NASA Contractor Report 181722

Expansion Tube Test Time Predictions

C.M. Gourlay

UNIVERSITY of QUEENSLAND

St. Lucia, Queensland, 4067
AUSTRALIA

Grant NAGW-674

September 1988

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

ABSTRACT

The interaction of an interface between two gases and a strong expansion is investigated and the effect on flow in an expansion tube is examined. Two mechanisms for the unsteady pitot-pressure fluctuations found in the test section of an expansion tube are proposed. The first mechanism depends on the Rayleigh-Taylor instability of the driver-test gas interface in the presence on a strong expansion. The second mechanism depends on the reflection of the strong expansion from the interface. Predictions compare favourably with experimental results. The theory is expected to be independent of the absolute values of the initial expansion tube filling pressures.

ACKNOWLEDGEMENTS

The author would like to thank Professor R. J. Stalker for the opportunity of working on this project and NASA Langley for funding it.

TABLE OF CONTENTS

APPENDICES

1. INTRODUCTION

An expansion tube is a facility for producing high-enthalpy short-duration hypersonic gas flows. The principle of operation is to use an unsteady expansion for the purpose of expanding the test gas, rather than a nozzle as in a shock tunnel. A facility built at NASA Langley (Moore, 1975) was expected to outperform conventional shock tunnels due to total-enthalpy multiplication (Trimpi, 1962). Experimental experience in the Langley expansion tube (Moore, 1975; Miller, 1977; Miller, 1978; Shinn and Miller, 1978) indicated that the duration of useful test gas flow was much less than expected. Evidence for this was primarily in the form of pitot-pressure time-histories measured at the test section. The pitot pressure time-histories indicated two unexpected phenomena. Firstly, the region of constant pressure test flow was found to be disturbed by large pitot pressure peturbations and, secondly, the magnitude of the pitot pressure was seen to 'dip' under some circumstances (Miller, 1977; Miller, 1978).

This work is aimed at explaining the first mentioned phenomenon, that is the pitot-pressure perturbations. It is expected that explanation of the basic phenomenon, or phenomena, will enable a range of useful test conditions to be established for expansion tubes. The theory formulated here will be applicable to free-piston driven expansion tubes such as at the University of Queensland.

The chapters in this report have been arranged in the following order; firstly, a description of the expansion tube (ideal and real); secondly, a review of the literature relating to the basic mechanisms causing reduction in expansion tube test times; thirdly, the new theory and computer implementation; fourthly, comparison to experiment; and fifthly, the conclusions.

2. THE EXPANSION TUBE

2.1 The Ideal Expansion Tube

The expansion tube in which the experimental data was obtained is the small 'TQ' expansion tube in the Department of Mechanical Engineering at the University of Queensland, Brisbane, Australia. The major difference in operation between this facility and the Langley facility is the free-piston driver (Stalker, 1967). The first advantage of this type of driver is that higher driver temperatures can be achieved than with a conventional driver. Secondly, the temperature and pressure of the driver gas can be varied over a wide range by different choice of diaphragm rupture pressure and filling pressures. Thirdly, the driver is at approximately constant pressure during the shock/expansion tube flow rather than the driver being a constant volume.

Figure 1 shows the wave diagram for a free-piston driven expansion tube. The flow is in three stages. In the first stage the piston is driven down the compression tube by air at high pressure thus compressing the driver gas. The driver gas is chosen to have a high speed of sound. When the piston has imparted most of its energy to the driver gas the pressure of the gas is enough to rupture the primary diaphragm.

In the second stage the hot, high pressure driver gas flows into the shock tube causing a strong shock wave to be propagated down the tube through the test gas. As driver gas flows out of the driver tube the piston velocity is chosen to match this flow-rate and hence to maintain the driver pressure at an approximately constant level. An interface, or contact surface, separates the driver and test gases.

Upon the primary shocks arrival at the secondary diaphragm, which initially separates the test gas from the low pressure acceleration gas, the third stage of flow is initiated. The secondary diaphragm bursts and a strong shock wave propagates through the acceleration gas. An second interface separates the test gas and the acceleration gas. A shock wave may be reflected at the secondary diaphragm. The test gas expands through the strong isentropic centred expansion wave generated by the low gas pressure in the expansion tube, thus acquiring kinetic energy. This expanded test gas arrives at the end of the tube and flows into the test section.

Figure 2 shows the ideal pitot pressure time history at the test section. The acceleration tube flow causes the initial step in pitot pressure and the test gas causes the second much greater step (the magnitude of the step is greater because the temperature of the test gas is significantly less than the acceleration gas). The test period continues until the arrival of the tail of the strong expansion when the pitot pressure begins to ramp up (due to the decrease in Mach number).

2.2 Boundary Layer Entrainment Effect

The effect on shock tube flow of unsteady boundary layers which develop behind the primary shock wave have been studied by Mirels (1963) and (1964) for laminar and turbulent boundary layers. The effect of the boundary layer is to entrain fluid from the region between the primary shock and the interface (see Figure 3). This causes the shock wave and the interface to approach each other, reaching a maximum separation if the tube is long enough. It can be seen that the flow between the shock and the interface is non-uniform in shock-fixed coordinates. When the limiting separation has been reached the free-stream flow has a finite subsonic speed after processing by the (fixed) shock but the contact surface is stationary. Therefore the flow between the shock and the contact surface is non-uniform. As a first approximation the free-stream flow can be assumed to be uniform. This will be true exactly for strong shocks as the shock speed approaches infinity. To find the separation of the shock and the contact surface as a function of distance the approximation of a uniform free-stream can be made and the flow is further assumed to be steady at each instant. The shock is assumed to be strong with constant speed and hence each gas particle undergoes the same increase in entropy as it is processed by the shock. Mirels has derived expressions for the limiting separations and the separation function with distance for both laminar and turbulent boundary layers for a range of real and ideal gases.

This effect has important ramifications on expansion tube flow since it means that the time interval between incident shock and tail of expansion wave arrival at the test section will be decreased (Figure 4).

2.3 Real Gas Effects

Since high enthalpies are expected behind strong shock waves such as those generated in an expansion tube (up to 5 kms^{-1} in TQ acceleration tube section and about 2 kms^{-1} in shock tube section with helium driver - Paull, Stalker and Stringer, 1988) real gas effects such as vibrational excitation, dissociation and relaxation are expected to occur. However, according to Trimpi (1962), less dissociation would be expected to occur than in a reflected shock tunnel. There is the possibility of the flow freezing while being expanded but this should not be significant due to the fact that, except for near the centre of the expansion wave, the expansion is spread over a significant proportion of the acceleration tube length as opposed to the relatively short length of a nozzle in a shock tunnel. Hence it would be expected that there would be time for the gas to relax to equilibrium.

Moore (1975) used two real air model to predict the wall static pressure and pitot-pressure at the test section of the Langley expansion tube as a function of interface velocities. The interface velocity was inferred from measurements of the incident shock wave and by using the theory of Mirels. The two models of air were firstly, thermodynamic equilibrium and secondly, vibrational and chemical freezing. The reflected shock wave from the secondary diaphragm was assumed to lie between the limits of being degenerate or of standing at the secondary diaphragm station. The measured wall static pressures agreed closely with the equilibrium model while the pitot-pressures were between the equilibrium and the frozen predictions. However, Miller (1975) found that predictions assuming equilibrium expansion for air with no reflected shock wave gave the best comparison with experiment.

2.4 Experimental Results from Expansion Tubes

Unsteadiness in Test Section Pitot-Pressure

Results from the Langley and the TQ expansion tubes both reveal unsteady pitot pressure effects showing variation of the acceleration tube pressure (Figure 5). The flow conditions in the University of Queensland facility were chosen to duplicate the Reynolds number, based on shock tube diameter, at the same shock velocities as in the Langley tube (Paull, Stalker and Stringer, 1988). The pitot pressure traces are similar except for the 'dip' phenomenon observed in the Langley tube (Moore, 1975; Miller, 1977 and Miller, 1978). It can be seen from the experimental results that when the acceleration tube pressure is increased, for a constant shock tube pressure, that the frequency of the pressure fluctuations increases. This suggests that there could be more than one mechanism causing fluctuations.

Shock Generated by Secondary Diaphragm Rupture

Ideally the secondary diaphragm which initially separates the test and acceleration tube gases, should be light and rupture instantaneously. However experiments by Shinn and Miller (1978) indicated that these conditions were very often not met in practice. They obtained from tube wall pressure transducers evidence that a shock wave was reflected from the secondary diaphragm and traveled upstream against the oncoming test gas flow. Subsequently the shock wave reflected from the interface between the driver and test gases. In some cases, the shock overtook the acceleration tube incident shock thus increasing wall pressures (see Figure 6). This effect was more pronounced when the secondary diaphragm was of greater thickness and when helium was used as a test gas. In the case of air and carbon dioxide test gases the shock wave was not strong enough to travel upstream and consequently was swept downstream by the oncoming test gas flow (Miller, 1975).

Boundary Layer Transition Effect

It was shown by Shinn and Miller (1978) that the reason for the dip in the pitot pressure of the Langley tube is due to the transition of the boundary layer behind the incident shock wave in the acceleration tube section.

3. LITERATURE REVIEW

3.1 Turbulence at the Interface and Development of Mixing Region

The interface between the driver and test gases in a shock tunnel is expected to be a region of high turbulence (Hooker, 1961) partly explained by non-ideal diaphragm rupture (White, 1958) and Rayleigh-Taylor instability (Taylor, 1950; Lewis, 1950; Lin and Fyfe, 1961). This turbulence leads to mixing of the driver and test gases. Because of mixing, less test gas will be available for expansion through the nozzle into the test section since the interface becomes a mixing region. This phenomenon is also relevant to the driver-test gas interface in an expansion tube since less test gas will be available for processing by the strong expansion and hence the test time will be shortened.

