Int. J. Appl. Engng Ed. Vol. 7, No. 3, pp. 242–250, 1991
Printed in Great Britain.

0742–0269/91 $3.00+0.00

An Interactive Method-of-Characteristics Program for Gas-Dynamic Calculations*‡

P. A. JACOBS†
Department of Mechanical Engineering, The University of Queensland, St Lucia, Qld 4067, Australia
C. M. GOURLAY
W.B.M-Stalker Pty Ltd, Aerospace Design Engineers, P.O. Box 203, *Spring Hill, Qld* 4004, *Australia*

An interactive program for computing Method-of-Characteristics solutions to gas-dynamic problems is described. The program is based on isentropic flow theory for axisymmetric or plane two-dimensional geometries. The point-and-do approach taken by the program allows the user to perform geometrically complex calculations without the need to program special routines. When computing a solution the user makes the grid generation decisions interactively while the program handles the detailed computations. Two examples of the program's use are provided.

1. INTRODUCTION

HISTORICALLY, the Method-of-Characteristics (MOC) has been the main approach to solving the Euler equations for supersonic flow in two or three dimensions. The method is well described in texts such as those by Liepmann and Roshko [1] and Ferri [2]. These texts describe procedures for performing the finite-difference calculations by hand and recording the flow solutions in tabular form. This approach is still a fundamental part of the gas dynamics course in many engineering schools. However it is tedious for any but the simplest examples.

Although a number of automatic computer programs have been written to perform the computations of fine grids (see e.g. Richardson [3]) these programs are tuned to compute the solution for a specific flow geometry. This is because the decision making process required to generate a solution grid is difficult to define for a general flow geometry. Recently, characteristic solutions have been embedded in more general finite-difference schemes (see e.g. Moretti [4]).

We have written a program which uses MOC to solve the Euler equations for homentropic two-dimensional compressible flow but we have taken the approach of letting the computer handle the 'book keeping' and detailed computations while letting the user make the logical decisions for grid generation. The calculation of the flow field is then performed by the user specifying what to compute (via a hierarchy of command menus) with the computer performing the detailed calculations and displaying the result graphically on the screen. The calculation process is interactive and the solution develops as the user makes decisions. This approach allows the student user to attempt to solve nontrivial flow situations and gain insight into the method of characteristics and the physics of compressible flows.

A typical flow configuration that may be solved by the program is shown in Fig. 1. Here, the gas flow is from left to right through a diverging duct and it is supersonic everywhere. The program will handle either axisymmetric or plane two-dimensional homentropic flows, and for the axisymmetric case, the axis of symmetry is always the x-axis while y becomes the radial coordinate. The flow conditions at the inflow (or upstream) boundary need to be specified together with the wall profile or a specified Mach number distribution along the x-axis. The downstream boundary should not be specified and is not considered by the program.

In section 2 we will specify the governing equations while in section 3, we give details of the algorithm (unit process) used for the computation of a general point in the free stream. Although the unit process for each type of point is different, there is sufficient similarity between them to be able to describe this case as a 'generic' process. The program arrangement, including data structure and

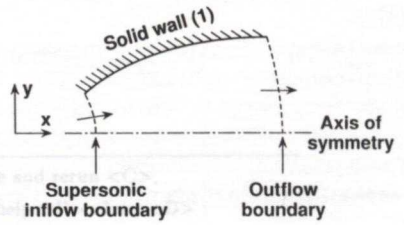


Fig. 1. Typical flow configuration for an axisymmetric problem.

* Paper accepted 14 August 1990.
† Present address: Institute for Computer Applications in Science and Engineering, NASA Langley Research Centre, Hampton, VA 23665, U.S.A.
‡ The program also features in the Software Survey Section of this issue.

command hierarchy, is described in section 4. We then describe two applications. These are, firstly, the design of a Mach 8 nozzle for a reflected-mode shock tunnel (Stalker [5], Jacobs and Stalker [6]), and, secondly, the interaction of a hot jet with an expansion fan (Stalker *et al*. [7]).

2. GOVERNING EQUATIONS

The method of characteristics is implemented as a finite-difference code in which the computational grid is constructed together with the flow solution. There are essentially three parts to the formulation (Anderson [8]).

1. Determine the physical characteristic lines along which the solutions will be propagated. These are the Mach lines for a perfect gas.
2. Determine the compatibility equations which describe the state characteristics. These are ordinary differential equations that govern the dependent variables along the characteristic lines.
3. Solve the compatibility equations along the characteristic lines. The mechanism for doing this is a series of specific computations called 'unit processes'. The particular unit processes used will depend on the situation. For example, computing a new point at the wall is a different process to computing a new point in the free stream.

