratio of specific heats, respectively, and let $\delta = (\gamma - 1)/2$. In one-dimensional flows, certain combinations of the speed of sound and velocity, such as $R = a/\delta \pm u$, may be called Riemann variables, only because in simple waves, where they remain constant along certain lines, they are commonly known as Riemann invariants. In nonisentropic flows, depending on any number of space variables, combinations of increments, such as

$$\Delta R - a\Delta S \tag{1}$$

are physically equivalent to combination of increments, such as

$$\Delta(\ell n\, p) \pm (\gamma/a)\Delta u \tag{2}$$

which express the propagation of pressure signals along generators of the characteristic controls in space and time. The term, "generalized Riemann increments" is used here for any expression such as Eq. (1).

Importance of the Riemann Variables in Shock Fitting

A shock fitting code should take full advantage of the fact that the value of one Riemann variable on either side of the shock is independent of its value on the opposite side. Such a variable, indeed, can be computed on either side of the shock without using any information from the other side; in other words, without approximating derivatives by differences taken across a shock. This eliminates all wiggles in a second-order-accurate technique without forcing the order of accuracy to be lowered locally, or introducing artificial viscosities. In mathematical terms, the convergence-in-the-mean of undamped shock-capturing codes is replaced by uniform convergence of algorithms, separately valid on each of the two regions divided by the shock. As stated on repeated occasions, the argument is consistent with the convergence-in-the-mean of the Fourier expansion of a function with jumps, and its uniform convergence when the jumps are removed.

Connection with the λ-Formulation

Not by accident, the Riemann variable that proves to be so useful in the application of shock fitting is one of the basic variables in the current version of the λ-formulation.[1] And since the same variables are the best choice for a proper handling of boundary conditions, we conclude that the most homogeneous and natural way of computing is a combination of the λ-formulation and shock fitting.

Two-Dimensional Shock Fitting

The simplicity of shock fitting is obvious in one-dimensional problems. The shock depends on its environment, through that one Riemann variable R, which can be computed correctly on both sides of the shock without having to use any information from the shock itself. In fact, the relative shock Mach number M is a single-valued function of the jump in R and, knowing M, the Rankine-Hugoniot conditions can be applied to determine all downstream values. Specifically, if we identify the two sides of the shock with the two grid nodes bracketing it, a one-dimensional calculation of a shock requires one statement to define ΔR, five statements to obtain M [solving the equation $f(\Delta R, M) = 0$ by iteration], and five more statements to get the shock velocity, to displace the shock accordingly, and to apply the Rankine-Hugoniot conditions.

Simplicity is not lost in two dimensions. The basic procedure, indeed, remains the same since the local calculation of a shock point is one-dimensional. The equations are only slightly altered by a slope factor, and one more Rankine-Hugoniot condition is needed.[1] We still need only as few as 12 statements to compute a shock point.

Topological Problem of the Shock Slope

Apparently, however, the evaluation of the shock slope may cause difficulties. Shocks move across a given grid and may have any slope with respect to the coordinate lines. Some portions of a shock are better defined by their intersections with one family of grid lines, other portions by their intersections with the opposite family. Moreover, to get the slope of a shock at any point, it seems that different shocks should be treated separately and, for each one of them, all its points should be stored in an orderly sequence. All this would make the logic of a complicated shock pattern extremely difficult; fortunately, a closer look shows that the problem can be solved in a much simpler way.

Let us begin by noting that we can store all the information pertinent to shock points in single arrays, the shock points being numbered for identification but not necessarily stored in

Received March 27, 1987; revision received Nov. 1, 1987.

*Consultant. Fellow AIAA.

any particular order (in other words, two shock points stored next to one another may be located at a large distance in the physical plane; conversely, two shock points located next to one another in space are not necessarily identified by two successive indices). This freedom of storage eliminates any need for complicated logics to reorganize shock points along shock lines, in a rigid sequence, as such points appear, move, or disappear.