An early analysis to determine the conditions under which a mixing region developed was by White (1958). White considered equal amounts of driver and test gas (volume $V/2$), at different temperatures (T_a and T_b), mixing at the interface at constant pressure. Taking the limit where the temperature ratio across the interface, $N = T_a/T_b$, was large, the change in volume of the interface could be determined. Making the assumption that the driver gas had a smaller molar specific heat, C_{p_a}, (i.e. a monatomic gas) than the test gas, C_{p_b}, an increase in volume was obtained when the driver gas was cooler than the test gas at the interface. The change in volume is given by,

$$1 + \frac{\Delta V}{V} = \frac{1 + N}{2} \left(\frac{1 + C_{p_a}/C_{p_b}}{N + C_{p_a}/C_{p_b}} \right) \qquad (1)$$

and for $N >> 1$,

$$1 + \frac{\Delta V}{V} = \frac{1}{2} \left(1 + C_{p_a}/C_{p_b}\right) \qquad (2)$$

This situation occurs in conventional shock tubes where there is no pre-heating of the driver gas, and in free-piston driven facilities for some conditions. It should be noted that the higher the primary shock Mach number the hotter the test gas in relation to the driver gas and hence the more spread out the mixing region. The flow between the incident shock wave and the interface will be affected by this change in contact region volume, which can be thought of as an increase in effective "piston" velocity. In the other limit where the *expanded* driver gas is much hotter than the test gas a decrease in volume of the mixing region would be expected.

Lin and Fyfe (1961) showed by dimensional arguments that the eddy diffusivity, which controls the spreading rate of the mixing region, was proportional to primary diaphragm diameter.

3.2 Rayleigh-Taylor Instability

Taylor (1950) and Lewis (1950) showed theoretically and experimentally that "...when two superposed fluids of different densities are accelerated in a direction perpendicular to their interface, this surface is stable or unstable according to whether the acceleration is directed from the heavier to the lighter fluid or vice-versa." The amplification and suppression of interface instability is shown in Figure 7. This phenomenon is known as Rayleigh-Taylor instability of accelerated interfaces and is applicable in shock and expansion tube flow to the driver/test gas interface.

3.3 Conditions for Rayleigh-Taylor Instability in Shock Tubes

An analysis was carried out by Levine (1970) who assumed that Rayleigh-Taylor instability of the driver/test gas interface caused a reduction in available test gas in a shock tube. A density gradient was produced by the mixing of cold driver gas with hot test gas at the interface in different proportions assuming constant pressure. A minimum density was found since the driver gas has a smaller average molecular weight than the test gas. This meant that the density of some of the gas in the mixing region was less than the hot gas sample and the driver gas. The acceleration field required to accelerate the less dense gas was provided by relaxation effects in an ionized monatomic test gas behind a strong shock wave. The test gas ionised a certain time after being processed by the primary shock wave, resulting in a reduction in temperature and an increase in density and hence, by continuity, an acceleration (Figure 8).

Levine used a semi-empirical approach to determine the mixing rate at the interface and hence the minimum density and the resulting test gas sample size. He derived an equation of motion for a 'blob' of light gas projected ahead of the contact surface in the presence of a heavier test gas. A simplifying assumption was made that the *ratio* of less to more dense gas remained constant during the period of the shock tube flow. From this he determined whether a test gas sample was likely to accumulate or not for given shock tube conditions. Gas at density ρ_{min} is buoyant in fluid of density ρ_{max} under pseudo-gravitational field g where v_m is the velocity at which fluid is propelled ahead of the contact surface. The equation is,

$$\rho_{min}\frac{dv_m}{dt} = (\rho_{max} - \rho_{min})g \qquad (3)$$

Houwing, Hornung and Sandeman (1981) and Houwing and Sandeman (1983) investigated Rayleigh-Taylor instability of an interface in shock tube flow similar to the case of Levine. They showed that less dense "blobs" can occur under two conditions. Firstly when the driver gas was less dense than the test gas or, as in the case of Levine, when the driver and test gases were mixed. Density profiles as a function of the proportion of driver gas are shown in Figure 9 and are reproduced from Houwing, Hornung and Sandeman (1981). In both cases the test gas temperature was greater than that of the driver gas. Houwing and Sandeman make the statement that if the ratio of the minimum density to the test gas density is calculated using the same method as Levine it is approximately equal to the ratio of average molecular weights across the interface.

Houwing, Hornung and Sandeman considered acceleration fields caused firstly by relaxation effects, due to vibrational non-equilibrium and dissociation behind the primary shock wave, and secondly from boundary layer mass entrainment effects. Only the mass entrainment effect is considered here since real gas effects are not expected to be as significant in expansion tube flow; and will not be taken into account in this analysis.

Houwing et al. (1981) and (1983) derived a more complete equation of motion for the blobs than Levine by including the virtual mass of the buoyant sphere. The equation of motion follows that derived by Batchelor (1967) and is reproduced from Houwing and Sandeman (1983),

$$M\frac{du_b}{dt} = M_0\frac{du_2}{dt} - \frac{1}{2}M_0\frac{d(u_b - u_2)}{dt} \qquad (4)$$

where ρ_m is the density and u_b is the velocity of a non-deforming sphere in a frictionless accelerating fluid of density ρ_2 and velocity u_2. Here M is the mass of the sphere and M_0 is the mass of the fluid displaced. The blobs are assumed to be typical of a large number of particles which comprise the mixing region. When the sphere distorts to conform to the enveloping streamlines, as in the actual flow, the buoyant gas acts like a continuum. The equation of motion is then integrated to obtain the blob velocity as a function of distance downstream of the diaphragm station with the lower limit that the blobs have the same velocity as the contact surface immediately after diaphragm rupture. It is assumed that the flow is steady and that the free-stream velocity decreases monotonically with distance from the shock wave.

Boundary layer entrainment will cause the interface to accelerate due to removal of gas from the region of flow behind the primary shock wave. Hence if blobs which are less dense than the test gas have been generated by interface mixing then a mechanism exists for accelerating some of the interface gas more than the test gas.

4. MECHANISMS CAUSING EARLY PRESSURE FLUCTUATIONS

4.1 Equations of Motion of a Minimum Density Blob

This section discusses pitot-pressure fluctuations caused by blobs of light gas. Due to Rayleigh-Taylor instability of accelerated interfaces blobs of gas, of a lower density than the test gas, can be generated by mixing at the interface. These blobs tend to accelerate more rapidly than the surrounding test gas, in the direction of the acceleration. Hence in the acceleration field of the strong expansion they overtake the test gas and have the potential to arrive at the test section during the period of useful test flow causing pressure fluctuations (see Figure 10). As mentioned above there are two ways of generating lower density blobs. Firstly if the driver gas is less dense than the test gas blobs of driver gas will be buoyant in the test gas; and secondly by mixing in different proportions a cold monatomic driver gas with a hot diatomic test gas, where the driver gas has a smaller average molecular weight than the test gas, a blob with a density less than that of both gases can be produced.

The mechanism is implemented in three stages. Firstly the driver and test gases mix generating less dense blobs of gas. Secondly the blobs separate from the contact surface in the shock tube flow region, due to Rayleigh-Taylor instablity, and are propelled forward of the test gas by the boundary layer entrainment effect in the shock tube region. Thirdly the blobs are propelled forward by the strong expansion in the acceleration tube region. The mixing model of Levine was used for the generation of the blobs at the interface. In the shock tube the equations used were similar to those of Houwing and Sandeman. New equations are developed for flow in the strong expansion region.

Generation of Density Minimum

The minimum density due to mixing at the interface is derived below.

-Conservation of Energy

$$m_d h_d + m_t h_t = mh \qquad (5)$$

$$\alpha \frac{5}{2} R_d T_d + (1 - \alpha) \frac{9}{2} R_t T_t = h \qquad (6)$$

$$\alpha = \frac{m_d}{m_d + m_t} \qquad (7)$$

$$R_i = \frac{\bar{R}}{W_i} \qquad (8)$$

where	d	=	denotes driver gas
	t	=	denotes test gas
	m	=	mass
	h	=	static enthalpy
	R	=	engineering gas constant
	T	=	static temperature
	σ	=	driver mass fraction
	$\bar{R}$	=	universal gas constant
	W_i	=	molecular weight

-Enthalpy of Mixture at Interface (average translational and rotational kinetic energy)

$$\frac{h}{m} = \frac{n_t\left(\frac{3}{2}\bar{R}\bar{T} + 3\bar{R}\bar{T}\right) + n_d\left(\frac{3}{2}\bar{R}\bar{T}\right)}{m_t + m_d}$$

$$= \alpha\frac{5}{2}\,R_dT + (1-\alpha)\,\frac{7}{2}\,R_tT \qquad (9)$$

$$T = \frac{5\alpha R_dT_d + 7(1-\alpha)R_tT_t}{5\alpha R_d + 7(1-\alpha)R_t} \qquad (10)$$

-Equation of State

$$\rho = \frac{m_d + m_t}{V} = \frac{p}{(\alpha R_d + (1-\alpha)R_t)\ T}$$

$$\rho = \frac{p\left((5R_d - 7R_t)\alpha + 9R_t\right)}{\left((R_d - R_t)\alpha + R_t\right)\left((5R_dT_d - 7R_tT_t)\alpha + 7R_tT_t\right)} \qquad (11)$$

where	p	=	static pressure
	ρ	=	static density

-Density Ratio

$$\frac{\rho}{\rho_t} = \frac{\left(5\frac{W_t}{W_d} - 7\right)\alpha + 7}{\left(\left(\frac{W_t}{W_d} - 1\right)\alpha + 1\right)\left(\left(5\,\frac{T_d}{T_t}\,\frac{W_t}{W_d} - 7\right)\alpha + 7\right)} \qquad (12)$$

when $\alpha \rightarrow 0$ then $\frac{\rho}{\rho_t} \rightarrow 1$

when $\alpha \rightarrow 1$ then $\frac{\rho}{\rho_t} \rightarrow \frac{T_t}{T_d}\,\frac{W_d}{W_t}$

Hence for an ideal gas the density minimum depends on the ratio of molecular weights and the temperature ratio across the interface, assuming monatomic driver gas and diatomic test gas.