The equations of motion for axisymmetric flow in streamwise coordinates are (Liepmann and Roshko [1]),

$$\frac{\cot^2\mu}{w}\frac{\partial w}{\partial s} - \frac{\partial \theta}{\partial n} = \frac{\sin\theta}{y}, \tag{1}$$

$$\frac{1}{w}\frac{\partial w}{\partial n} - \frac{\partial \theta}{\partial s} = 0. \tag{2}$$

where n and s are streamline coordinates (see Fig. 2). The velocity is expressed as a magnitude, w, and direction θ, (in a meridian plane) and

$$\mu = \arcsin\left(\frac{1}{M}\right) \tag{3}$$

where μ is the Mach angle. The physical characteristics (Mach lines) in cartesian coordinates are given by

$$\frac{dy}{dx} = \tan(\theta \pm \mu), \tag{4}$$

as shown in Fig. 2. Using the Prandtl-Meyer function

$$\nu = \sqrt{\frac{\gamma+1}{\gamma-1}}\arctan\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)} - \arctan\sqrt{M^2-1}, \tag{5}$$

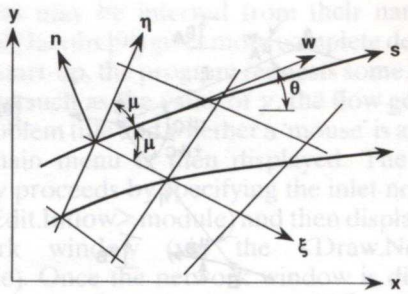


Fig. 2. Coordinates used in the method of characteristics. The radial coordinate in axisymmetric flow and the traverse dimension for 2-D plane flow is represented by 'y'. The families of characteristics are labelled ξ and η while the thick lines represent streamlines.

equations (1–2) may be recast into

$$\frac{\partial \nu}{\partial s} - \tan\mu\frac{\partial \theta}{\partial n} = \tan - \mu\frac{\sin\theta}{y}, \tag{6}$$

$$\tan\mu\frac{\partial \nu}{\partial n} - \frac{\partial \theta}{\partial s} = 0. \tag{7}$$

Transforming to characteristic directions (η, ξ), equations (6–7) become the compatibility relations between the flow quantities ν and θ.

$$\frac{\partial}{\partial \eta}(\nu - \theta) = \sin\mu\frac{\sin\theta}{y}, \tag{8}$$

$$\frac{\partial}{\partial \xi}(\nu + \theta) = \sin\mu\frac{\sin\theta}{y}. \tag{9}$$

The program is essentially an application of the compatibility equations to the computation of the flow conditions for new points along the physical characteristics (η, ξ). We work in flow variables ν and θ and once these are known, we may obtain M and μ from equations (4) and (3) respectively, static pressure from

$$\frac{P}{P_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{-\gamma/\gamma-1}, \tag{10}$$

the static temperature from

$$\frac{T}{T_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{-1} \tag{11}$$

and the velocity from

$$w = M\sqrt{\gamma RT} \tag{12}$$

3. UNIT PROCESS FOR A GENERAL POINT

As an example of a generic unit process the flow solutions at a new free-stream point 'C' can be computed from the known solution at two upstream points 'A', 'B' by taking the following steps. Refer to Fig. 3 for the geometry.

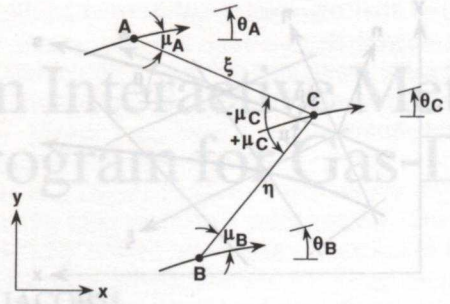


Fig. 3. The geometric layout for computing a new point, C, downstream of two known points, A and B.

Step 1. Guess the flow solution at 'C' by taking the average of the appropriate quantity at A and B. Hence,

$$\theta_C = 1/2(\theta_A + \theta_B), \tag{13a}$$

$$\mu_C = 1/2(\mu_A + \mu_B). \tag{13b}$$

Step 2. Approximate the η and ξ characteristics by straight line segments BC and AC respectively. In general the characteristics will be curved but the straight line mesh should converge to the correct solution as the grid spacing is reduced. Locate point 'C' with coordinates (x_C, y_C)

$$x_C = x_A + \lambda_A(\text{avcos}_A), \tag{14a}$$

$$y_C = y_A + \lambda_A(\text{avsin}_A), \tag{14b}$$

$$\lambda_A = \frac{(x_B - x_A)\text{avsin}_B - (y_B - y_A)\text{avcos}_B}{\text{avcos}_A\text{avsin}_B - \text{avsin}_A\text{avcos}_B}, \tag{15}$$

and

$$\text{avcos}_B = 1/2(\cos(\theta_B + \mu_B) + \cos(\theta_C + \mu_C)), \tag{16a}$$

$$\text{avsin}_B = 1/2(\sin(\theta_B + \mu_B) + \sin(\theta_C + \mu_C)), \tag{16b}$$

$$\text{avcos}_A = 1/2(\cos(\theta_A - \mu_A) + \cos(\theta_C - \mu_C)), \tag{16c}$$

$$\text{avsin}_A = 1/2(\sin(\theta_A - \mu_A) + \sin(\theta_C - \mu_C)), \tag{16d}$$

