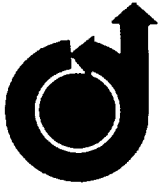


REPRINT
NIS
7N-34-TM
CWAIVED



AIAA 79-1465R

**Two-Dimensional Unsteady Euler-Equation
Solver for Arbitrarily Shaped Flow Regions**

R. G. Hindman, P. Kutler, D. Anderson

Reprinted from

AIAA Journal

Volume 19, Number 4, April 1981, Page 424
Copyright American Institute of Aeronautics and Astronautics, Inc., 1980. All
rights reserved

AIAA 79-1465R

Two-Dimensional Unsteady Euler-Equation Solver for Arbitrarily Shaped Flow Regions

R. G. Hindman*

Iowa State University, Ames, Iowa

Paul Kutler†

NASA Ames Research Center, Moffett Field, Calif.

and

Dale Anderson‡

Iowa State University, Ames, Iowa

A new technique is described for solving supersonic fluid dynamics problems containing multiple regions of continuous flow, each bounded by a permeable or impermeable surface. Region boundaries are, in general, arbitrarily shaped and time dependent. Discretization of such a region for solution by conventional finite difference procedures is accomplished using an elliptic solver which alleviates the dependence on a particular base coordinate system. Multiple regions are coupled together through the boundary conditions. The technique has been applied to a variety of problems including a shock diffraction problem and supersonic flow over a pointed ogive.

Nomenclature

C_1, C_2, \dots, C_8	= coefficients used in matrix S
E	= error norm in integrated Jacobian
e	= total energy per unit volume
\bar{F}	= ξ -flux vector
f	= x -flux vector
G	= grid operator
\bar{G}	= η -flux vector
\bar{g}	= y -flux vector
I	= inverse Jacobian
J	= Jacobian
M_s	= shock Mach number
P	= ξ -forcing function for grid and grid speed equations
p	= static pressure
Q	= η -forcing function for grid and grid speed equations
\bar{Q}	= scaled solution vector, dependent variable vector
\bar{q}	= solution vector, dependent variable vector
r	= right-hand side of linear grid speed equation
S	= coefficient operator of linear grid speed equation
S_j^+	= positive shift operator indicating a forward shift on j -index (i.e., $S_j^+[\phi_{j,k}] = \phi_{j+1,k}$)
S_j^-	= negative shift operator indicating a backward shift on j -index (i.e., $S_j^-[\phi_{j,k}] = \phi_{j-1,k}$)
S_k^+	= positive shift operator indicating a forward shift on k -index (i.e., $S_k^+[\phi_{j,k}] = \phi_{j,k+1}$)
S_k^-	= negative shift operator indicating a backward shift on k -index (i.e., $S_k^-[\phi_{j,k}] = \phi_{j,k-1}$)

t	= time
u, v	= x, y velocity components, respectively
x, y	= Cartesian coordinates
\bar{Z}	= grid speed vector
α, β	= coefficients in grid and grid speed operators
$\Gamma_{\xi}, \Gamma_{\eta}, \Gamma_{\xi\eta}$	= τ, ξ, η direction finite-difference operators, respectively
γ	= coefficient in grid and grid speed operators
$\tilde{\gamma}$	= ratio of specific heats
Δ	= forward difference operator or denotes incremental value
δ	= central difference operator
ϵ	= upper bound for error norm in integrated Jacobian
η, ξ	= general curvilinear coordinates
ρ	= density
τ	= general curvilinear marching coordinate
∇	= backward difference operator
∞	= infinity

Subscripts

g	= indicates association with actual finite-difference grid
j, k	= indices in ξ, η directions, respectively
t, x, y, η, ξ, τ	= partial differentiation with respect to this independent variable
$\eta\eta, \xi\xi$	= second partial derivative with respect to this independent variable
$\xi\eta$	= cross partial derivative

Superscripts

a, b	= used to identify two numerical representations of the same metric quantity
n	= marching direction index

Presented as Paper 79-1465 at the AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., July 23-24, 1979; submitted Nov. 30, 1979; revision received Aug. 14, 1980. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1980. All rights reserved.

*Research Assistant, Iowa State University and NASA Ames Research Center; presently, Assistant Professor, Dept. of Aerospace Engineering. Member AIAA.

†Branch Chief, Applied Computational Aerodynamics Branch. Associate Fellow AIAA.

‡Professor, Dept. of Aerospace Engineering. Member AIAA.

Introduction

THE electronic revolution, which undoubtedly is still in its infancy, has given man an incredibly powerful tool with which to solve many of the problems faced in today's highly technological society. This paper deals with the application of this tool, the high-speed digital computer, to problem solving in fluid dynamics.

Since the advent of the high-speed digital computer, an extensive effort has been made toward obtaining solutions to

fluid dynamics problems for which there exists a concise mathematical treatment resulting in a system of equations and boundary conditions describing an approximation to various physical processes. Many different mathematical descriptions exist depending upon the nature of the assumptions inherent to the derivation. The present work concentrates on fluid dynamics problems in which the flow is assumed to have a compressible, nonviscous nature. Such flow problems are adequately described by the Euler equations. An additional simplification (not restriction) which is made in the present work is that of two-dimensional flow.

There are many computer codes in existence for solving the two-dimensional Euler equations.¹⁻³ Unfortunately, each of them was written with a particular class of problems in mind. This places restrictions on the range of applicability of any computer code. Such restrictions may be categorized as those pertaining to the physics of the flow itself and those pertaining to the problem geometry description. These two categories are not necessarily independent. The former category includes such things as the development of flow singularities, discontinuities, or steep gradients for which there is inadequate numerical treatment. The latter category involves the manner in which the flowfield grid point distribution is defined relative to a base coordinate system. For example, a typical restriction of this type arises when a shock boundary is defined in Cartesian variables as $x_s = x_s(y, t)$ and possibly due to some interaction process, the shock slope, $\partial x_s / \partial y$, becomes unbounded at some point. This difficulty can certainly be remedied in a *given* situation by a coordinate rotation or some other trick, but in the *general* case, any such trick, which may well require extensive code modification, will undoubtedly have similar limitations of its own. It is the purpose of this research effort to remove some of these restrictions by developing a generalized two-dimensional Euler equation solver using a modular approach and a very general treatment of the module geometry, thus providing one computer code capable of solving a wide variety of two-dimensional fluid dynamics problems.