-Minimum Density Ratio, obtained by differentiating (12),

$$\alpha = -\frac{b}{a} \pm \sqrt{\left(\frac{b}{a}\right)^2 - \frac{bd}{ac} - \frac{bf}{ae} + \frac{df}{ce}} \qquad (13)$$

where

$$a = 5\,\frac{W_t}{W_d} - 7$$

$$b = 7$$

$$c = \frac{W_t}{W_d} - 1$$

$$d = 1$$

$$e = 5\,\frac{T_d}{T_t}\,\frac{W_t}{W_d} - 7$$

$$f = 7 \qquad (14)$$

In a real gas an increase in C_p due to vibration and dissociation will produce a lower minimum density that for an calorically ideal gas.

Acceleration of Blobs and Interface during Expansion Tube Flow

It is assumed that no heat is transferred to the blobs from the test gas during their flight. An analysis was performed to determine the maximum blob size which could be heated significantly during the period of the shock tube flow. The heated blobs were found to be too small to be important with a diameter less than one thirtieth of the expansion tube diameter.

The blobs are assumed to be in mechanical equilibrium with the test gas during the period of their flight through the test gas, i.e. at the same pressure. Thus when the test gas pressure changes due to the expansion wave the blob properties change accordingly, assuming no heat transfer, to keep them at the same pressure as the surrounding test gas.

Following Batchelor,

$$M\,\underline{U} = -\frac{M_0}{2}\left(\underline{U} - \underline{V}\right) + M_0\,\underline{V} \qquad (15)$$

where M is the mass of the sphere, M_0 is the mass of the fluid displaced by the sphere, $\underline{U}$ is the velocity of the sphere, and $\underline{V}$ is the velocity the

ORIGINAL PAGE IS
OF POOR QUALITY

surrounding fluid would have had if the sphere was not present. The first term represents the acceleration of the sphere, the second represents the acceleration reaction of the displaced fluid on the sphere and the third represents the buoyancy force. Rearranging and taking differentials one obtains,

$$d\underset{\sim}{U} = \frac{\frac{3}{2} M_0}{M + \frac{1}{2} M_0} \, d\underset{\sim}{V} \qquad (16)$$

for a small change in the velocity of the sphere as a function of a small change in velocity of the surrounding fluid. It can be seen that when $M/M_0 < 1$ that $d\underset{\sim}{U} > d\underset{\sim}{V}$ and hence if this model was applied to blobs of less dense gas generated at the interface then they would accelerated more quickly than the surrounding test gas. The equation of motion can be integrated one mesh step at a time taking local values of $\underset{\sim}{V}$ and M/M_0.

Effect on Pitot Pressure

The effect of blobs on the test section pitot-pressure, is expected to be fluctuations due to the difference in temperature and density of the blobs compared to the test gas. The frequency of the fluctuations is expected to relate to the most probable blob size. Thus the presence of blobs means the pitot-pressure trace will display a contact region spread over a considerable time period instead of a sharply defined interface.

4.2 Reflection of Waves from the Contact Surface

Another possible mechanism for producing pitot-pressure perturbations is now discussed. Under some circumstances the strong expansion through which the test gas expands, after reflecting from the driver-test gas interface, can arrive at the test section during the test period. Since the interface is expected to be a region of high turbulence due to non-ideal diaphragm rupture there is the potential for unsteady pressure perturbations to be propagated along the characteristics of the reflected expansion and hence to disrupt conditions at the test section during the test period. The effect of the reflected expansion on the pitot pressure trace is shown in Figure 11. It can be seen that the pitot pressure falls, until the arrival of the contact surface, rather than rises as in the case where the reflected expansion does not arrive at the test section. This is due to the reversal of the velocity gradient. Unsteady effects which exist at the interface can then be propagated along the characteristics of the reflected expansion. It should be noted that the trajectory of the reflection of the head of the strong expansion can be determined analytically.

If small perturbations of the flow properties, generated at the contact surface, are assumed this is equivalent to having another two families of physical characteristics and another two families of state characteristics corresponding to the perturbations of the gas properties. Mirels and Braun (1962) solved the problem of the propagation of small perturbations in uniform and self-similar flows. In their cases the physical characteristics were coincident for both the perturbed and unperturbed components of the state properties. Hence the magnitude of the perturbations of the state variables could be integrated along characteristics in the expansion wave, since it was self-similar, and the pitot pressure fluctuations calculated. The magnitudes of the fluctuations depended on the turbulence at the interface. However in this analysis only the time of arrival of pressure perturbations is sought so the magnitude of the perturbations is not required.

As found from the Langley experiments an upstream propagating shock wave can be generated by the rupture of the secondary diaphragm. An estimate of the effect of this shock wave on the test section flow can be obtained by noting that the trajectory of a very weak shock wave is the same as that of the reflected head of the strong expansion (Figure 12). Thus an approximation to the time of arrival of such a shock wave can be gained by finding the time at which the reflected head of the strong expansion arrives at the test section.

Another possible effect of the reflected shock wave is that after it has been transmitted *through* the driver-test gas interface bifurcation may occur. Bifurcation occurs when the tube wall boundary layer stagnation pressure is not great enough to allow it to be decelerated by a normal shock and hence oblique shocks form and gas collects at the foot causing it to grow with time (Figure 13). This means that a jet of gas can be generated on the walls of the tube, formed by the oblique shock waves, which has a *greater* velocity towards the test section end of the tube than does the gas processed by the normal shock wave. Thus driver gas can arrive at the test section earlier than expected. This mechanism has been examined by Davies and Wilson (1969) and others. It will not be pursued here.

It should be noted that no pitot-pressure perturbations occurred in the Langley tube without the presence of a secondary diaphragm (Shinn and

Miller, 1978). Hence the secondary diaphragm must be important in the generation of pitot-pressure fluctuations.

5. IMPLEMENTATION OF SOLUTION

The method of characteristics for unsteady flow in one dimension has been used to predict the flow in the expansion tube assuming perfect gases. The effect of boundary layer entrainment has been included approximately by calculating new trajectories for the driver-test and test-acceleration gas interfaces. The effect of the entrainment on the free-stream flow has not been considered; this is known as the uniform free-stream approximation. The pitot pressure has been predicted as a function of time at the test section by the Rayleigh pitot pressure formula with an empirical correction being employed to account for the higher predicted shock speeds than those measured in experiment.

5.1 Basic Assumptions

The gases are all assumed to be thermally and calorifically perfect and in thermodynamic equilibrium. In the expansion tube flow ideal diaphragm rupture has been assumed. The free-piston driver is treated as a constant pressure reservoir with the conditions calculated using isentropic compression of the driver gas. The Mirels boundary layer entrainment effect has been included assuming the uniform free-stream approximation for the contact surface trajectories. Primary shock waves have been assumed to have constant velocity and hence no entropy variation exists for different particles of gas. The latter two assumptions are both applicable for strong shock waves. At the interface mixing occurs adiabatically and isobarically in an initial thin contact surface. The blobs of low density gas generated are small, non-deforming spheres in mechanical equilibrium with the surrounding gas flow and are typical of a large number of such which make up the mixing front. The test section flow is assumed to be quasi-steady for the pitot pressure determination.

5.2 Computer Program

The finite difference equations for the method of characteristics for one-dimensional unsteady flow are given in Appendix A. The method was implemented on a Apple Macintosh Plus Personal Computer in compiled BASIC. The method uses a combined graphical-numerical approach. The computer implementation is interactive and the procedure is similar to that required if the wave diagram were to be constructed on graph paper, except the machine does all the calculations and the 'house-keeping'. A flow chart of the program logic is shown as Figure 14. The program waits for the user to select from the menu the next type of point he wishes to calculate, for example; 'Interior', 'Contact', or 'Expansion'. Once the user has defined this he then selects, using the mouse, the existing points from which he wants the new point to be calculated. The computer then calculates the new point and displays its location on the screen. The properties at a point can be perused at any time by the user. A database is generated on disc as calculation proceeds so that the solution can be regenerated or added to at a later date. The program listing can be found in Appendix B.

When calculating the wave diagram it becomes necessary to refine the mesh if flow properties are changing rapidly. In this case the program has a facility for 'splitting' the mesh by linear interpolation of properties between known points. This raises the problem of how to save the data for each point in the database such that it can be retrieved and the flowfield reconstructed correctly. The storage of data adopts a method of inter-relating records known as linked records. Stored with the values of the properties at each point are two numbers. These numbers give the numbers of the records where the properties of the two upwind points on which the point depends are stored. It is easy therefore to split the mesh and to change the way the records are linked when a new intermediate point is created.

5.3 Verification of Computer Code and Truncation Error

The computer code was checked by calculating the trajectory of the contact surface through the expansion fan when the same gas at the same conditions is on either side. This is the same as calculating a particle path. The three families of characteristics give,

$$\frac{dt}{dx} = \frac{1}{u - a} \tag{17}$$

$$\frac{u}{2} + \frac{a}{\gamma - 1} = \frac{u_1}{2} + \frac{a_1}{\gamma - 1} = \frac{u_2}{2} + \frac{a_2}{\gamma - 1} \tag{18}$$

$$\frac{dt}{dx} = \frac{1}{u} \tag{19}$$

$$x = \frac{t}{1 - \gamma}\left[a_1(1 + \gamma)\left(\frac{t}{t_1}\right)^{\frac{1-\gamma}{1+\gamma}} - 2\left(a_1 + \frac{\gamma - 1}{2}u_1\right)\right] \tag{20}$$

The numerical solution to the wave diagram is given as Figure 15. The analytical solution for the path line is exactly coincident to the numerical solution to the resolution of the diagram.

The compatibility relations of the method of characteristics depend on the mesh and so approximations must be made in computing flow properties. Prior to use of this procedure, the point properties are assumed to vary in a polynomial fashion along characteristics between the known and unknown points. The order of the polynomial variation can be selected according to the desired accuracy required of the solution. A method of improving these inherent approximations is to use a mesh size which is appropriate for the level of accuracy required. The average value of the properties was used for calculation of the physical characteristics hence the accuracy of the mesh is of the order of $(\Delta x)^3$ and $(\Delta t)^3$. For calculations of flow properties on the contact surface average values were also used but iteration was required hence the maximum accuracy expected, after convergence, is of the order of $(\Delta u)^3$ and $(\Delta p)^3$. The calculation of flow properties at other points is exact.