The variable names such as avsin_B have been used here to be consistent with the naming convention used in the program listed in Jacobs [9]. If the denominator in eqn (15) is zero then the characteristic lines are parallel and we do not have a physically valid case. This may occur during the computation if points A and B are incorrectly specified.

Step 3: Integrate the compatibility relations (8), (9) along straight characteristic segments using the trapezoidal rule (i.e. assume a linear variation of properties along the segment) to get

$$(\nu_C - \theta_C) - (\nu_B - \theta_B) = \left[\sin\mu_B \frac{\sin\theta_B}{y_B} + \sin\mu_C \frac{\sin\theta_C}{y_C}\right]\frac{\Delta\eta_{BC}}{2} = \text{avaxi}_{BC}, \tag{17a}$$

$$(\nu_C + \theta_C) - (\nu_A + \theta_A) = \left[\sin\mu_A \frac{\sin\theta_A}{y_A} + \sin\mu_C \frac{\sin\theta_C}{y_C}\right]\frac{\Delta\xi_{AC}}{2} = \text{avaxi}_{AC}, \tag{17b}$$

where

$$\Delta\eta_{BC} = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}, \tag{18a}$$

$$\Delta\xi_{AC} = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}. \tag{18b}$$

Equations (17a, b) can be combined to give

$$\nu_C = 1/2(\nu_A + \nu_B) + 1/2(\theta_A - \theta_B) + 1/2(\text{avaxi}_{AC} + \text{avaxi}_{BC}), \tag{19a}$$

$$\theta_C = 1/2(\nu_A - \nu_B) + 1/2(\theta_A + \theta_B) + 1/2(\text{avaxi}_{AC} - \text{avaxi}_{BC}), \tag{19b}$$

Step. 4 Since equations (19a, b) are nonlinear and implicit in θ_C, we use the newly computed values as the current guess for the properties at point 'C' and go back to Step 2. Iteration is stopped when the newly computed location of point 'C' is less than a distance ($0.0001\,\Delta\xi_{AC}$) from the previous guess. In general a maximum of 4 iterations is used with 2 or 3 being typical.

For the special case $y_B = 0$ (i.e. 'B' is on the axis of symmetry), we assume that $\partial\theta/\partial y$ = constant as $y \to 0$ and set

$$\text{avaxi}_{BC} = \left[\sin\mu_B \frac{\sin\theta_A}{y_A} + \sin\mu_C \frac{\sin\theta_C}{y_C}\right]\frac{\Delta\eta_{BC}}{2}. \tag{20}$$

Note also that, for plane two-dimensional flow,

$$\text{avaxi}_{BC} = \text{avaxi}_{AC} = 0. \tag{21}$$

4. PROGRAM DESCRIPTION

The information in the characteristic solution is stored in records, one for each node (or point) on the mesh. The data stored in each record includes the (x, y) coordinates, the flow angle, the Prandtl-Meyer function and links to the four nearest neighbour nodes on the mesh as shown in Fig. 4. The program uses this linked list structure to automatically generate new characteristic lines from previously computed lines. Note that, although the data is stored as an array, there is no direct connection between the node number and its position in the characteristic mesh.

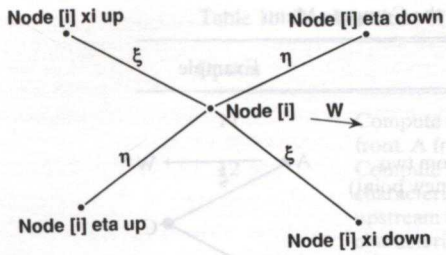


Fig. 4. The arrangement of the linked-list to store the nearest neighbours to node [i]. If a neighbouring node does not exist, zero is stored in the link variable.

The program is comprised of a set of modules (Pascal procedures) that operate on the data structure. Some of these modules perform the unit processes mentioned in section 3 (or combinations of these) while others peform various housekeeping chores such as redrawing the graphics screen and filing the data. Individual modules are activated via a hierarchy of menus as shown in Fig. 5. We identify modules/commands by their menu path. For example, the module to save the current node data is represented by <Edit.Save_Nodes> and the module to compute a new node from two upstream nodes is <Compute.1>. Once a module is activated, the user may be prompted for more information to complete the specific task. For example resizing the displayed window requires the new coordinates of the lower-left point and the upper-right point. Although the action of most modules may be inferred from their names, the manual (Jacobs [9]) gives more complete details.