To obtain the slope of a shock at one of its points, we must determine which other shock points, if any, are in its immediate neighborhood (that is, within a square of six grid cells). Here is a simple device for cross referencing shock points and grid points. To simplify the discussion, let us consider shock points located on one family of grid lines (for example, $Y =$ const lines; see Fig. 1). A double array, $JSX(N, M)$, is set equal to zero at all grid nodes ($X = N\,\Delta X$, $Y = M\,\Delta Y$). If a shock point occupying the Jth position in the shock storage (denoted by an x in Fig. 1) exists on the Mth line of the grid, between the Nth and the $(N+1)$th point, we will let $JSX(N+1, M) = J$. When computing shock point J, we explore the intervals marked by the heavy line a in Fig. 1, looking at the values of JSX at the grid points $(N, M+1)$, $(N+1, M+1)$, and $(N+2, M+1)$. If one of them contains a value of JSX different from zero (JP, say), we know that the shock occupying the JP location in the shock arrays is the neighbor of the Jth shock point from above. If more than one shock point exists on a, we say that the neighbor of J on the $(M+1)$th line is the baricenter of all such points. We also scan the heavy marked intervals HD, FB, EA, and GC of Fig. 1, searching for possible shocks located on $X =$ const lines; such points will be considered as upper neighbors of J and combined with the ones found on line a. We repeat the operation at the $(M-1)$th line on the b interval, and we scan the heavy marked intervals below H, F, E, and G to find possible points on $X =$ const lines, to be considered as lower neighbors of J. The slope can thus be computed using the coordinates of the upper and lower neighbors. If no shock is in the neighborhood of the Jth shock from above, the slope is computed by using the coordinates of J and the lower neighbor. We proceed in a similar way for the lower neighbor. If the Jth shock point is isolated, it must be dropped.

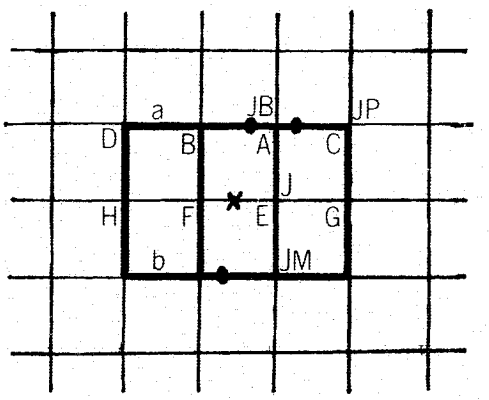

Fig. 1 Neighborhood of a shock point.

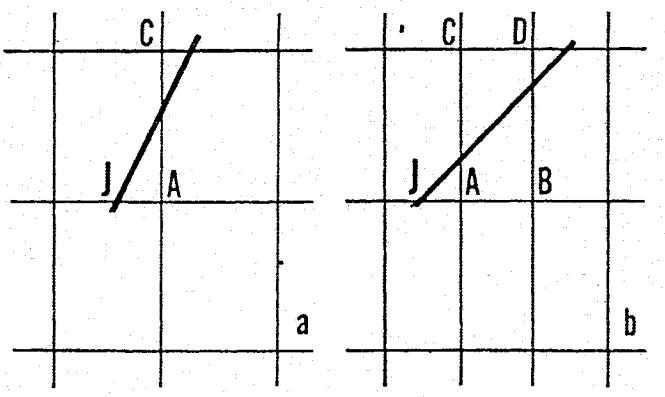

Fig. 2 Shocks crossing grid lines.

Transverse Crossings

Once the neighbors of a shock point have been identified as just outlined, the acquired knowledge can be used in two more instances. The first occurs when a shock, identified by two points on $Y =$ const lines, crosses one or more $X =$ const lines; in such a case, no Y derivatives at the nodes immediately above and below the shock (such as A, B, C, and D in Fig. 2) can be approximated by differences taken across it. Similarly, if a shock, identified by two points on $X =$ const lines, crosses a $Y =$ const line, no X derivatives at the nodes immediately to the left and to the right of the shock can be approximated by differences taken across it. Since such transverse crossings are pinpointed by the aforementioned analysis, we know where to act. As regards the action itself, it seems that approximating derivatives by differences taken on the "wrong" side suffices.

Shock Geometry Smoothing

The second instance in which a knowledge of the neighbors of a shock point is needed is related to the control of the shock shape. Wrinkled shocks, indeed, cannot exist. If a shock had a spike pointing to the left, the velocity vectors downstream would diverge and create a diffuser effect that would suck the spike toward the right, and vice versa.[2] In a numerical calculation, however, wrinkles on a shock may occur because of a faulty evaluation of the initial location of a shock point. The effect is particularly evident when the environment does not force the shock to move. If centered differences are used to determine the slope of the shock, the wrinkle is not detected, the physical mechanism of geometry smoothing is not interpreted numerically, and the wrinkles cannot be eliminated. On the other hand, choosing the proper side for one-sided differences in a complicated shock pattern would again require a major logical effort. A simple, physically consistent numerical smoothing of the shock geometry can be implemented by modifying the shock Mach number in order to account for the local curvature of the shock.[3]

Structure of the Computational Code

Consequently, a computational step is structured as follows. Prior to computing shocks, all nodes in the computational mesh are computed without special provisions. A scheme as simple as the λ-scheme can be used to update all nodes with second-order accuracy, using a vectorized code. This part of the code uses standard "predictor," "corrector," and "boundary" subroutines.[1] Next, the array of shock points is scanned (we repeat this, examining the points in the order in which they are stored, not in any geometrical or physical order). The object of the operation is to remove all results obtained using differences taken between points lying on opposite sides of any shock and to replace them as previously outlined. Therefore, at all grid nodes in the neighborhood of a shock, derivatives that would naturally be approximated by differences across the shock, are forced to be approximated by differences in the opposite direction. The updated values at nodes neighboring shocks are corrected accordingly. This is accomplished in a "neighborhood" subroutine. Then, all jumps across shocks are evaluated, together with the shock velocities. The shock Mach numbers are obtained as outlined above in a "shock" subroutine. Finally, a subroutine is called that performs many tasks:

1) Moving shock points according to their velocities.