ORIGINAL PAGE IS
OF POOR QUALITY

6. COMPARISON OF COMPUTATIONS WITH EXPERIMENT

6.1 Shock Speed

The predicted shock speeds are up to thirty percent higher than the measured ones. (All the following experimental results are taken from Paull, Stalker and Stringer, 1988.) This was accounted for in the pitot-pressure prediction by the use of an empirical correction factor.

6.2 Langley Results

As the acceleration tube pressure is increased the model predicts that unsteady effects, due to the reflected expansion, should arrive earlier. Blobs are predicted but they arrive very much later than in the useful test time and so are not relevant. There is evidence of another unsteady effect at the lower acceleration tube pressures possibly due to waves being reflected from the walls of the tube. The dip noted in the case with the highest shock tube pressure is due to boundary-layer transition in the acceleration tube.

The reflected expansion trends compare favourably to reflected shock trends as determined by wall pressures measurements (Shinn and Miller (1978). Hence the reflected head of the expansion predicts the reflected shock behaviour at least qualitatively.

6.3 U.Q. Argon Driver Results

As the acceleration tube pressure is increased the model predicts that unsteady effects, due to the reflected expansion, should arrive earlier. Blobs are *not* predicted. There is evidence of another unsteady effect at the lower acceleration tube pressures possibly due to waves being reflected from the walls of the tube.

No blobs are predicted for any case with an argon driver. (For an ideal gas the density minimum depends on the ratio of molecular weights and the temperature ratio across the interface, assuming monatomic driver and diatomic test gas).

The absence of the dip phenomenon can be explained by the fact that boundary layer transition would not be expected from Reynolds number calculations based on the acceleration tube length of TQ.

6.4 U.Q. Helium Driver Results

Taking the column of results for which the acceleration tube pressure is approximately 120 mm it can be seen for lower shock tube pressures the reflected expansion arrives before the blobs while for the higher shock tube pressures the blobs arrive before the reflected expansion. The blobs arrive *latest* for the central case (p_1 = 13.8 kPa), while the reflected expansion arrives *latest* for the p_1 = 101 kPa case. It can also be seen that the blobs tend to produce large scale pitot pressure fluctuations while the reflected expansion causes fluctuations on a smaller scale.

Considering holding shock tube pressure constant while varying the acceleration tube pressure; an increase in acceleration tube pressure causes both the blobs *and* the reflected expansion to arrive earlier. This agrees with the Langley and argon driver predictions (for the reflected expansion). These effects can be seen by considering either the top row or the bottom row of the array.

(It should be noted that for the case in the extreme upper right corner of the array that the expansion reflected from the driver-test gas interface is predicted to further interact with the test-acceleration gas interface. This effect was not included in the model and hence this prediction is less certain. What *is* certain is that the reflected expansion arrives very early.)

6.5 U.Q. Air Driver Result

There were no blobs predicted for the case with an air driver and although the reflected expansion is predicted to arrive reasonably early the fluctuations are not sufficient to degrade to a serious extent the relatively long period of test flow found in this case.

7. CONCLUSIONS

The model developed here explains some of the previously unexplained features of expansion tube flow tolerably well. It also indicates that the two mechanisms considered are pressure independent, except for a small pressure dependence of the boundary layer entrainment effect. Therefore either scaling the initial pressure filling *ratios* either up or down should produce flow with the same characteristics. Hence the initial filling pressure ratios that produce the longest period of test flow can be obtained. Therefore no additional work is required to determine the best pressure ratios for higher absolute pressure conditions.

8. REFERENCES

Batchelor, G. K. (1967) *An Introduction to Fluid Dynamics*. Cambridge, University Press.

Davies, L. and Wilson, J.L. (1969) "Influence of Reflected Shock and Boundary-Layer Interaction on Shock-Tube Flows." *Phys Fluids Suppl I*. 37-43.

Ferri, A. ed. (1961) *Fundamental Data Obtained from Shock-Tube Experiments*. AGARD, Pergamon.

Hooker, W. J. (1961) "Testing Time and Contact-Zone Penomena in Shock Tube-Flows." *Phys Fluids* 4. 1451-1463.

Houwing, A. F. P. and Sandeman, R. J. (1983) "Contact Zone Instability due to Real Gas Effects in Shock Tube Flows." *Proc 14th Int Sym Shock Tubes Waves*, Sydney. 285-292.

Houwing, A. F. P., Hornung, H. G. and Sandeman, R. J. (1981) "Investigation of the Distortion of Shock-Fronts in Real Gases." *Proc 13th Int Sym Shock Tubes Waves*, Niagara Falls. 176-184.

Levine, M. A. (1970) "Turbulent Mixing at the Contact Surface in a Driven Shock Wave." *Phy Fluids* 13. 1166-1171.

Lewis, D. J. (1950) "The Instability of Liquid Surfaces when Accelerated in a Direction Perpendicular to their Planes. II" *Proc Roy Soc* A202, London. 81-96.

Liepmann, H. W. and Roshko, A. (1957) *Elements of Gas Dynamics*. New York, John Wiley.

Lin, S-C and Fyfe, W. I. (1961) "Low-Density Shock Tube for Chemical Kinetics Studies." *Phys Fluids* 4. 238-249.

Miller, C. G. (1975) "Shock Shapes on Blunt Bodies in Hypersonic-Hypervelocity Helium, Air, and CO_2 Flows, and Calibration Results in Langley 6-Inch Expansion Tube." *NASA Technical Note* D-7800.

Miller, C. G. (1977) "Operational Experience in the Langley Expansion Tube with Various Test Gases." *NASA Technical Memorandum* 78637.

Miller, C. G. (1978) "Operational Experience in the Langley Expansion Tube with Various Test Gases." *AIAA Journal* 16. 195-196.

Mirels, H. (1963) *"Test Time in Low-Pressure Shock Tubes."* Phys Fluids 6. 1201-1214.

Mirels, H. (1964) "Shock Tube Test Time Limitation Due to Turbulent-Wall Boundary Layer." *AIAA Journal* 2. 84-93.

Mirels, H. and Braun, W. H. (1962) "Perturbed One-Dimensional Unsteady Flows Including Transverse Magnetic-Field Effects." *Phys Fluids* 5. 259-265.

Mirels, H. and Mullen, J. F. (1964) "Small Perturbation Theory for Shock-Tube Attenuation and Nonuniformity." *Phys Fluids* 7. 1208-1218.

Moore, J. A. (1975) "Description and Initial Operating Performance of the Langley 6-Inch Expansion Tube using Heated Helium Driver Gas." *NASA Technical Memorandum* X-3240.

Rudinger, G. (1955) *Wave Diagrams for Nonsteady Flow in Ducts*. New York, Van Nostrand.

Shinn, J. L. and Miller, C. G. (1978) "Experimental Perfect-Gas Study of Expansion-Tube Flow Characteristics." *NASA Technical Paper* 1317.

Stalker, R. J. (1964) "Area Change with a Free-Piston Shock Tube." *AIAA Journal* 2. 396-397.

Stalker, R. J. (1967) "A Study of the Free-Piston Shock Tunnel." *AIAA Journal* 5. 2160-2165.

Paull, A., Stalker, R. J. and Stringer, I. (1988) "Experiments on an Expansion Tube with a Free-Piston Driver." Submitted to *AIAA Journal*.

Taylor, Sir G. I. (1950) "The Instability of Liquid Surfaces when Accelerated in a Direction Perpendicular to their Planes. I" *Proc Roy Soc* A201, London. 192-196.

Trimpi, R. L. (1962) "A Preliminary Theoretical Study of the Expansion Tube, a New Device for Producing High-Enthalpy Short-Duration Hypersonic Gas Flows." *NASA Technical Report* R-133.

White, D. R. (1958) "Influence of Diaphragm Opening Time on Shock-Tube Flows." *J Fluid Mech* 4. 585-599.