Modular Approach

Although the present approach and the resulting computer code are in no way limited to supersonic flow problems, such problems do provide a more extensive test of the general concepts developed. As a result, the discussion presented herein will tend to emphasize the application of the present approach to supersonic flow.

Many steady and unsteady supersonic flow problems contain multiple regions of continuous flow, each of which is either bounded by a surface across which the flow is discontinuous such as shock waves or slip surfaces or bounded by an impermeable surface such as the body. The present effort is to develop a computational solver which is designed to compute the solution to the Euler equations in an *arbitrary* region or module. The complete flow problem may consist of many of these modules coupled together through the appropriate application of boundary condition procedures. For example, the single Mach reflection planar shock diffraction problem shown in Fig. 1 may be described with two modules. These modules share a slip surface as a common boundary.

Since the present approach requires that the flow module boundaries also be computational boundaries a generalized mapping of the independent variables must be performed (see Fig. 2). It is clear that in the general case these module boundaries are time-varying in nature. Such boundaries must therefore be capable of assuming virtually any shape dictated by the governing equations. Consequently, it is especially important that there is no built-in dependence of the validity of the module geometry description (grid point distribution and movement) on the particular base coordinate system chosen as a reference frame. The unique manner in which the present technique avoids such dependence is described in a later section.

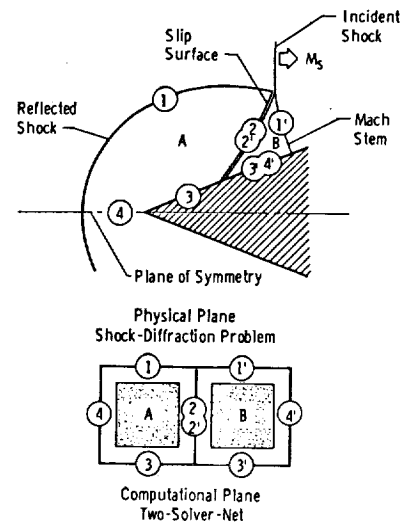


Fig. 1 Conceptual network of solvers.

A literature search reveals that while extensive information exists on the idea of patching together of solutions, for example in boundary layer-inviscid interactions, etc., very little information is available on the present modular type of approach. Ludloff and Friedman⁴ solve the Mach reflection planar shock diffraction problem with what is apparently two modules although they do not specifically indicate such an approach.

In contrast to the lack of information available on the modular type approach, considerable information is available on the subject of generalized geometry. Various types of automatic grid generation procedures have been developed⁵⁻¹⁴ some of which have been applied to domains with moving boundaries.^{1,12,15} The present approach uses the automatic grid generation procedure of Thompson et al.⁵ and extends it in a unique way to allow for domains with time varying as well as arbitrarily shaped boundaries.

Governing Equations

The two-dimensional unsteady Euler equations are written in conservation-law form in Cartesian coordinates as

$$\frac{\partial \bar{q}}{\partial t} + \frac{\partial \bar{f}}{\partial x} + \frac{\partial \bar{g}}{\partial y} = 0 \quad (1)$$

where \bar{q} , \bar{f} , \bar{g} are given by

$$\bar{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \bar{f} = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (p + e)u \end{pmatrix}, \quad \bar{g} = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (p + e)v \end{pmatrix}$$

with (u, v) representing the Cartesian (x, y) velocity components, ρ the density, p the static pressure, and e the total energy per unit volume. e is related to p, ρ, u, v through the equation

$$e = \frac{p}{\gamma - 1} + \frac{\rho}{2} (u^2 + v^2)$$

where γ is the ratio of thermal capacities of the fluid.

Cartesian coordinates are used as the base coordinate system but in order to map bodies and other surfaces onto constant coordinate lines, the following coordinate mapping is introduced (see Fig. 2):

$$\tau = t, \quad \xi = \xi(t, x, y), \quad \eta = \eta(t, x, y) \quad (2)$$

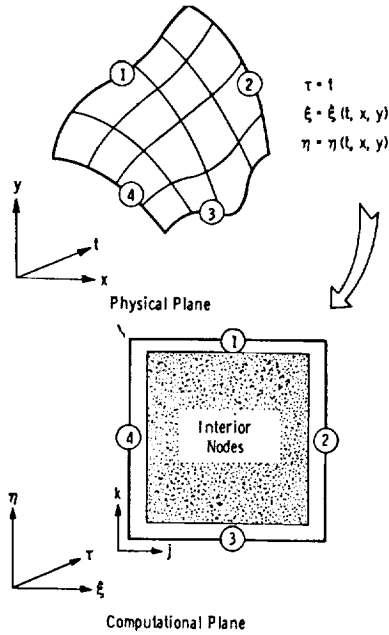


Fig. 2 Transformation of physical plane to computational plane for single solver.

The governing equations, Eq. (1), are transformed by this mapping according to Viviani¹⁶ into a new strong conservation-law form written as

$$\frac{\partial \bar{Q}}{\partial \tau} + \frac{\partial \bar{F}}{\partial \xi} + \frac{\partial \bar{G}}{\partial \eta} = 0 \quad (3)$$

where

$$\begin{aligned} \bar{Q} &= I\bar{q} \\ \bar{F} &= I(\xi_t \bar{q} + \xi_x \bar{f} + \xi_y \bar{g}) \\ \bar{G} &= I(\eta_t \bar{q} + \eta_x \bar{f} + \eta_y \bar{g}) \end{aligned} \quad (4)$$

and

$$\begin{aligned} I &= x_\xi y_\eta - x_\eta y_\xi = \frac{1}{J} \\ J &= \xi_x \eta_y - \xi_y \eta_x \end{aligned}$$

where J is the Jacobian of the mapping and I is the Jacobian of the inverse mapping. The metrics of the mapping are related to the metrics of the inverse mapping by the equations

$$\begin{aligned} I\xi_t &= y_\tau x_\eta - x_\tau y_\eta, & I\xi_x &= y_\eta, & I\xi_y &= -x_\eta \\ I\eta_t &= x_\tau y_\xi - y_\tau x_\xi, & I\eta_x &= -y_\xi, & I\eta_y &= x_\xi \end{aligned} \quad (5)$$

Using Eq. (5), Eq. (4) is rewritten as

$$\begin{aligned} \bar{Q} &= I\bar{q} \\ \bar{F} &= (y_\tau x_\eta - x_\tau y_\eta) \bar{q} + y_\eta \bar{f} - x_\eta \bar{g} \\ \bar{G} &= (x_\tau y_\xi - y_\tau x_\xi) \bar{q} - y_\xi \bar{f} + x_\xi \bar{g} \end{aligned} \quad (6)$$

Conservation-law form of the governing equations is necessary according to Lax¹⁷ to ensure that the jump conditions existing across weak solutions are automatically satisfied. This form of the governing equations thus adds a shock capturing capability to the resulting computer code.