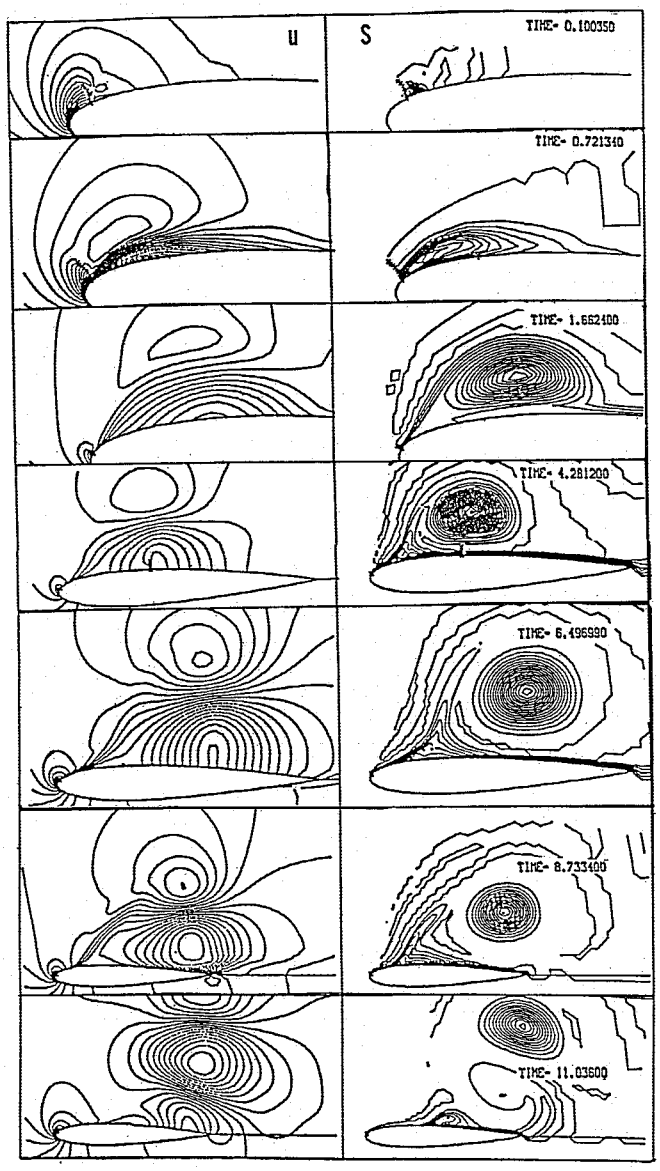

Fig. 3 Constant u lines and constant S lines about a NACA 0012 airfoil, unsteady flow.

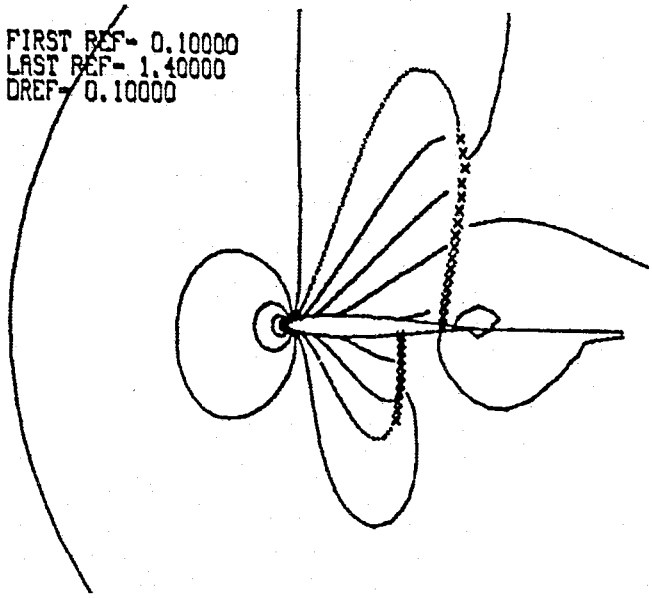

Fig. 4 Steady flow about a NACA 0012 airfoil, $M_\infty = 0.85$, $\alpha = 1$ deg.