9. FIGURES

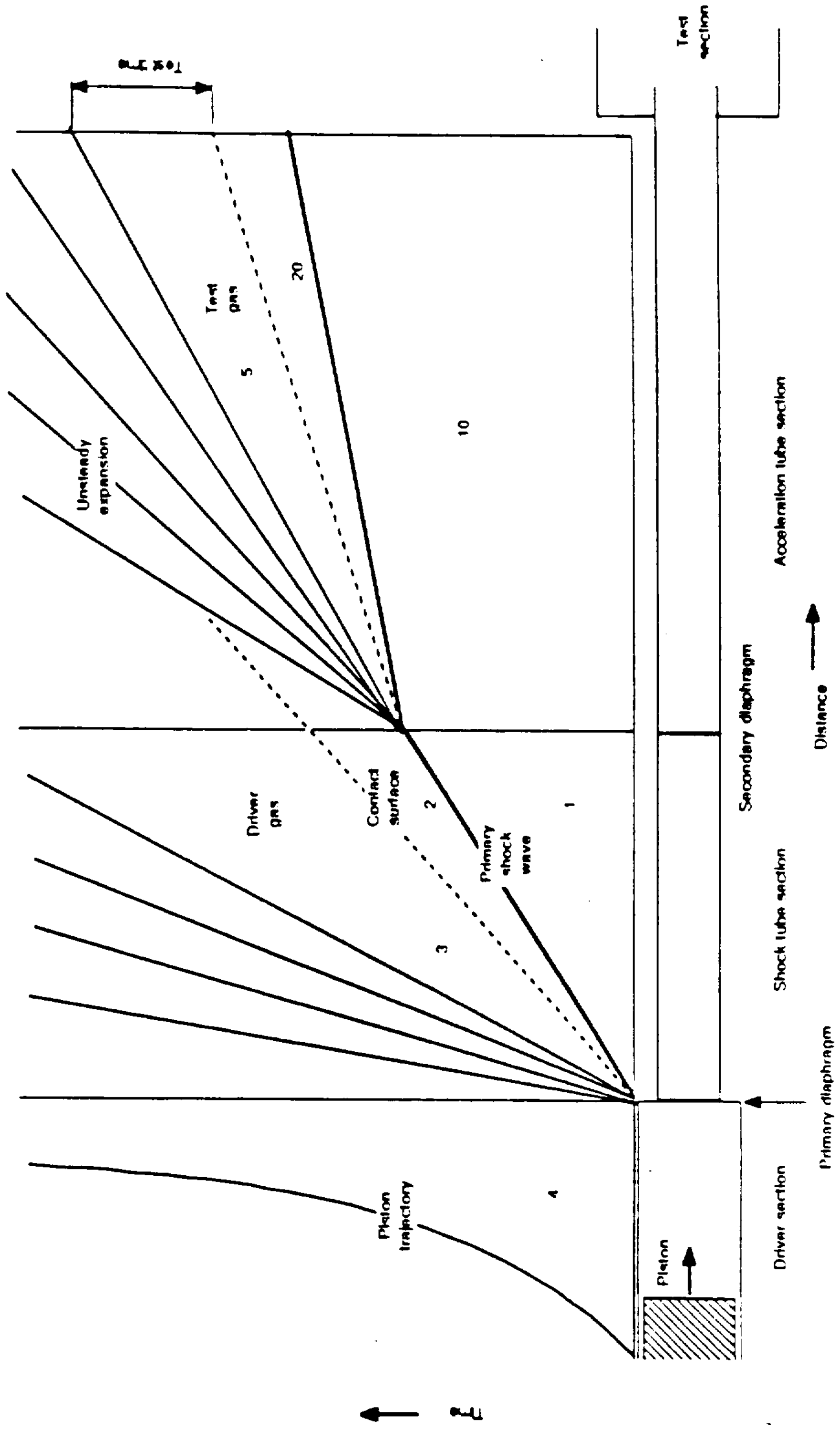
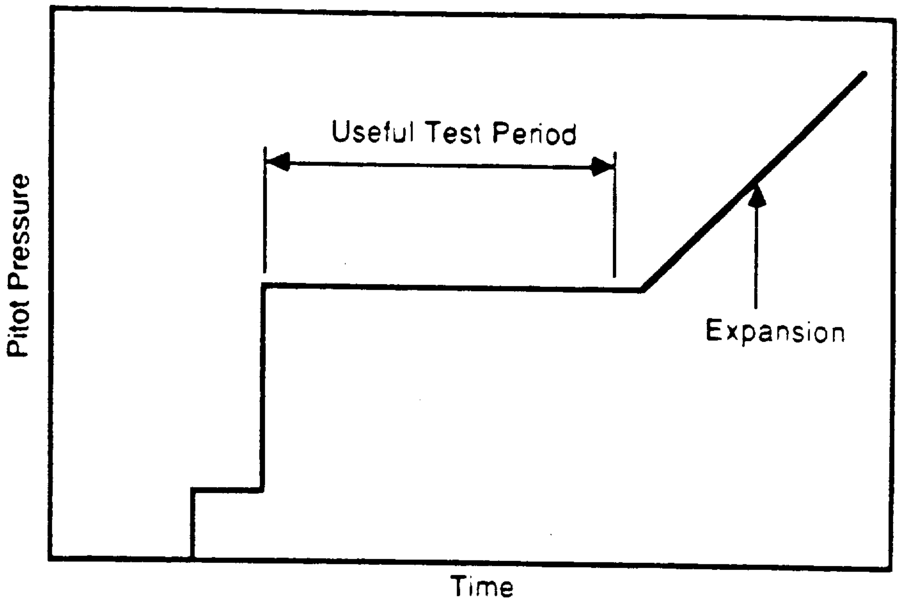


Figure 1: Wave diagram of ideal expansion tube flow.



No Reflected Expansion

Figure 2: Ideal pitot-pressure time-history at the test section.

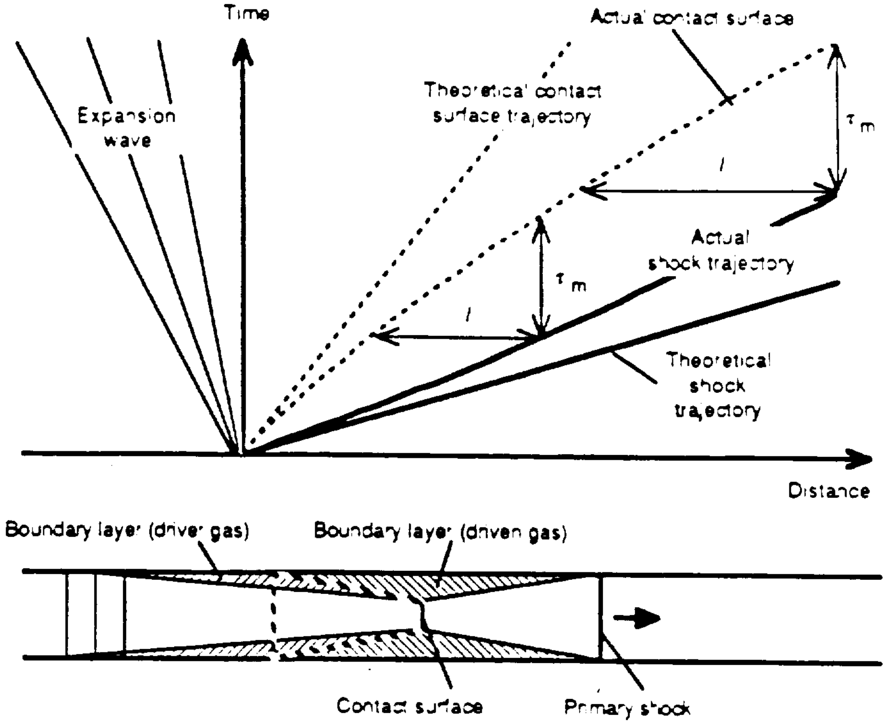


Figure 3: Mirel's boundary layer entrainment effect.

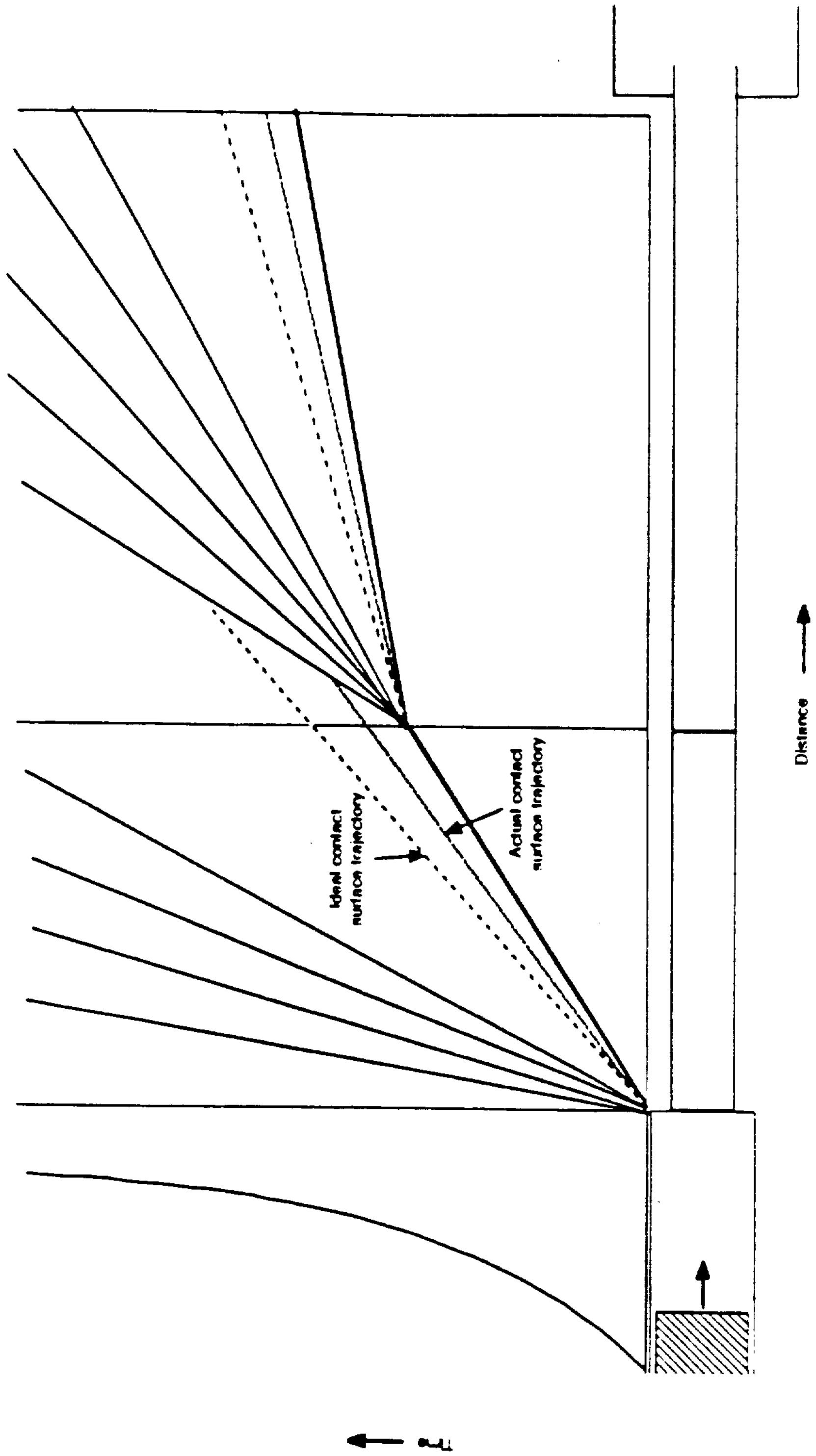

Ideal contact surface trajectory
Actual contact surface trajectory
Distance
t

Figure 4: Entrainment effect on expansion tube flow.