Geometry

There appears to be no universal procedure in the literature for treating the metric and Jacobian terms appearing in Eq. (6). This section, therefore, attempts to identify some logical ground rules which are followed in the treatment of these terms. The concepts involved are more completely discussed by Steger.¹ The rules stem from accuracy considerations and are based on the intuitive suggestion that for a scheme to be considered acceptable, the flowfield code which it supports must be capable of exact reproduction of a uniform flow. That is, with the boundary values held fixed at some uniform flow conditions and the initial flowfield set to these same conditions, the finite difference algorithm should exactly reproduce this same flowfield for all time. This is actually a statement of the independence that should exist between the physical flow and the grid distribution for the case of uniform flow. It provides a simple test for an existing numerical algorithm and a criterion which ties down many of the questions arising in code development work regarding the manner in which the metrics of Eq. (6) should be computed.

Consider the expanded form of Eq. (3);

$$\begin{aligned} \frac{\partial I\bar{q}}{\partial \tau} + \frac{\partial}{\partial \xi} [(y_\tau x_\eta - x_\tau y_\eta) \bar{q} + y_\eta \bar{f} - x_\eta \bar{g}] \\ + \frac{\partial}{\partial \eta} [(x_\tau y_\xi - y_\tau x_\xi) \bar{q} - y_\xi \bar{f} + x_\xi \bar{g}] = 0 \end{aligned} \quad (7)$$

Replace $\partial/\partial\tau$, $\partial/\partial\xi$, $\partial/\partial\eta$ by some finite difference operators, say

$$\frac{\partial}{\partial \tau} \rightarrow \Gamma_\tau, \quad \frac{\partial}{\partial \xi} \rightarrow \Gamma_\xi, \quad \frac{\partial}{\partial \eta} \rightarrow \Gamma_\eta$$

resulting in

$$\begin{aligned} \Gamma_\tau [I\bar{q}] + \Gamma_\xi [(y_\tau x_\eta^a - x_\tau y_\eta^a) \bar{q}] + \Gamma_\eta [(x_\tau y_\xi^a - y_\tau x_\xi^a) \bar{q}] \\ + \Gamma_\xi [y_\eta^b \bar{f}] + \Gamma_\eta [-y_\xi^b \bar{f}] + \Gamma_\xi [-x_\eta^b \bar{g}] + \Gamma_\eta [x_\xi^b \bar{g}] = 0 \end{aligned} \quad (8)$$

where the superscripts a and b appearing on the metric terms are used for later reference to identify two different numerical representations of the same quantity. Now for a uniform flow, since $\bar{f} = f(\bar{q})$, $\bar{g} = g(\bar{q})$, and it is required that $\bar{q} = \text{constant}$ throughout the flowfield for all time, then

$$\Gamma_\tau [I\bar{q}] = \Gamma_\xi [I\bar{q}] = \Gamma_\eta [I\bar{q}] = \Gamma_\xi [I\bar{f}] = \Gamma_\eta [I\bar{f}] = \Gamma_\xi [I\bar{g}] = \Gamma_\eta [I\bar{g}] = 0$$

and the following conditions result:

$$\Gamma_\tau [I] + \Gamma_\xi [y_\tau x_\eta^a - x_\tau y_\eta^a] + \Gamma_\eta [x_\tau y_\xi^a - y_\tau x_\xi^a] = 0 \quad (9)$$

$$\Gamma_\xi [y_\eta^b] - \Gamma_\eta [y_\xi^b] = 0 \quad (10)$$

$$\Gamma_\xi [x_\eta^b] - \Gamma_\eta [x_\xi^b] = 0 \quad (11)$$

Note that the differential analogs of these equations are simply identities for a well behaved mapping. Clearly, Eqs. (10) and (11) are satisfied if the metrics are differenced with the same operators as those used in the finite difference scheme. Equation (9), however, is a numerical representation of an identity coined by Thomas and Lombard¹⁸ as the geometric conservation law (GCL). It says that the GCL equation

$$\frac{\partial I}{\partial \tau} + \frac{\partial}{\partial \xi} [y_\tau x_\eta - x_\tau y_\eta] + \frac{\partial}{\partial \eta} [x_\tau y_\xi - y_\tau x_\xi] = 0 \quad (12)$$

must be differenced in an identical manner to the flow equations, Eq. (7). This result is nontrivial only in the case of

a time varying grid. Thomas and Lombard¹⁸ reached the same conclusion with consistency arguments and analogy with finite volume methods. It must be pointed out, however, that conditions (9-11) insofar as the present effort is concerned are strictly a result of choosing the strong conservation law representation of the governing equations. If the weak conservation-law equations had been chosen, a different set of conditions would result, this time for the differencing of the metric gradients appearing in the source terms. The use of the nonconservative form of the equations results in no special geometry differencing requirements whatsoever.

Thus far two sets of metric values have been identified. They are referred to as *a*-metrics and *b*-metrics based on their superscript. Conditions (10) and (11) dictate the manner in which the *b*-metrics are to be computed. However, the calculation of the *a*-metrics is still a free choice. Although the two sets represent the same physical quantities, they need not be numerically equivalent. In fact, it is shown in a subsequent section that the manner in which the *a*-metrics are computed is dictated by the accuracy with which the integrated Jacobian value, *I*, resulting from Eq. (9), represents the actual Jacobian of the mapping.

The calculation of the metrics requires knowledge of the coordinates (*x,y*) of each grid point. Determination of these coordinates and the speed with which the points move (*x_τ,y_τ*) is the subject of the next section.