On start-up, the program requests some initializing data such as the value of γ, the flow geometry, the problem title and whether a 'mouse' is available. The main menu is then displayed. The session usually proceeds by specifying the inlet nodes (via the <Edit.Inflow> module) and then displaying the network window (via the <Draw.Network> module). Once the network window is displayed, the screen is divided into a graphics section that displays a portion of the characteristics mesh and a text section on the lowest two lines that indicates the program status or displays a prompt for user input. Computation of new nodes on the characteristic mesh may then be done using modules from the <Compute> menu. Once computing has started, most of the user's time is spent selecting the nodes on which the <Compute> modules operate. This may be done in one of three ways (as selected at start-up). The user may (1) key-in the node number, (2) point to the node by moving the graphics cursor with the cursor keys or (3) point to the node by moving the graphics cursor with the mouse (if available).

5. AXISYMMETRIC NOZZLE DESIGN

As an example of an axisymmetric flow, we will consider the design of a hypervelocity shock-tunnel nozzle with exit Mach number, $M_e = 8$. To

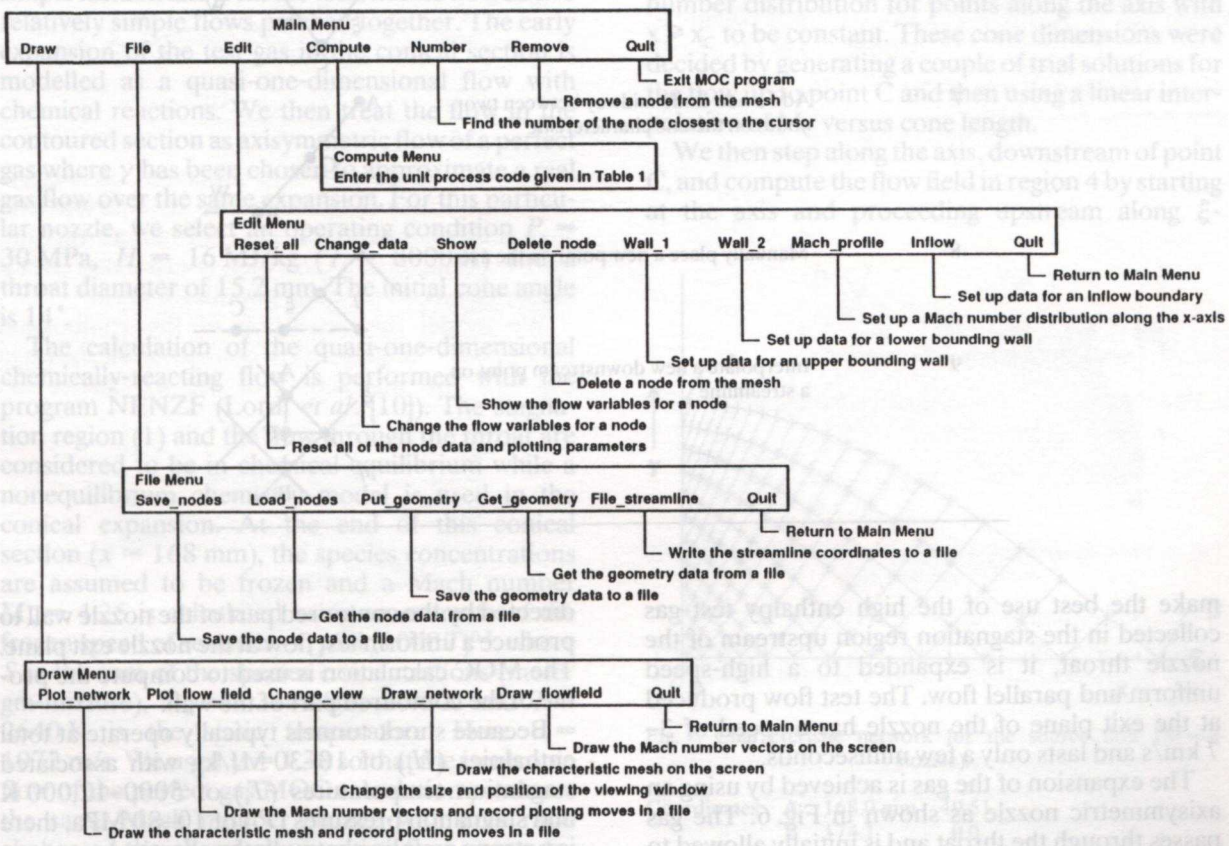
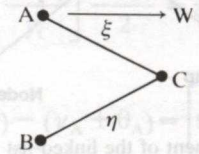
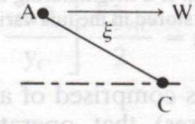
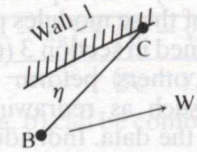
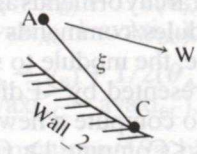
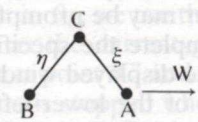
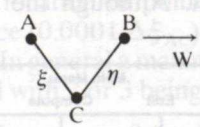
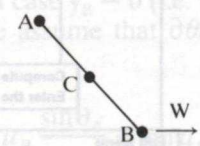
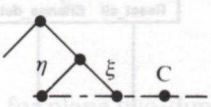
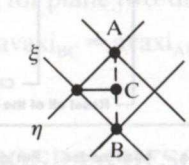