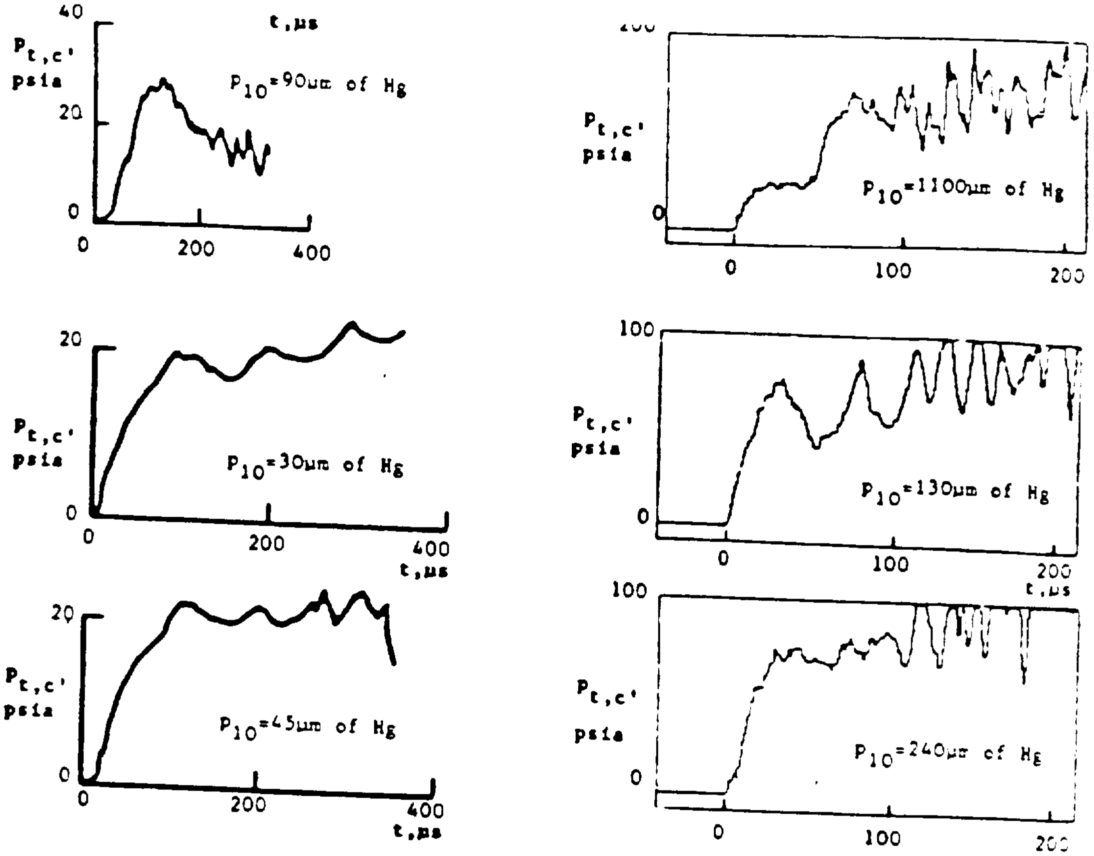


Figure 5: Typical measured pitot-pressure time-histories.

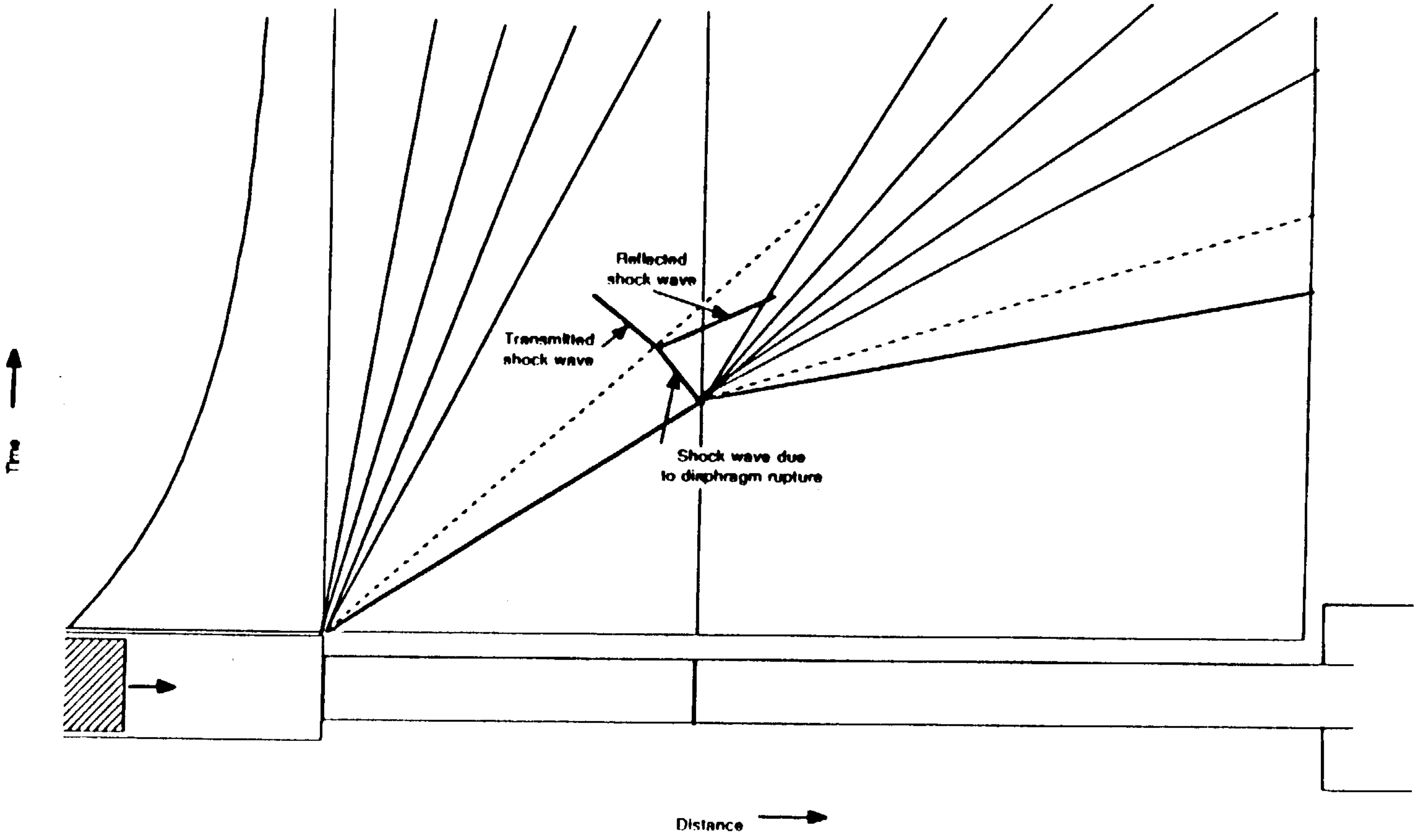


Figure 6: Shock wave generated by the secondary diaphragm

ORIGINAL PAGE IS
OF POOR QUALITY

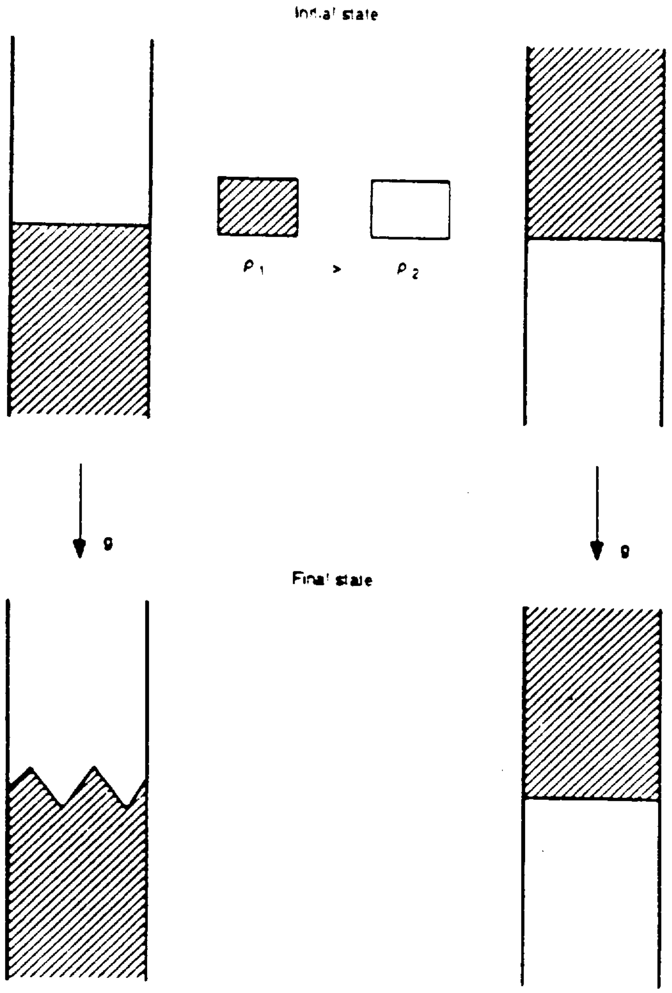


Figure 7: Rayleigh-Taylor instability of accelerated interfaces.