Grid and Grid Speed Operators

In order to determine the metric quantities, the coordinates (*x,y*) of each grid point must be known. Also, grid point speeds (*x_τ,y_τ*) are required to advance both the flow solution and the Jacobian [Eqs. (7) and (12)] in time. Due to the time varying nature of the grid boundaries, the location and speed of each interior grid point are necessarily dependent upon the location and speed of the boundary points. Two methods for obtaining such dependence are now described. This elliptic-type dependence may be obtained for the grid by following the approach of Thompson et al.⁵ Given the boundary point coordinates (*x,y*), the interior grid point coordinates are required to satisfy the nonlinear elliptic coupled partial differential equations

$$G[x] = 0, \quad G[y] = 0 \tag{13}$$

with specified boundary values, where

$$G = \alpha \frac{\partial^2}{\partial \xi^2} - 2\beta \frac{\partial^2}{\partial \xi \partial \eta} + \gamma \frac{\partial^2}{\partial \eta^2} + I^2 \left\{ P(\tau, \xi, \eta) \frac{\partial}{\partial \xi} + Q(\tau, \xi, \eta) \frac{\partial}{\partial \eta} \right\} \tag{14}$$

where

$$\alpha = x_\eta^2 + y_\eta^2 \tag{15}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{16}$$

$$\gamma = x_\xi^2 + y_\xi^2 \tag{17}$$

The forcing functions *P*(τ, ξ, η) and *Q*(τ, ξ, η) are used to concentrate the grid lines where they are most needed. For the present study these functions are set to zero.

The requirement that the coordinates (*x,y*) satisfy Eq. (13) plus boundary conditions allows determination of the grid. However, the grid speed values are still unknown on the interior points. The boundary point speeds are assumed known since these values typically represent the speed of shock points, etc., which are determined from the flow solution. Interior values for (*x_τ,y_τ*) could, of course, be obtained with backward finite differences but this requires

extra information at the initial data surface and may be inconsistent with the scheme used to advance the boundary point locations in time. In addition, the solution to Eq. (13) must be iterative due to the nonlinearity of the operator, *G*. Another approach, and the one used in the present study, is to differentiate Eqs. (13) with respect to τ which, for a one-to-one mapping, yields the equation

$$S[\bar{Z}] = \bar{r} \tag{18}$$

where

$$\bar{Z} = (x_\tau, y_\tau)$$

$$\bar{r} = -I^2 (P_\tau x_\xi + Q_\tau x_\eta, P_\tau y_\xi + Q_\tau y_\eta) \tag{19}$$

$$S = \begin{pmatrix} G + C_1(\tau, \xi, \eta) \frac{\partial}{\partial \xi} & C_3(\tau, \xi, \eta) \frac{\partial}{\partial \xi} \\ + C_2(\tau, \xi, \eta) \frac{\partial}{\partial \eta} & + C_4(\tau, \xi, \eta) \frac{\partial}{\partial \eta} \\ \hline C_5(\tau, \xi, \eta) \frac{\partial}{\partial \xi} & G + C_7(\tau, \xi, \eta) \frac{\partial}{\partial \xi} \\ + C_6(\tau, \xi, \eta) \frac{\partial}{\partial \eta} & + C_8(\tau, \xi, \eta) \frac{\partial}{\partial \eta} \end{pmatrix}$$

and

$$\begin{aligned} C_1(\tau, \xi, \eta) &= 2(x_{\eta\eta} x_\xi - x_{\xi\eta} x_\eta + I(Px_\xi + Qx_\eta)y_\eta) \\ C_2(\tau, \xi, \eta) &= 2(x_{\xi\xi} x_\eta - x_{\xi\eta} x_\xi - I(Px_\xi + Qx_\eta)y_\xi) \\ C_3(\tau, \xi, \eta) &= 2(x_{\eta\eta} y_\xi - x_{\xi\eta} y_\eta - I(Px_\xi + Qx_\eta)x_\eta) \\ C_4(\tau, \xi, \eta) &= 2(x_{\xi\xi} y_\eta - x_{\xi\eta} y_\xi + I(Px_\xi + Qx_\eta)x_\xi) \\ C_5(\tau, \xi, \eta) &= 2(y_{\eta\eta} x_\xi - y_{\xi\eta} x_\eta + I(Py_\xi + Qy_\eta)y_\eta) \\ C_6(\tau, \xi, \eta) &= 2(y_{\xi\xi} x_\eta - y_{\xi\eta} x_\xi - I(Py_\xi + Qy_\eta)y_\xi) \\ C_7(\tau, \xi, \eta) &= 2(y_{\eta\eta} y_\xi - y_{\xi\eta} y_\eta - I(Py_\xi + Qy_\eta)x_\eta) \\ C_8(\tau, \xi, \eta) &= 2(y_{\xi\xi} y_\eta - y_{\xi\eta} y_\xi + I(Py_\xi + Qy_\eta)x_\xi) \end{aligned} \tag{20}$$

Once the coordinates (*x,y*) are known at each grid point, the metric quantities and their derivatives may be obtained with finite differences. The result is that the system of partial differential equations represented by Eq. (18) is linear in the dependent variables (*x_τ,y_τ*) with known variable coefficients and a direct method of solution to its finite difference representation may be employed to determine (*x_τ,y_τ*) at all interior points when given the boundary point speeds. In addition, the grid-point locations may be determined from a simple time integration of these computed speeds rather than by solving the nonlinear system, Eq. (13). This point is discussed in the next section.