Fig. 5. Hierarchy of command menus.

Table 1a. Unit processes available from the Compute Menu

Option	Action	Example
0	Return to the main menu	
1	Compute a new general point from two upstream points (C denotes the new point)	
2	Compute an axis point from one upstream point	
3	Compute an upper wall point from one upstream point	
4	Compute a lower wall point from one upstream point	
5	Compute an upstream point along an ξ characteristic	
6	Compute an upstream point along an η characteristic	
7	Add a new node midway between two nodes on the one characteristic	
8	Manually place a new point on the axis	
9	Interpolate a new downstream point on a streamline	

make the best use of the high enthalpy test gas collected in the stagnation region upstream of the nozzle throat, it is expanded to a high-speed uniform and parallel flow. The test flow produced at the exit plane of the nozzle has a speed of 3–7 km/s and lasts only a few milliseconds.

The expansion of the gas is achieved by using an axisymmetric nozzle as shown in Fig. 6. The gas passes through the throat and is initially allowed to expand through a conical section. It is then redirected by the contoured part of the nozzle wall to produce a uniform test flow at the nozzle exit plane. The MOC calculation is used to compute the profile of the contoured part of the wall.

Because shock-tunnels typically operate at total enthalpies (H_s) of 10–30 MJ/kg with associated stagnation temperatures (T_s) of 5000–12,000 K and stagnation pressures (P_s) of 10–80 MPa, there is a strong coupling between the chemical reactions of the dissociated air and the axisymmetric gas

Table 1b. Combinations of unit processes available from the Compute Menu

Option	Action
11	Compute a new front of nodes downstream of an existing front. A front extends from the axis to Wall 1.
12	Compute a new line of nodes downstream along an ξ characteristic. The new line is generated starting with an upstream node and following an already existing ξ characteristic.
13	Compute a new line of nodes downstream along an η characteristic.
14	Compute a new line of nodes upstream along an ξ characteristic. Start at a new downstream node, possibly on the axis and work upstream following an already existing ξ characteristic.
15	Perform <Compute.14> several times while placing new points along the axis with a specified spacing.
21	Interpolate several new points along a streamline.

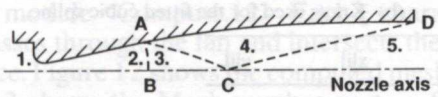


Fig. 6. Conceptual view of a hypersonic nozzle showing the flow regions.
1. Stagnation region.
2. Transition to source flow.
3. Source flow region.
4. Transition to parallel flow.
5. Uniform flow parallel to nozzle axis.

flow. This coupling complicates the flow calculations but we approach the design process in a simple fashion and treat the total nozzle flow as two relatively simple flows patched together. The early expansion of the test gas in the conical section is modelled as a quasi-one-dimensional flow with chemical reactions. We then treat the flow in the contoured section as axisymmetric flow of a perfect gas where γ has been chosen to approximate a real gas flow over the same expansion. For this particular nozzle, we select an operating condition P_s = 30 MPa, H_s = 16 MJ/kg (T_s = 8000 K) and a throat diameter of 15.2 mm. The initial cone angle is 14°.

The calculation of the quasi-one-dimensional chemically-reacting flow is performed with the program NENZF (Lordi *et al.* [10]). The stagnation region (1) and the flow through the throat are considered to be in chemical equilibrium while a nonequilibrium chemistry model is used in the conical expansion. At the end of this conical section (x = 168 mm), the species concentrations are assumed to be frozen and a Mach number M_A = 4.25 is calculated using an estimate of the frozen speed of sound $a = (\gamma S \Re_0 1000 T)^{0.5}$, where S is the sum of the species concentrations (mole/gm-mixture), $\Re_0$ = 8.314 J/gm-mole/K and T = 2640 K is the static temperature. Here a = 1072 m/s. We used γ = 1.38 so that the isentropic flow of the perfect gas MOC calculation produces the same Mach number change with area ratio change as the chemically reacting flow computation.