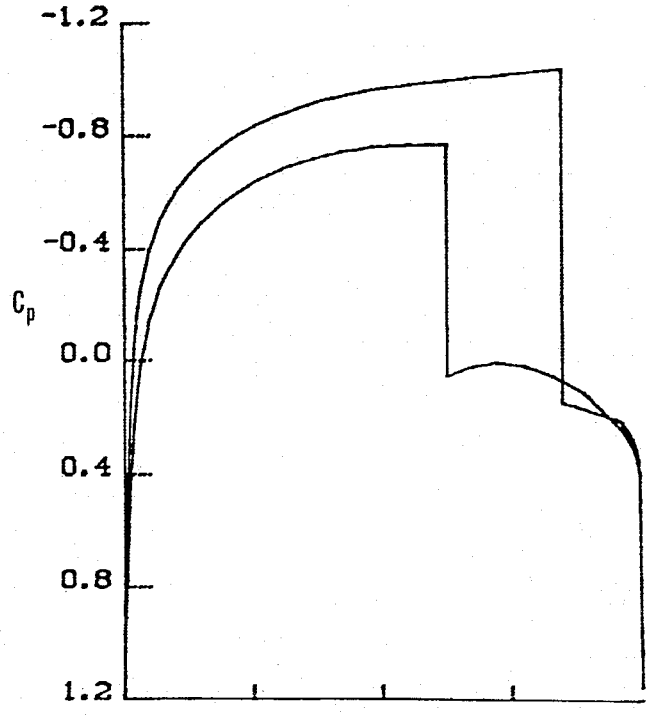

Fig. 5 C_p about a NACA 0012 airfoil, $M_\infty = 0.85$, $\alpha = 1$ deg.

2) Resetting values at a grid point, if the shock crosses over it in going from one cell to a neighboring one.

3) Searching for the formation of new shock points.

4) Dropping one shock point when there are two in the same cell.

5) Evaluating the slope of the shock at all shock points.

6) Dropping isolated points.

7) Dropping shock points on X lines, if their slope is less than 45 deg and shock points on Y lines, if their slope is greater than 45 deg.

Computed Examples

In the rest of the paper, I present some results of calculations of flowfields with shocks, at different degrees of difficulty, with brief comments.

Transonic Airfoils

Transonic flows about airfoils, with imbedded shocks, are apparently the easiest problem for various reasons. Only shocks of one family exist, and they do not interact with each other. The shocks are almost parallel to grid lines. Shock points tend to stack up from the airfoil into the freestream in an orderly fashion. Orthogonal meshes can be used. The flows of current interest are steady. The shocks always separate an upstream supersonic flow from a downstream subsonic flow; therefore, they are easily detected by looking for changes in the sign of $(u - a)$. Moreover, the Riemann variable of interest is exactly defined on both sides of the shock. Nevertheless, many interesting details require a careful numerical treatment, if one searches for a solution with a high degree of accuracy.

1) An imbedded shock, even if almost parallel to a family of grid lines, is not necessarily confined to a single strip of the grid. As a transverse crossing of the mesh occurs, the difficulty just mentioned is present here as in any more complicated problem.

2) The necessity of having the calculation converge to a steady solution complicates the issues. Indeed, a shock point may tend to be located on a node of the grid, or at least very close to it. A minimal change in its location may shift it from one cell to a neighboring one. If we identify the two sides of the shock with the two grid nodes bracketing it, the crossing procedure presented in the previous section may produce an unphysical jump in the information used to recompute the shock. Consequently, the point can be pushed back into the

original cell and the oscillation can repeat itself ad infinitum. The remedy[2] consists of defining the values of the Riemann variable not at grid nodes bracketing the shock point but at the point itself, by extrapolating from either side; however, the extrapolation has to be defined in such a way that the values at the shock do not jump when the shock point moves from one cell to another.

3) Another problem arises at the tip of the shock, where the last physical shock point has a Mach number of 1. Such a point does not necessarily lie on a grid line, but this is not a cause of trouble since very weak shocks are well captured by the λ-scheme. It is thus convenient not to compute the last shock point as a shock but to locate it, by interpolation, where the flow Mach number is 1; the location of the point is then used to obtain the shock slope at the next shock point.