Coupling of Geometry Treatment and Finite Difference Scheme

Several points are considered in this section involving the accuracy of the procedures developed. This accuracy is intimately connected to the coupling which exists between the finite difference scheme chosen to integrate the governing equations and the treatment of the geometry. MacCormack's standard unsplit predictor-corrector scheme¹⁹ is used to integrate the flow equations, Eq. (7), and the GCL equation, Eq. (12), in time. Application of this scheme yields the

corresponding finite-difference equations:

Predictors

$$(I\bar{q})^{\overline{n+1}} = (I\bar{q})^n - \frac{\Delta\tau}{\Delta\xi} \Delta_j [(y_\tau x_\eta^a - x_\tau y_\eta^a) \bar{q} + y_\eta^b f - x_\eta^b \bar{g}]^n - \frac{\Delta\tau}{\Delta\eta} \Delta_k [(x_\tau y_\xi^a - y_\tau x_\xi^a) \bar{q} - y_\xi^b f + x_\xi^b \bar{g}]^n \quad (21)$$

$$I^{\overline{n+1}} = I^n - \frac{\Delta\tau}{\Delta\xi} \Delta_j [y_\tau x_\eta^a - x_\tau y_\eta^a]^n - \frac{\Delta\tau}{\Delta\eta} \Delta_k [x_\tau y_\xi^a - y_\tau x_\xi^a]^n \quad (22)$$

Correctors

$$(I\bar{q})^{n+1} = \frac{1}{2} \left\{ (I\bar{q})^{\overline{n+1}} + (I\bar{q})^n - \frac{\Delta\tau}{\Delta\xi} \nabla_j [(y_\tau x_\eta^a - x_\tau y_\eta^a) \bar{q} + y_\eta^b f - x_\eta^b \bar{g}]^{\overline{n+1}} - \frac{\Delta\tau}{\Delta\eta} \nabla_k [(x_\tau y_\xi^a - y_\tau x_\xi^a) \bar{q} - y_\xi^b f + x_\xi^b \bar{g}]^{\overline{n+1}} \right\} \quad (23)$$

$$I^{n+1} = \frac{1}{2} \left\{ I^{\overline{n+1}} + I^n - \frac{\Delta\tau}{\Delta\xi} \nabla_j [y_\tau x_\eta^a - x_\tau y_\eta^a]^{\overline{n+1}} - \frac{\Delta\tau}{\Delta\eta} \nabla_k [x_\tau y_\xi^a - y_\tau x_\xi^a]^{\overline{n+1}} \right\} \quad (24)$$

where Δ , ∇ are the usual forward and backward finite-difference operators.

The value of I^{n+1} obtained from Eq. (24) must accurately reflect the true grid structure at $n+1$. The grid must therefore be advanced in time in such a way that the Jacobian, $I^{\overline{n+1}}$, as computed with finite differences from the new grid point locations, $(x, y)^{\overline{n+1}}$, is an accurate representation of $I^{\overline{n+1}}$. That is

$$E = \|I^{\overline{n+1}} - I^{n+1}\| \leq \epsilon$$

where ϵ is a small number. Two methods exist for controlling the order of ϵ . The manner in which the new grid is obtained from a time integration controls the value of $I^{\overline{n+1}}$ and the manner in which the a -metrics in Eqs. (21-24) are treated controls the value of I^{n+1} . It is shown by Hindman²⁰ that if the grid is advanced in time by the Euler predictor-modified Euler corrector scheme,

$$\begin{aligned} x^{\overline{n+1}} &= x^n + \Delta\tau x_\tau^n \\ y^{\overline{n+1}} &= y^n + \Delta\tau y_\tau^n \end{aligned} \quad (25)$$

$$\begin{aligned} x^{n+1} &= \frac{1}{2} (x^n + x^{\overline{n+1}} + \Delta\tau x_\tau^{\overline{n+1}}) \\ y^{n+1} &= \frac{1}{2} (y^n + y^{\overline{n+1}} + \Delta\tau y_\tau^{\overline{n+1}}) \end{aligned} \quad (26)$$

and the a -metrics are differenced as (with $\Delta\xi = \Delta\eta = 1$)

$$\begin{aligned} x_\eta^a &= \delta_k S_j^-[x^n], & y_\eta^a &= \delta_k S_j^-[y^n] \\ x_\xi^a &= \delta_j S_k^-[x^n], & y_\xi^a &= \delta_j S_k^-[y^n] \end{aligned} \quad (27)$$

$$\begin{aligned} x_\eta^{\overline{n+1}} &= \delta_k S_j^+[x^{\overline{n+1}}], & y_\eta^{\overline{n+1}} &= \delta_k S_j^+[y^{\overline{n+1}}] \\ x_\xi^{\overline{n+1}} &= \delta_j S_k^+[x^{\overline{n+1}}], & y_\xi^{\overline{n+1}} &= \delta_j S_k^+[y^{\overline{n+1}}] \end{aligned} \quad (28)$$

where δ denotes the central difference operator and S denotes the shift operator, then

$$\epsilon = O(\Delta\tau^3, \Delta\tau^2\Delta\xi, \Delta\tau^2\Delta\eta)$$

This is a higher order than the accuracy of the numerical determination of $I^{\overline{n+1}}$ from the formula

$$I^{\overline{n+1}} = \delta_j [x^{\overline{n+1}}] \delta_k [y^{\overline{n+1}}] - \delta_j [y^{\overline{n+1}}] \delta_k [x^{\overline{n+1}}] \quad (29)$$

An additional result of obtaining $(x, y)^{\overline{n+1}}$ from Eqs. (25) and (26) is that

$$\|G^{\overline{n+1}}[x^{\overline{n+1}}]\| = \epsilon_1, \quad \|G^{\overline{n+1}}[y^{\overline{n+1}}]\| = \epsilon_2$$

where $\epsilon_{1,2} \sim O(\Delta\tau^3)$. That is (x, y) at the new time are very accurate representations of the values obtained by solving Eq. (13) at $n+1$. This fact provides some assurance that the grid will remain nicely structured as it moves in time.

Algorithm

The algorithm for applying the preceding procedures to an arbitrary module is described in the following 15 steps. A priori knowledge of the initial boundary point locations and their speeds, and the initial flow solution, \bar{q}^n , is assumed.

1) Given (x^n, y^n) at all boundary points, compute (x^n, y^n) at all interior points initially by solving the coupled equations $G^n[x^n] = 0$, $G^n[y^n] = 0$.

2) Compute the a -metrics from Eq. (27).

3) Compute the b -metrics from

$$y_\eta^b = \Delta_k [y^n], \quad x_\eta^b = \Delta_k [x^n]$$

$$y_\xi^b = \Delta_j [y^n], \quad x_\xi^b = \Delta_j [x^n]$$

4) Given (x_τ^n, y_τ^n) at all boundary points, compute (x_τ^n, y_τ^n) at all interior points by solving Eq. (18).