For the MOC calculation, we assume that the flow across the spherical surface (AB) is uniform source-flow and set up an inflow boundary as a number of nodes positioned in a circular arc between points A and B. Only 12 points are used on this boundary as the program currently retains all of the node data in the limited memory of the microcomputer. Figure 7 shows the mesh that was calculated for the source flow region. Individual ξ-characteristic lines were generated by using module <Compute.12> to start at the inlet boundary and compute a line of nodes down to the axis. At the axial point C, M = 8.02. This is close to our desired exit Mach number so we set the Mach number distribution for points along the axis with $x \geqslant x_C$ to be constant. These cone dimensions were decided by generating a couple of trial solutions for the flow up to point C and then using a linear interpolation of M_C versus cone length.

We then step along the axis, downstream of point C, and compute the flow field in region 4 by starting at the axis and proceeding upstream along ξ-

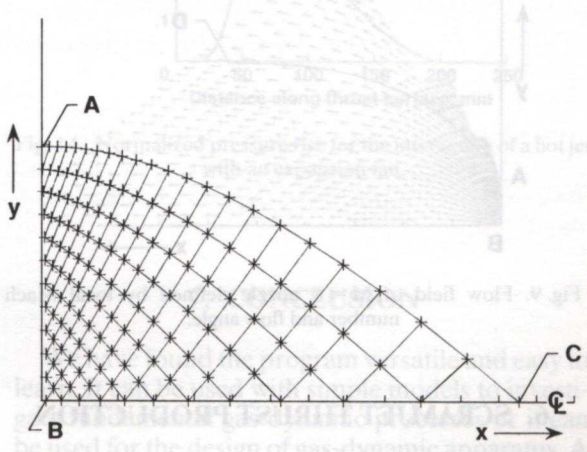


Fig. 7. Characteristic network for the source flow (M = 8 nozzle).

		x	y
Coordinates	A	168.0 mm	49.51
	B	174.1	0.0
	C	800.0	0.0

The x and y scales are unequally stretched.

characteristics (module <Compute.15>). Once the characteristic mesh is generated (see Fig. 8), a streamline can be interpolated through the mesh (module <Compute.21>), starting at point A and finishing when it intersects with the η-characteristic CD. Figure 9 shows the computed Mach number vectors for this flow. The uniform flow region downstream of CD provides a check on the solution consistency.

The data points on the interpolated streamline are then used in a spline fitting routine (Jacobs and Stalker, [6]) to define the nozzle wall as a cubic spline through 8 knots. Table 2 displays the coordinates of the interpolated streamline and the knots for the fitted spline. After the nozzle is fabricated, it is calibrated by measuring Pitot pressure at a number of cross-sections downstream of the exit plane. These measurements (Jacobs [11]) indicate that the nozzle produces reasonably uniform and parallel flow over a range of operating conditions.

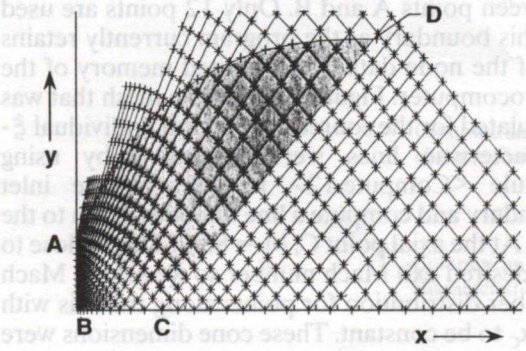


Fig. 8. Characteristic network for the source flow, transition and uniform flow regions (M = 8 nozzle). The interpolated streamline is AD.

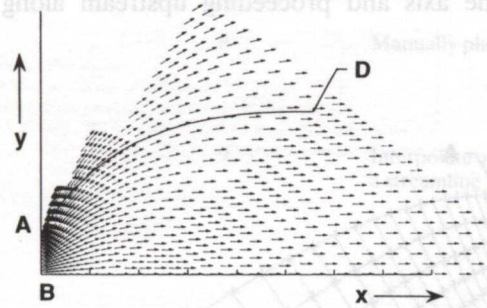


Fig. 9. Flow field of M = 8 nozzle defined by local Mach number and flow angle.