The technique of shock fitting, as just outlined in connection with time-dependent calculations for unsteady flows, has been applied to the transonic airfoil problem, both for unsteady calculations and as a way to obtain asymptotically steady solutions.[4] In Fig. 3, we present a sequence of flowfield patterns in a case where no steady solution exists (a NACA 0012 airfoil in a freestream with $M_\infty = 0.3$ at a 20 deg angle of attack). The grid used in the calculation is a C grid, and u is the velocity component along grid lines wrapped around the body and the wake. On the left-hand side of the figure, u = const lines are plotted; dotted lines correspond to negative values of u. On the right-hand side of the figure, S = const lines are plotted. A sequence of plots is presented for increasing values of time. A strong shock forms near the leading edge and it generates a strong vorticity. The shock is rapidly pushed upstream, and a recirculation bubble appears behind it. At $t = 1.66$, the shock has been pushed very far toward the stagnation point and it has lost both length and strength. Meanwhile, the bubble has grown bigger. The recirculation is now so violent that (at $t = 4.28$) a secondary shock appears in the reverse flow. The bubble is finally shed; the main shock is so weak that a new clean region of isentropic compression with positive velocities is formed ($t = 11$). The isentropic compression piles up into a new shock and at $t = 13$ the same pattern as at $t = 0.1$ appears and a new cycle begins.

To acquire steady solutions, the same technique of shock fitting has been inserted on a fast, implicit Euler solver.[5] Typical results are shown in Figs. 4 and 5 for a NACA 0012 airfoil at $M_\infty = 0.85$ and a 1 deg angle of attack. Figure 4 shows the isobar pattern about the airfoil, and Fig. 5 is the distribution of C_p on the upper and lower surfaces of the airfoil. The CRAY-XMP computational time for this case (using a C grid of 128×32 intervals) is less than 20 s.

Transonic Flows in Channels

Transonic flows in channels offer a different range of difficulties. According to the geometry of the channel, and the entry and exit conditions, various types of shocks may appear, including shocks similar to those seen on airfoils. An abrupt change in the slope of a wall may produce an oblique shock. Under steady boundary conditions, such a shock is steady and the flow is supersonic on both sides of it. The same wall geometry may generate a detached shock, followed by a transonic flow. All shocks may reach a wall and be reflected from it. The reflection may be regular or a Mach reflection; and one type of reflection can evolve into the other. This brings in various types of shock interactions. Finally, the computational mesh is not necessarily orthogonal.

We start with a case of relative simplicity. The channel (Fig. 6) has two semi-infinite parts, of different cross-sectional areas, connected by a straight transition. The flow entering from the left is supersonic and uniform. An impulsive start produces a perturbed field in the transition and beyond. The data of the problem can be adjusted to provide a variety of cases, from a pattern of steady, regular reflections to the formation of Mach reflections and an unsteady choking of the channel.

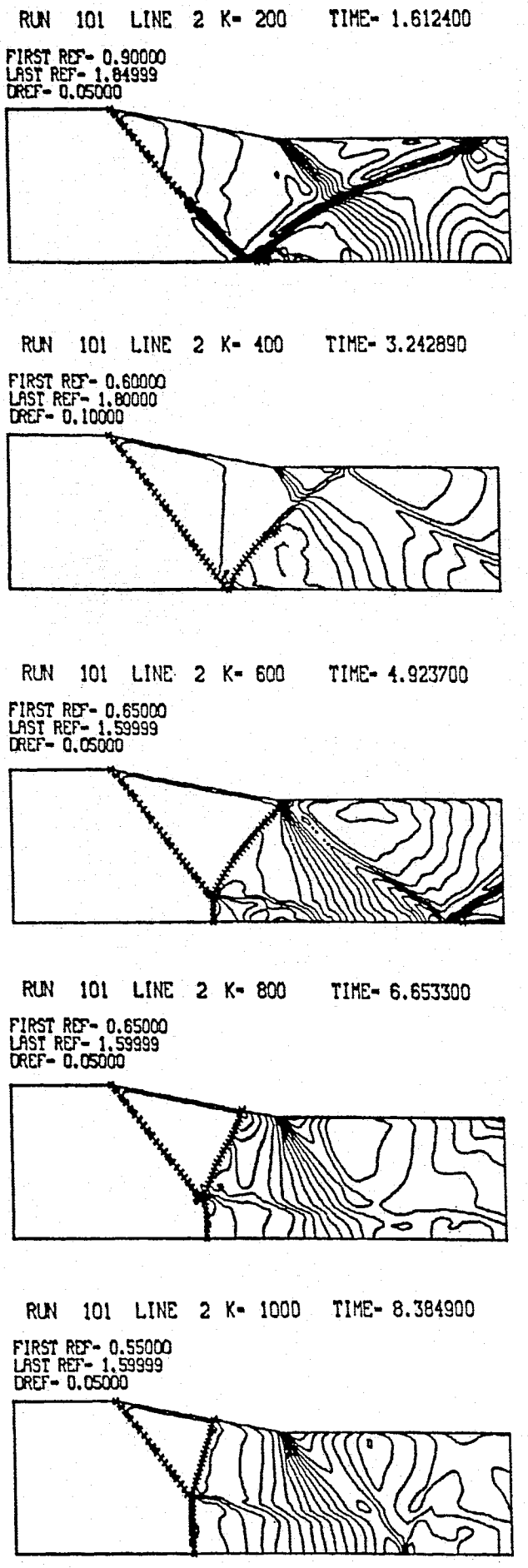

Fig. 6 Unsteady flow in a duct, M = constant lines.