5) Compute the Jacobian I^n from the equation,

$$I^n = I_g^n = \delta_j [x^n] \delta_k [y^n] - \delta_k [x^n] \delta_j [y^n]$$

6) Apply Eqs. (21) and (22) to yield $(I\bar{q})^{\overline{n+1}}$ and $I^{\overline{n+1}}$ at all interior grid points. (Special difference equations are used at boundary points.)

7) Apply Eq. (25) to yield $(x^{\overline{n+1}}, y^{\overline{n+1}})$ at all grid points.

8) Compute $\bar{q}^{\overline{n+1}} = (I\bar{q})^{\overline{n+1}}/I^{\overline{n+1}}$ and apply boundary conditions to obtain the boundary speed $(x_\tau^{\overline{n+1}}, y_\tau^{\overline{n+1}})$ (i.e., shock speed, etc.).

9) Compute a -metrics from Eq. (28).

10) Compute b -metrics from

$$y_\eta^{\overline{n+1}} = \nabla_k [y^{\overline{n+1}}], \quad x_\eta^{\overline{n+1}} = \nabla_k [x^{\overline{n+1}}]$$

$$y_\xi^{\overline{n+1}} = \nabla_j [y^{\overline{n+1}}], \quad x_\xi^{\overline{n+1}} = \nabla_j [x^{\overline{n+1}}]$$

11) Given $(x_\tau^{\overline{n+1}}, y_\tau^{\overline{n+1}})$ at all boundary points, compute $(x_\tau^{\overline{n+1}}, y_\tau^{\overline{n+1}})$ at all interior points by solving Eq. (18).

12) Apply Eqs. (23) and (24) to yield $(I\bar{q})^{n+1}$, I^{n+1} at all interior grid points. (Special difference equations are used at boundary points.)

13) Apply Eq. (26) to yield (x^{n+1}, y^{n+1}) at all grid points.

14) Compute $\bar{q}^{n+1} = (I\bar{q})^{n+1}/I^{n+1}$ and apply boundary conditions to correct \bar{q}^{n+1} and provide $(x_\tau^{n+1}, y_\tau^{n+1})$ on the boundaries.

15) Go to step 2.

Initial and Boundary Conditions

Initial Conditions

The techniques developed in the present effort revolve around the concept of generality. The determination of an

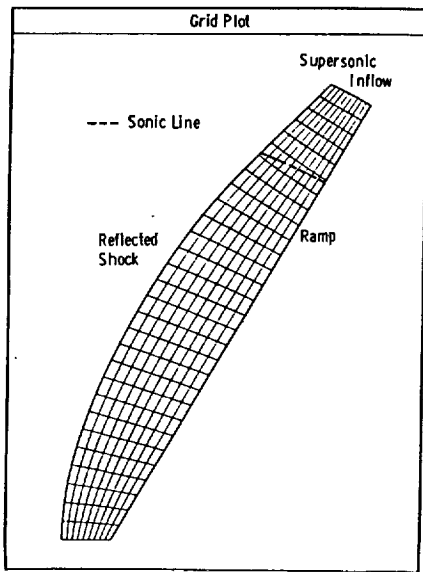


Fig. 3 Grid for regular reflection planar shock diffraction, ramp angle = 60 deg, shock Mach number = 4.71.

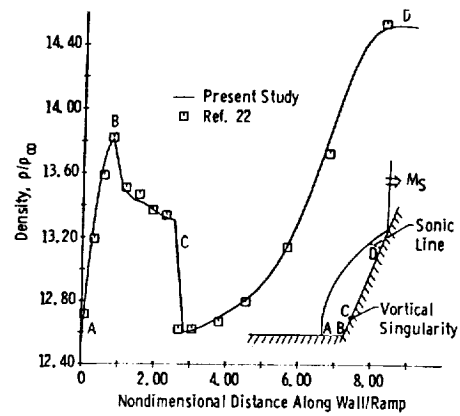


Fig. 5 Wall/ramp density distribution for regular reflection planar shock diffraction, ramp angle = 60 deg, shock Mach number = 4.71.

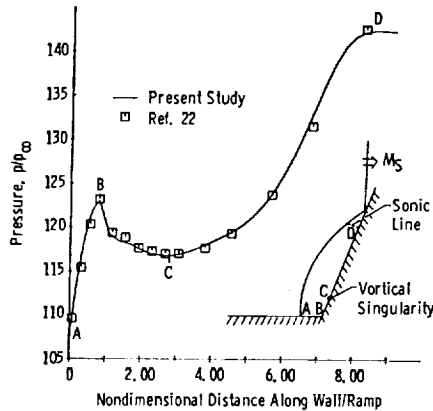


Fig. 4 Wall/ramp pressure distribution for regular reflection planar shock diffraction, ramp angle = 60 deg, shock Mach number = 4.71.

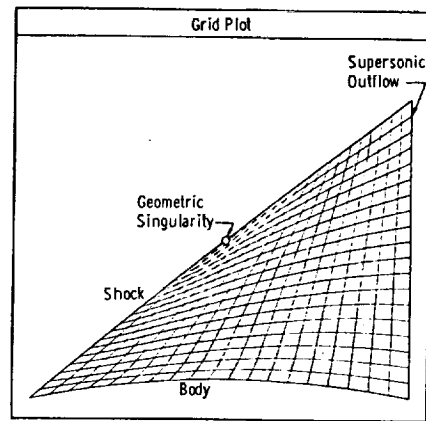


Fig. 6 Grid for single module ogive with geometric singularity on shock, Mach number = 2.0, thickness = 0.1 chord.

initial flowfield for an arbitrary problem does not lend itself to a general treatment. As a result, the initial solution for a given flow configuration is treated as an independent problem and is not part of the existing computer code.

Boundary Conditions

The algorithm described in the previous section applies to an arbitrary module. Steps 8 and 14 require the application of boundary condition procedures. The space limitation prevents a discussion of these procedures here but such a discussion may be found in Ref. 21.

Numerical Results

This section consists of two parts. The first deals with problems which were solved with one module. These problems provide an excellent test of the new geometry procedures developed in this study. The second part deals with a simple preliminary test of the multiple module capability for a problem with two modules and one interface boundary.