6. SCRAMJET THRUST PRODUCTION

As a second example, we will consider one of the mechanisms of thrust production in a plane two-dimensional supersonic-combustion ramjet (henceforth scramjet). Figure 10 shows a generic scramjet model. The free-stream air enters the inlet and is compressed by a pair of oblique shock

Table 2. (a) Coordinates for the streamline defining contoured section of the M = 8 nozzle wall. (All dimensions in mm) The throat is located at x = 0 with radius y = 7.62 while the conical section has a contour slope dy/dx = 0.2493 finishing at x = 168. The end of the source flow on the axis (point C) occurs at x = 800.

x	y	x	y
168.00	49.512	581.606	123.432
179.953	52.466	667.084	133.856
192.752	55.527	713.254	138.892
220.620	61.917	813.014	148.497
252.618	68.985	931.265	158.278
270.466	72.742	1067.997	167.747
309.804	80.576	1219.466	176.257
354.340	88.934	1386.022	183.529
378.661	93.200	1567.687	189.320
406.530	97.909	1763.840	193.452
437.473	102.933	1972.340	195.889
505.266	113.069	2187.140	196.884

(b) Knots used for the fitted cubic spline.

j	x[j]	y[j]	
1	168.00	49.51	Slope = 0.2493
2	456.48	105.84	
3	744.88	142.08	
4	1033.36	165.52	
5	1321.76	181.04	
6	1610.24	190.40	
7	1898.72	195.20	
8	2187.12	196.88	Second derivative = 0

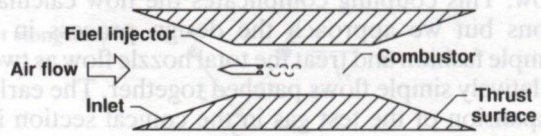


Fig. 10. Generic SCRAMjet model.

waves. The air flow remains supersonic as it enters the combustion chamber where it is mixed with fuel, possibly hydrogen. This hot fuel-air mixture burns spontaneously and then is expanded over a thrust surface to push the aircraft forward.

We model the expansion of the combustion products by the simple flow configuration shown in Fig. 11. The interaction of the expansion fan

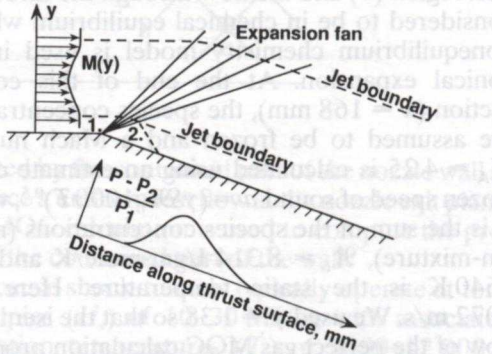


Fig. 11. Interaction of an expansion fan with a hot jet. Region 1:Free stream before expansion. Region 2: Free stream after expansion.

(propagating from the leading edge of the thrust surface) with the hot jet of combustion products has been identified by Stalker *et al.* [7] as one of the mechanisms for producing thrust in the scramjet model.

The flow upstream of the leading edge of the thrust surface ($x = 0$) is assumed to be parallel and the Mach number profile at the intersection of the jet with the leading characteristic of the expansion fan is given in Table 3. The thrust surface has a deflection angle of 15° and the nodes along an ξ characteristic through the expansion fan are given in Table 4. These data, together with the thrust surface position are entered into the program using modules <Edit.Change_Data> and <Edit. Wall_2> respectively. We then compute the jet-fan interaction by starting at nodes in the initial profile and working downstream along ξ characteristics using module <Compute.12>. Each ξ characteristic passes through the fan and intersects the thrust surface. Figure 12 shows the computed mesh while Fig. 13 shows the Mach number vectors for this flow field. Note the convergence of the η characteristics away from the thrust surface. We might expect these to coalesce and form a shock somewhere near the outer edge of the mesh.

Table 3. Conditions along the line x = 6.39 mm through the hot jet. All flow angles are zero. The node #2 here corresponds to node #2 in the expansion fan (Table 4)

Node	y, mm	M
2	2.00	3.35
3	3.37	3.29
4	4.74	3.22
5	6.11	3.15
6	7.47	3.08
7	8.84	3.01
8	10.21	2.95
9	11.58	2.90
10	12.95	2.87
11	14.32	2.85
12	15.68	2.85
13	17.05	2.87
14	18.42	2.90
15	19.79	2.95
16	21.16	3.01
17	22.53	3.08
18	23.89	3.15
19	25.26	3.22
20	26.63	3.29
21	28.00	3.35

Table 4. Nodes on an ξ characteristic through the expansion fan

Node	x, mm	y, mm	M	θ, deg
2	6.39	2.00	3.35	0
22	7.19	1.73	3.51	−3
23	8.09	1.40	3.67	−6
24	9.11	.96	3.85	−9
25	10.27	.42	4.04	−12
26	11.61	−.28	4.24	−15

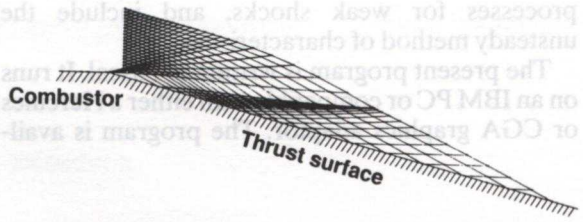


Fig. 12. Characteristic mesh for the interaction of a hot jet with an expansion fan.