Fig. 7 Intake geometry.

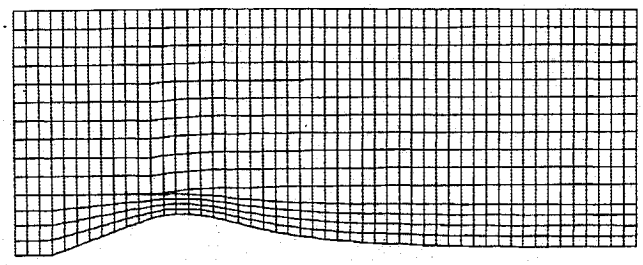

Fig. 8 *H*-grids for intake calculation.

Here, the logic for an easy treatment of multiple shocks finds its first application. The oblique shock (which is all imbedded in a supersonic region) cannot be detected by an analysis of the sign of $(u - a)$, but is detected by an evaluation of the jump in the pertinent Riemann variable. In this respect, we realize that, downstream of the oblique shock, the Riemann variable cannot be evaluated exactly. The derivatives needed to determine it in the λ-scheme calculation are approximated from downstream, in violation of the domain of dependence of $(a/\delta) - u$ in a supersonic flow. It has to be noted, however, that the procedure is justified by the fact that the Riemann variable relative to the shock, $(a/\delta) - (u - W)$, where W is the shock velocity, is carried by a characteristic having a slope equal to $(u - W) - a$, that is, from downstream. No appreciable error seems to affect the computed shock Math number; therefore, the downstream values are correctly reevaluated by the Rankine-Hugoniot conditions.

In Fig. 6 we present a sequence of isobars. The channel cross section is reduced from 1.268 to 1. The slope of the wall is 20%. The Mach number at the entrance is 1.6. At $t = 1.61$, the perturbation produced by the impulsive start has already evolved into an almost steady oblique shock, regularly reflected by the bottom wall, plus a centered expansion wave and its reflection. Note that the shock is picking up strength starting from the wedge corner downward (the bottom part is too weak yet to be fitted; the slope of the captured portion of the shock is evidently smaller than the correct slope of the fitted portion). The reflected shock is also weak and captured isentropically. At $t = 3.24$, the entire main shock is fitted. The strength of the shock is correct, as one can judge by measuring the shock slope (not on Fig. 6, which is affected by some stretching of the hardcopy paper). Note also that the flowfield behind the main shock is now uniform, with the correct Mach number. A region of subsonic flow is growing in size behind the reflected shock, which is obviously picking up strength. A regular reflection can no longer be sustained and it changes into a Mach reflection, as shown by the plot at $t = 4.92$. Here, the subsonic region has grown all the way up to the upper wall and no longer interacts with the centered expansion wave. Remnants of the second and third reflections of the main shock are visible in the lower right corner. The subsonic region pushes the reflected shock and the Mach stem upstream, as we can see at $t = 6.65$ and $t = 8.38$. Further stages of the calculation show the continuing upstream motion of the reflected shock, which eventually swallows the whole main shock, becomes practically a normal shock, and moves out of the computed region.

The results shown in Fig. 6 have been obtained using a 150×30 mesh. The computational time for 1000 steps, on a CRAY-XMP computer is 32 s (not including input/output time).