Single Module

Numerical results for two problems requiring a single module are presented. The results for a regular reflection planar shock diffraction problem with a shock Mach number of 4.71 and a ramp angle of 60 deg are shown in Figs. 3-5 and compared with the results of Kutler and Shankar²² in Figs. 4

and 5. Both the pressure and density distributions agree well with the Kutler and Shankar solution. It is interesting to note the behavior of the pressure profile for this problem as it approaches the stagnation point. The profile tends toward a zero slope when approaching along the stagnation streamline but tends to spike as the stagnation point is approached along the ramp. This problem is unique in that the grid never reaches a steady-state solution. As $\tau \rightarrow \infty$, $x_r \rightarrow -x/\tau$ and $y_r \rightarrow -y/\tau$ due to the self-similarity that exists with respect to time. The meaning of a converged grid in this case is when $|x_r - x/\tau|$ and $|y_r - y/\tau|$ are sufficiently small. Based on this definition, the converged grid is shown in Fig. 3. Note that a vortical singularity exists for this problem and is clearly shown in Fig. 5. This singularity was captured by the present finite difference algorithm. The computational domain for this problem represents a four-sided physical region. The supersonic inflow side of this region is an artificial boundary and its validity is due to the two-dimensional flow region which exists between the sonic line, shown in Fig. 3, and the shock ramp intersection point. If the numerical solution were required clear up to this intersection point, then the physical domain would have only three sides. The next example illustrates that such three-sided regions pose no serious threat to the present method.

Consider a 10% thick pointed ogive body immersed in a Mach 2 freestream. The converged grid is shown in Fig. 6, thus illustrating a three-sided physical domain. This region is mapped into a computational four-sided region by introducing a "corner" point along the leading edge shock. The Jacobian of the inverse transformation vanishes at this artificial corner thus creating a geometric singularity. This type of singularity is simply an additional corner point option in

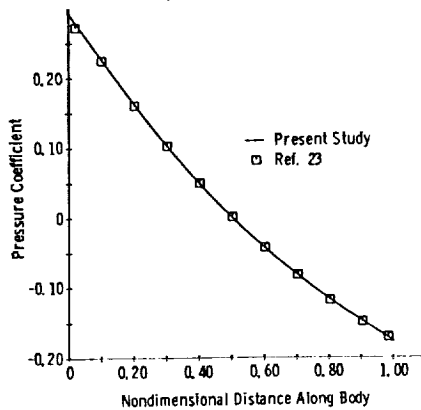


Fig. 7 Surface pressure coefficient for single module ogive with geometric singularity on shock, Mach number = 2.0, thickness = 0.1 chord.

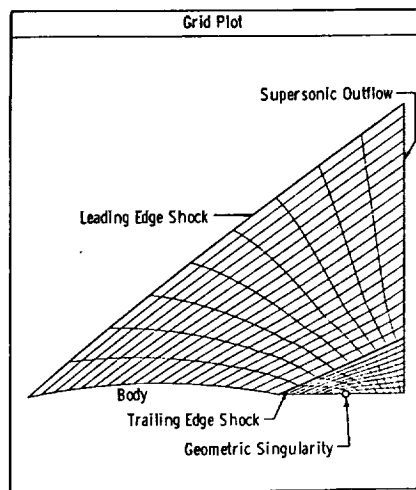


Fig. 8 Grid for double module ogive with aft region geometric singularity, Mach number = 2.0, thickness = 0.1 chord.

the existing computer code. This option is explained by Hindman.²⁰ Solutions were also obtained with this singularity introduced on the ogive surface and on the supersonic outflow boundary instead of on the shock. No difficulties were encountered with any of these cases. Figure 7 illustrates the ogive body surface pressure distribution along with the numerical results of Schiff.²³ The comparison is good. Special attention must now be directed to the grid structure in the vicinity of the geometric singularity. The grid is extremely nonorthogonal in this region. Some researchers^{24,25} have suggested in recent times that such nonorthogonality is undesirable from the standpoint of causing numerical difficulties or inaccuracies in the solution. This is certainly not true for the pointed ogive problem presented here.

Two Modules

The ogive body just discussed also serves as a test case of a double module problem with the trailing edge shock forming the interface boundary between the two modules. Figure 8 illustrates the converged grid and both the leading edge and trailing edge shocks. In this case, the aft region is three-sided and a geometric singularity is introduced along the symmetry boundary. The surface pressure distribution along the ogive and the symmetry boundary is depicted in Fig. 9 and compared to the shock capturing results of Schiff.²³ Generally good agreement is observed where the comparison can be made. The present solution, however, exhibits a slightly higher estimate of the pressure just behind the trailing edge shock. This reflects the fact that the shock slope is slightly

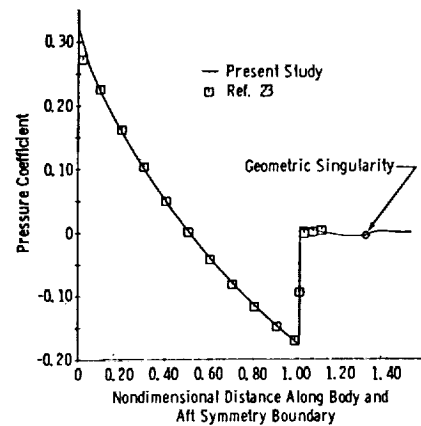


Fig. 9 Surface and aft symmetry boundary pressure coefficient for double module ogive with aft region geometric singularity, Mach number = 2.0, thickness = 0.1 chord.

larger than it should be at this point. This same behavior is observed behind the leading edge shock and is due to some slight inconsistency in the scheme used to treat this type of corner point. The anomaly appearing near the center of the aft region symmetry boundary is due to the existence of the geometric singularity. This is contrary to the findings for the three-sided region in the single module ogive case where no anomaly appeared. A possible explanation of this follows. The grid for the single module case (Fig. 6) appears to be nearly symmetric in some sense with respect to the singular point. Therefore, the truncation error in the solution, which is weakly dependent upon the problem geometry, will be essentially the same on either side of this point. Thus the solution will not be forced by the geometric contribution to the truncation error to exhibit a nonsmooth behavior through the singularity. On the other hand, this symmetry does not exist for the double module case as seen in Fig. 8. This causes the truncation error on one side of the singular point to be different from that on the other side which gives rise to a weak perturbation on the solution at points near the singularity where the geometric variables are rapidly changing.