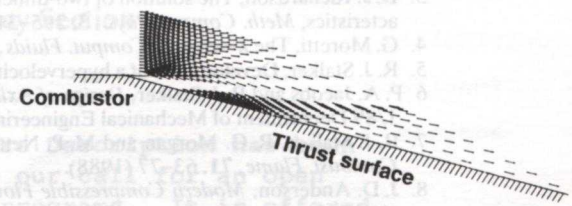


Fig. 13. Mach number vectors for the interaction of a hot jet with an expansion fan.

Given the value of M and the position of nodes along the thrust surface (obtained with module <Edit.Show_Data>). We can use the isentropic relation (10) to compute the static pressure profile that provides the thrust. Figure 14 plots the normalized pressure profile $(P - P_2)/P_1$ where P_1 is the free stream static pressure upstream from the thrust surface and P_2 is the free stream static pressure downstream of the expansion fan. Integrating this profile and taking the component in the streamwise direction gives a value for the thrust produced in our model.

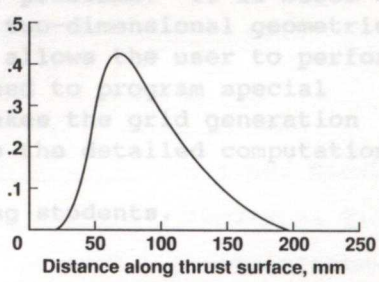


Fig. 14. Normalized pressure rise for the interaction of a hot jet with an expansion fan.

7. CONCLUSION

We have found the program versatile and easy to learn. It can be used with simple models to investigate fundamental gas-dynamic processes or it can be used for the design of gas-dynamic apparatus. A number of graduate students have used the program in the design of their experimental equipment. Although this program is limited to homentropic flows of a perfect gas, a future paper will describe a nonhomentropic program (based on the program in Gourlay, [12]) that will include unit

processes for weak shocks, and include the unsteady method of characteristics.

The present program is written in Pascal. It runs on an IBM PC or compatible with either a Hercules or CGA graphics adaptor. The program is available from the authors along with a user manual (Jacobs [9]). It can be supplied as an executable program or as source code that will compile under the Turbo-Pascal version 3 compiler. It also needs the Turbo-Pascal Graphix Toolbox.

REFERENCES

1. H. W. Liepmann and A. Roshko, *Elements of Gas Dynamics*. John Wiley, New York (1957).
2. A. Ferri, *Elements of Aerodynamics of Supersonic Flows*. Macmillan, New York (1949).
3. D. J. Richardson, The solution of two-dimensional hydrodynamic equations by the method of characteristics, *Meth. Comput. Phys.*, **3**, 295–318 (1964).
4. G. Moretti, The λ-scheme. *Comput. Fluids*, **7**, 191–205 (1979).
5. R. J. Stalker, Development of a hypervelocity wind tunnel. *Aeronaut. J.*, 374–384 (1972).
6. P. A. Jacobs and R. J. Stalker, *Design of Axisymmetric Nozzles for Reflected-Shock Tunnels*. Report 1/89 Department of Mechanical Engineering, The University of Queensland (1989).
7. R. J. Stalker, R. G. Morgan and M. P. Netterfield, Wave processes in Scramjet thrust generation. *Combust. Flame*, **71**, 63–77 (1988).
8. J. D. Anderson, *Modern Compressible Flow, with historical perspective*. McGraw Hill, New York (1982).
9. P. A. Jacobs, *An interactive graphics program for computer-assisted calculation of isentropic supersonic flows*. Report 7/88 Dept. Mechanical Engineering, The University of Queensland (1988).
10. J. A. Lordi, R. E. Mates and J. R. Moselle, Computer program for the numerical solution of non-equilibrium expansions of reacting gas mixtures. NASA-CR-472 (1966).
11. P. A. Jacobs, *An $M = 8$ nozzle for the T4 shock-tunnel*. Report 12/89, Dept. Mechanical Engineering, The University of Queensland (1989).
12. C. M. Gourlay, 'The flow-field generated by inclined ramp tabs in a rocket nozzle exhaust. PhD thesis, The University of Queensland (1988).