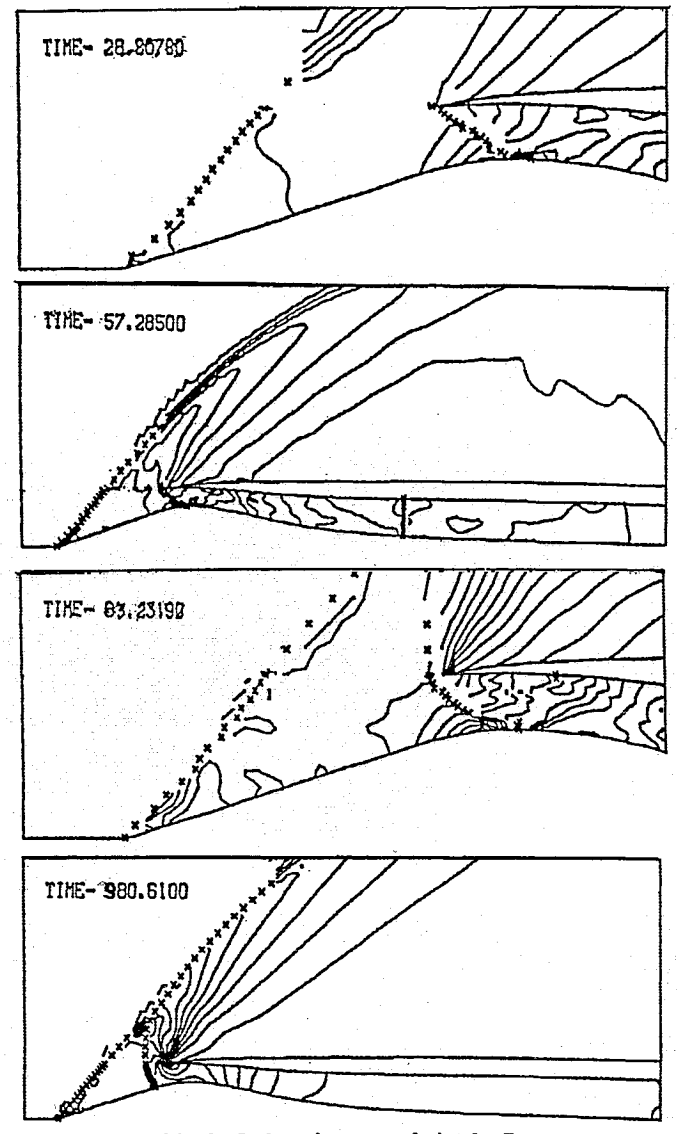

Fig. 9 Isobars for unsteady intake flow.

Intakes

Flows in and about intakes combine all the complications of the flows in ducts with the interaction of an external flowfield. The geometry shown in Fig. 7 has been proposed in a recent GAMM Workshop.[4] For the present calculation, the freestream Mach number is 2 and the Mach number at the end of the duct is 0.3. The calculation starts assuming freestream conditions everywhere, except within the intake, where a linear distribution of M is assumed. Two H grids are used; they match along a horizontal line going through the leading edge of the cowl. The grid entering the duct has 15 intervals vertically; the outer grid has 45 intervals vertically (Fig. 8 does not show the entire computational field). Both grids have 150 intervals horizontally. The vertical spacing of the two grids is not the same. Nevertheless, the computed values match smoothly across the seam; this effect is easily obtained when the λ-scheme is used. In Fig. 9, isobars and fitted shocks are shown. At $t = 28.8$, two supersonic-to-supersonic oblique shocks have formed already, but the one at the left is still too weak for fitting in its upper part. The effect is still evident at $t = 57.3$, but the fitted part has grown bigger. In the same plot, we see a supersonic-to-subsonic shock in the duct, traveling upstream and followed by a region of almost uniform flow. At $t = 85.2$, the oblique shock from the cowl lip has detached. The figure at $t = 180.6$ shows both the first oblique shock fully formed and the shock in the duct very close to the throat.

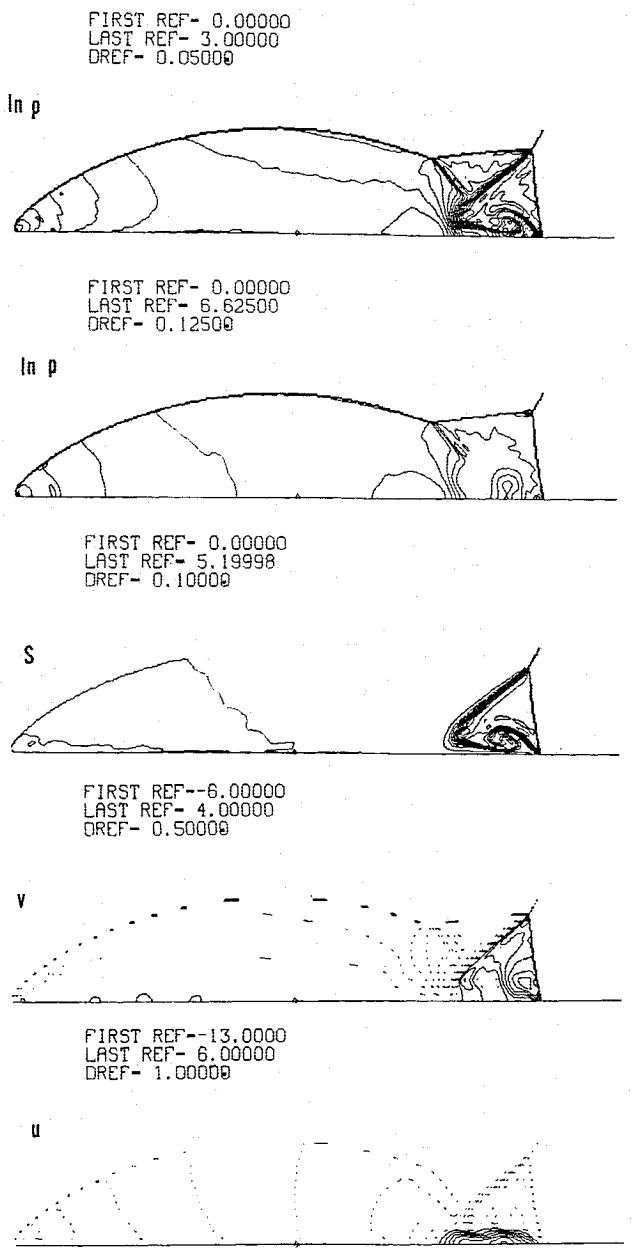

Fig. 10 Complex Mach reflection.