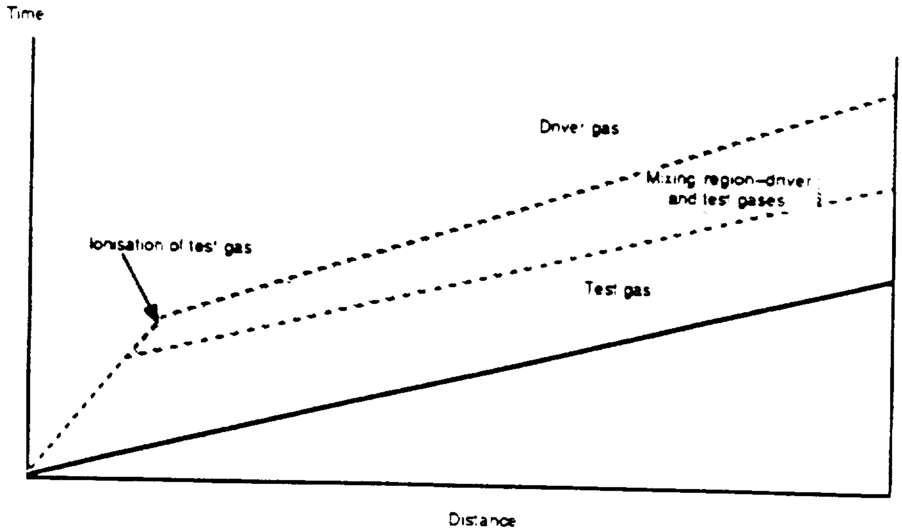


Figure 8: Wave diagram of development of mixing region.

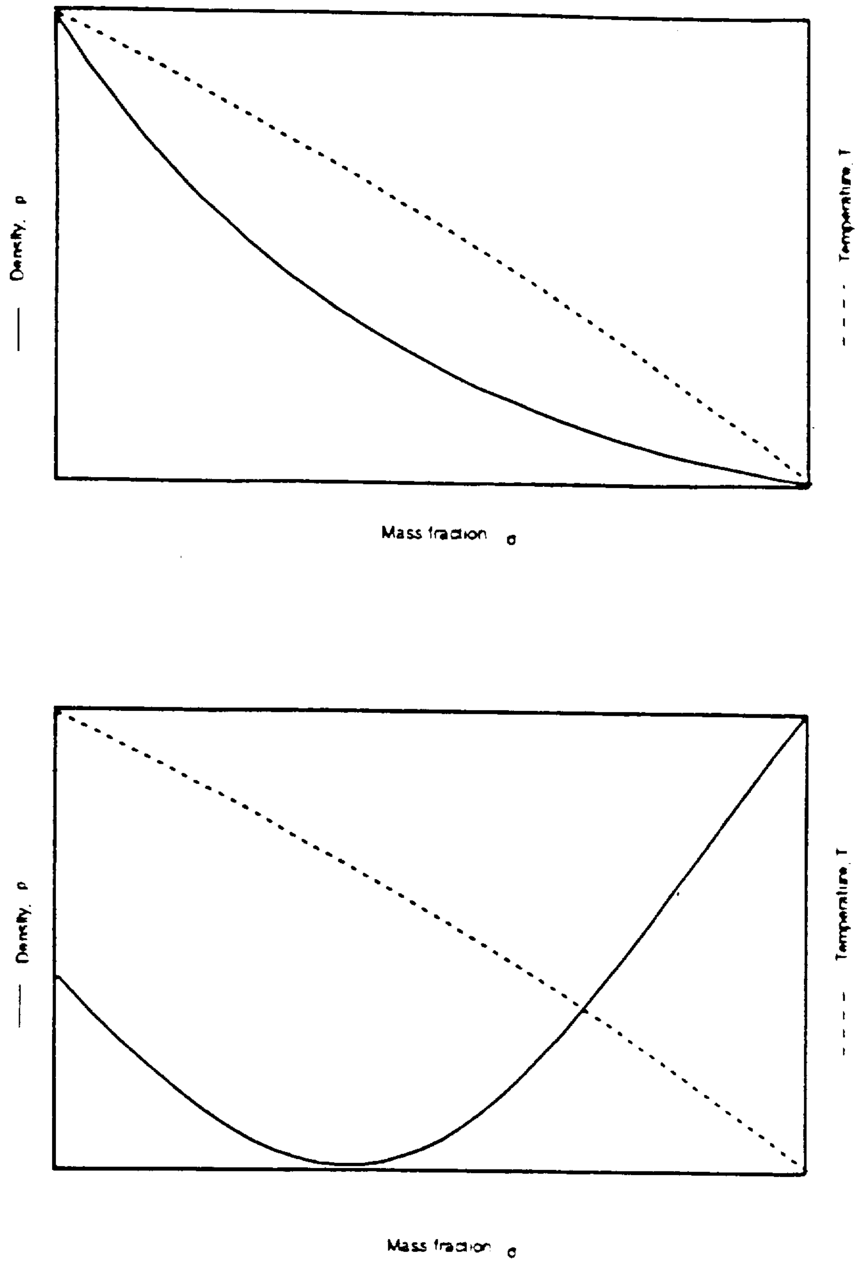


Figure 9: Profiles showing the minimum density in the mixing region

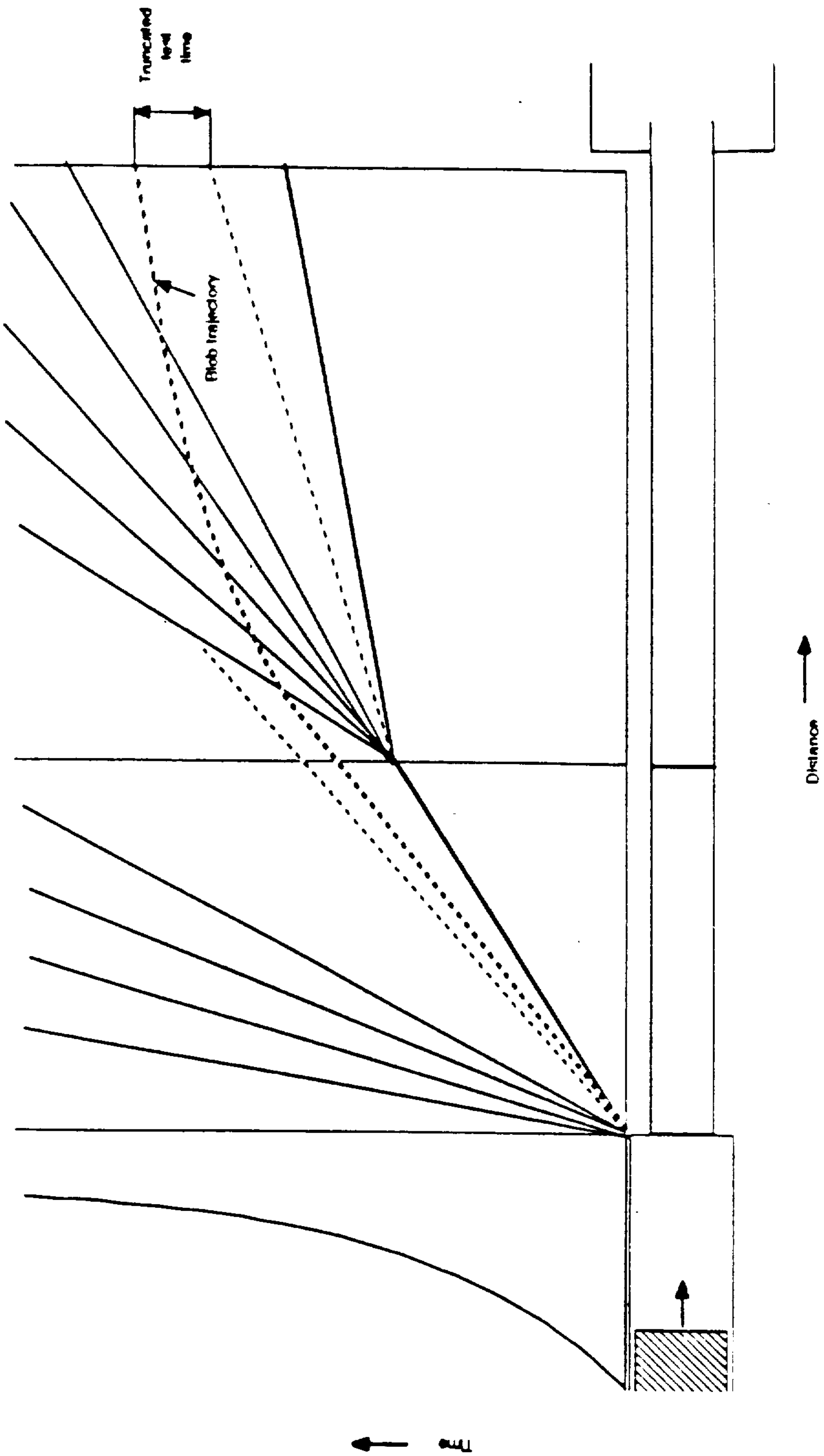


Figure 10: Wave diagram of time of arrival of blob at test section.

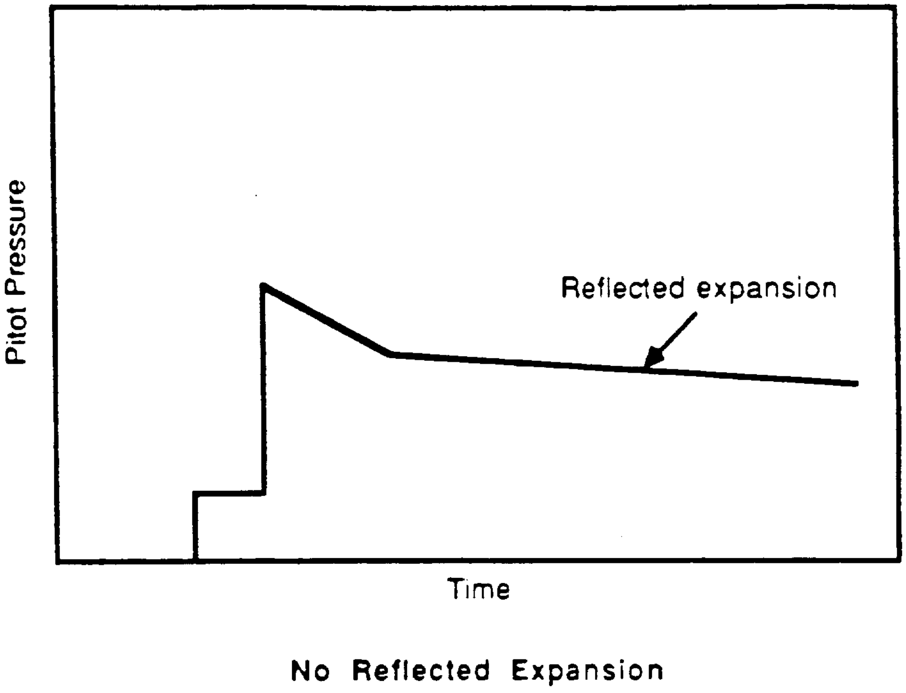


Figure 11: Ideal pitot-pressure trace showing affect of reflected head of expansion.