Concluding Remarks

A general method is presented for solving the unsteady two-dimensional Euler equations on multiple flow regions with arbitrarily-shaped and time-varying boundaries. The method is applicable to problems with moving boundaries provided the velocity of such movement can be determined or specified. This includes problems with moving pistons, structural deformations, accelerating bodies, moving or stationary discontinuity surfaces such as shocks and slip surfaces, etc. In the case of discontinuity surfaces, the scheme has the capability of capturing any discontinuities whose approximate shape and location is not known a priori provided the strength of such discontinuities is not excessive.

The resulting computer code may be used to solve a wide variety of two-dimensional flow problems. The multiple flow region capability may be used to compute one flowfield with multiple regions or it may be used to simultaneously perform a parametric study of the solution for one flow problem.

Acknowledgment

The authors thank the NASA-Ames University Consortium and Engineering Research Institute of Iowa State University for partial support of this work.

References

- 1 Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665, AIAA 10th Fluid and Plasmadynamics Conference, Albuquerque, N. Mex., June 27-29, 1977.

- ²Inouye, M., Rakich, J. V., and Lomax, H., "A Description of Numerical Methods and Computer Programs for Two-Dimensional and Axisymmetric Supersonic Flow Over Blunt-Nosed and Flared Bodies," NASA Tech. Note TN D-2970, 1965.
- ³Fuller, F. B., "Numerical Solutions for Supersonic Flow of an Ideal Gas Around Blunt Two-Dimensional Bodies," NASA Tech. Note TN D-791, 1961.
- ⁴Ludloff, H. F. and Friedman, M. B., "Aerodynamics of Blasts-Diffraction of Blast Around Finite Corners," *Journal of the Aeronautical Sciences*, Vol. 22, No. 1, Jan. 1955, pp. 27-34.
- ⁵Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, 1974, pp. 299-319.
- ⁶Thames, F. C., Thompson, J. F., and Mastin, C. M., "Numerical Solution of the Navier-Stokes Equations for Arbitrary Two-Dimensional Airfoils," NASA SP-347, Part I, March 1975, pp. 469-530.
- ⁷Ghia, U. and Ghia, K. N., "Numerical Generation of a System of Curvilinear Coordinates for Turbine Cascade Flow Analysis," University of Cincinnati Rept. No. AFL 75-4-17, 1975.
- ⁸Amsden, A. A. and Hirt, C. W., "A Simple Scheme for Generating Curvilinear Grids," *Journal of Computational Physics*, Vol. 11, 1973, pp. 348-359.
- ⁹Shapeev, V. P. and Shitov, Yu. A., "On a Method of Constructing Curvilinear Grids," *Chislennyye Metody Mekhaniki Sploshnoi Sredy*, 7, 2, 1976.
- ¹⁰Barfield, W. D., "Numerical Method for Generating Orthogonal Curvilinear Meshes," *Journal of Computational Physics*, Vol. 5, 1975, pp. 23-33.
- ¹¹Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain," *Journal of Computational Physics*, Vol. 8, 1971, pp. 392-408.
- ¹²Godunov, S. K. and Prokopov, G. P., "The Use of Moving Meshes in Gas-Dynamical Computations," *U.S.S.R. Computational Mathematics and Mathematical Physics*, Vol. 12, 1972, pp. 182-195.
- ¹³Danayev, N. T., Liseykin, V. D., and Yanenko, N. N., "A Method of Nonstationary Coordinates in Gas Dynamics," Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, FTD-ID(RS)T-1673-78, Oct. 1978.
- ¹⁴Kovenya, V. M. and Yanenko, N. N., "Implicit Difference Scheme on Mobile Grids for the Numerical Solution to Navier-Stokes Equations of Compressible Gas," Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, FTD-ID(RS)T-1671-78, Oct. 1978.
- ¹⁵Yanenko, N. N., Kroshko, E. Q., Liseikin, V. V., Fomin, V. M., Shapeev, V. P., and Shitov, Yu. A., "Methods for the Construction of Moving Grids for Problems of Fluid Dynamics with Big Deformations," *Proceedings of the 5th International Conference on Numerical Methods in Fluid Dynamics*, Enschede, Netherlands, June 28-July 3, 1976, Springer-Verlag, Berlin, 1976, pp. 454-459.
- ¹⁶Viviand, H., "Conservative Forms of Gas Dynamic Equations," *La Recherche Aerospaciale*, No. 1, Jan.-Feb. 1974, pp. 65-68.
- ¹⁷Lax, P. D., "Weak Solutions for Nonlinear Hyperbolic Equations and Their Numerical Computation," *Communications on Pure and Applied Mathematics*, Vol. 7, Feb. 1954, pp. 159-193.
- ¹⁸Thomas, P. D. and Lombard, C. K., "The Geometric Conservation Law—A Link Between Finite-Difference and Finite-Volume Methods of Flow Computation on Moving Grids," AIAA Paper 78-1208, AIAA 11th Fluid and Plasma Dynamics Conference, Seattle, Wash., July 1978.
- ¹⁹MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, AIAA 7th Aerospace Sciences Meeting, Cincinnati, Ohio, April 1969.
- ²⁰Hindman, R. G., "A Two-Dimensional Unsteady Euler Equation Solver for Flows in Arbitrarily Shaped Regions Using a Modular Concept," Ph.D. Dissertation, Iowa State University, Ames, Iowa, Feb. 1980.
- ²¹Hindman, R. G., Kutler, P., and Anderson, D. A., "A Two-Dimensional Unsteady Euler-Equation Solver for Flow Regions with Arbitrary Boundaries," AIAA Paper 79-1465, AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979.
- ²²Kutler, P. and Shankar, V.S.V., "Diffraction of a Shock Wave by a Compression Corner: Part I—Regular Reflection," AIAA Paper 76-323, July 1976.
- ²³Schiff, L. B., Private communication, NASA Ames Research Center, Moffett Field, Calif., 1978.
- ²⁴Hindman, R. G. and Anderson, D. A., "Numerical Solutions for Inviscid Supersonic Flow About Cones with Large Ellipticities," Iowa State University Engineering Research Institute, Special Rept. 77334, Sept. 1977.
- ²⁵Pandolfi, M., "Supersonic Flow About Elliptic Cones with Large Semi-axis Ratios," Istituto Di Macchine E Motori Per Aeromobili, Italy, Pubblicazione N. PP172, Dec. 1975.