Complex Mach Reflections

To conclude, we consider a problem which has been extensively discussed by Woodward and Colella,[6] the complex Mach reflection of a normal shock on an oblique wall. We use the same Cartesian grid as in Ref. 6, although with fewer points. The impinging shock has a Mach number of 10, so that the Mach number of the flow behind it is 1.83. The impinging shock forms an angle of 60 deg with the wall. The grid is parallel and orthogonal to the wall. With the given data, the reflected shock cannot be attached to the wall, but we force it to be, to match the assumptions of Ref. 6. The flow is self-similar; our calculation, however, begins when the impinging shock has just cleared the leading edge of the wall. We must let the pattern develop and we expect to reach self-similarity only after a reasonable resolution has been obtained.

The main features of this case of complex Mach reflections are: 1) the impinging shock, moving at a high speed into a gas at rest, and leaving behind a region of supersonic flow and high pressure (this is a novel case of subsonic-to-supersonic transition); 2) a Mach stem, moving even faster and growing in height; 3) a curved shock, beginning at the leading edge of the wall and practically normal to it, which turns into a shock parallel to the wall; 4) a kink in the curved shock, followed by a short, weak shock ending in a triple point on the impinging shock; 5) a contact discontinuity stemming from the triple point and reaching the wall; and 6) an extremely weak shock stemming from the kink and pointing toward the contact discontinuity.

In our calculation, no attempts have been made to fit the contact discontinuity; the shocks, however, have all been fitted. This problem requires the complete fitting technique, for shocks oriented in any direction. Typical results are shown in Fig. 10 ($\ln \rho$, $\ln p$, S, u-13.6626, and v). Our results should be compared with Fig. 4 of Ref. 6. Note, in particular, that our plot of u-13.6626 has to be compared with their plot of vx-11.547; the difference in the subtractive constant is due to a different scaling of velocities. Symbols for the fitted shocks are not shown in Fig. 10 (they are too large and the shock points are too close to one another; they would produce a wide black band without allowing points of a different nature to be differentiated). Consequently, the plots look like those obtained by shock-capturing techniques. The comparison with the best result of Ref. 6 is excellent. The entire calculation takes about 69 s for 523 steps on a CRAY-XMP computer. More details and a complete analysis of the problem will be presented elsewhere.

Acknowledgments

The results presented here are largely due to the dedicated, enthusiastic, and intelligent efforts of many of my friends: in particular, A. Dadone and A. Lippolis from the University of Bari, Italy, who did most of the work on airfoils; L. Zannetti from the Politecnico di Torino, Italy, who contributed to the formulation of the intake code; and G. Vradis from the Polytechnic University of Farmingdale, New York, who has extended the shock-fitting logic to a double reflections. Finally, special thanks are owing to C. Fahy, who keeps all our programs running and in good shape, despite our continuous and erratic changes of mind.

References

[1]Moretti, G., "A Technique for Integrating Two-Dimensional Euler Equations," *Computers and Fluids,* Vol. 15, 1987, pp. 59–75.

[2]Moretti, G., "Thoughts and Afterthoughts About Shock Computations," Polytechnic Inst. of Brooklyn, NY, Rept. PIBAL 72-37, 1972.

[3]Moretti, G., "Efficient Calculation of Two-Dimensional, Unsteady, Compressible Flows," in *Advancements in Computational Methods for PDE-VI*, edited by R. Vichnevetsky and R. S. Stepleman, IMACS, 1987, pp. 60–66.

[4]Moretti, G. and Lippolis, A., "Shock-Fitting Calculations of Airfoils and Intakes," GAMM Workshop, Rocquencourt, France, June 1986 (to appear in *Notes on Numerical Fluid Dynamics,* Vieweg Publ., 1987).

[5]Dadone, A. and Moretti, G., "Fast Euler Solver for Transonic Airfoils," *AIAA Journal,* Vol. 26, No. 4, pp. 409–424.

[6]Woodward, P. and Colella, P., "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," *Journal of Computational Physics,* Vol. 54, 1984, pp. 115–173.