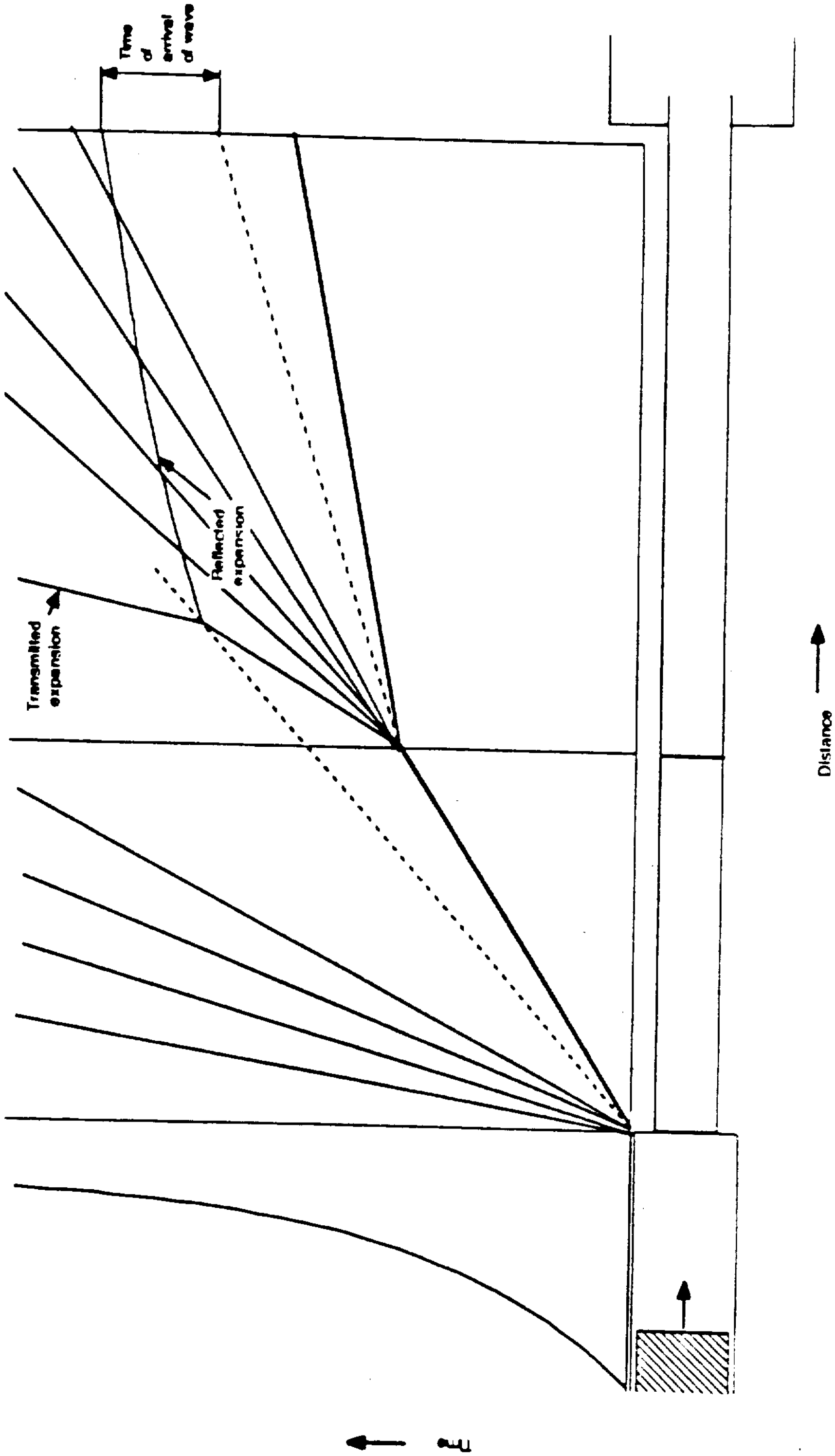


Figure 12: Wave diagram of of reflected head of expansion.

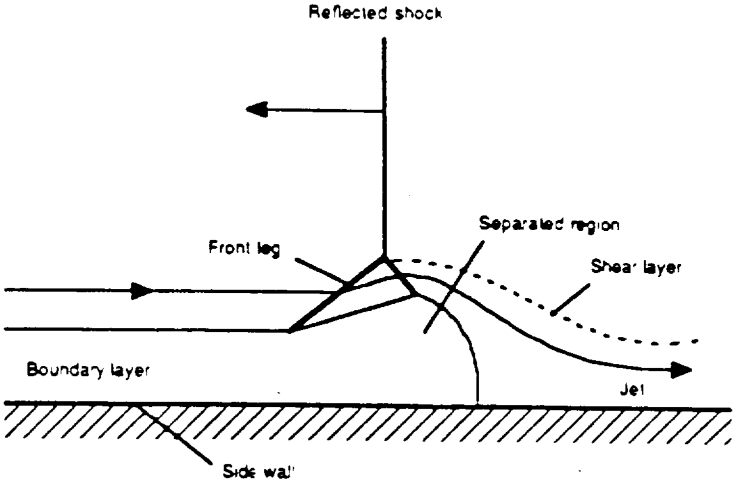


Figure 13: Reflected shock bifurcation.

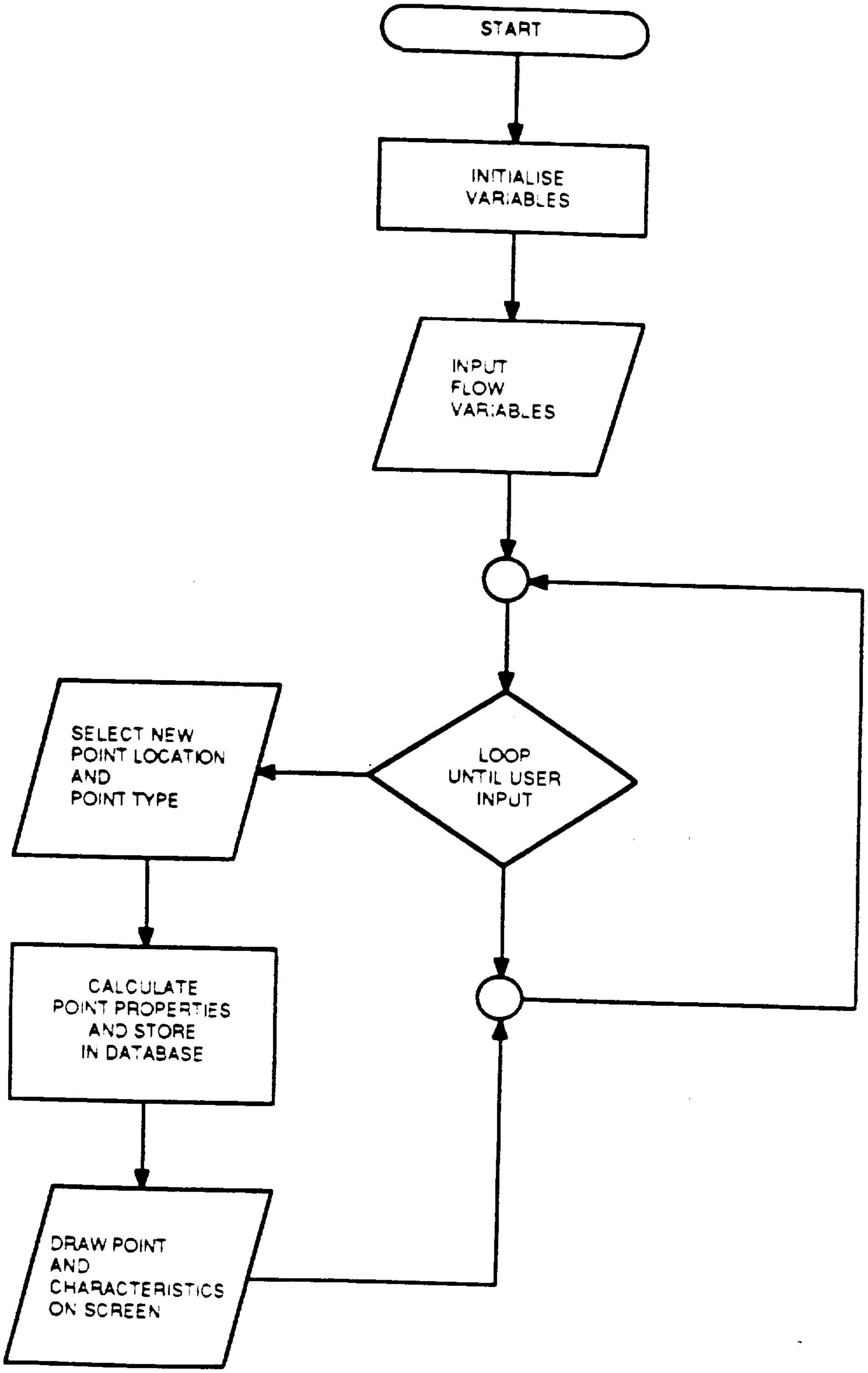

START
INITIALISE VARIABLES
INPUT FLOW VARIABLES
LOOP UNTIL USER INPUT
SELECT NEW POINT LOCATION AND POINT TYPE
CALCULATE POINT PROPERTIES AND STORE IN DATABASE
DRAW POINT AND CHARACTERISTICS ON SCREEN

Figure 14: Computer program flow chart.

ORIGINAL PAGE IS
OF POOR QUALITY

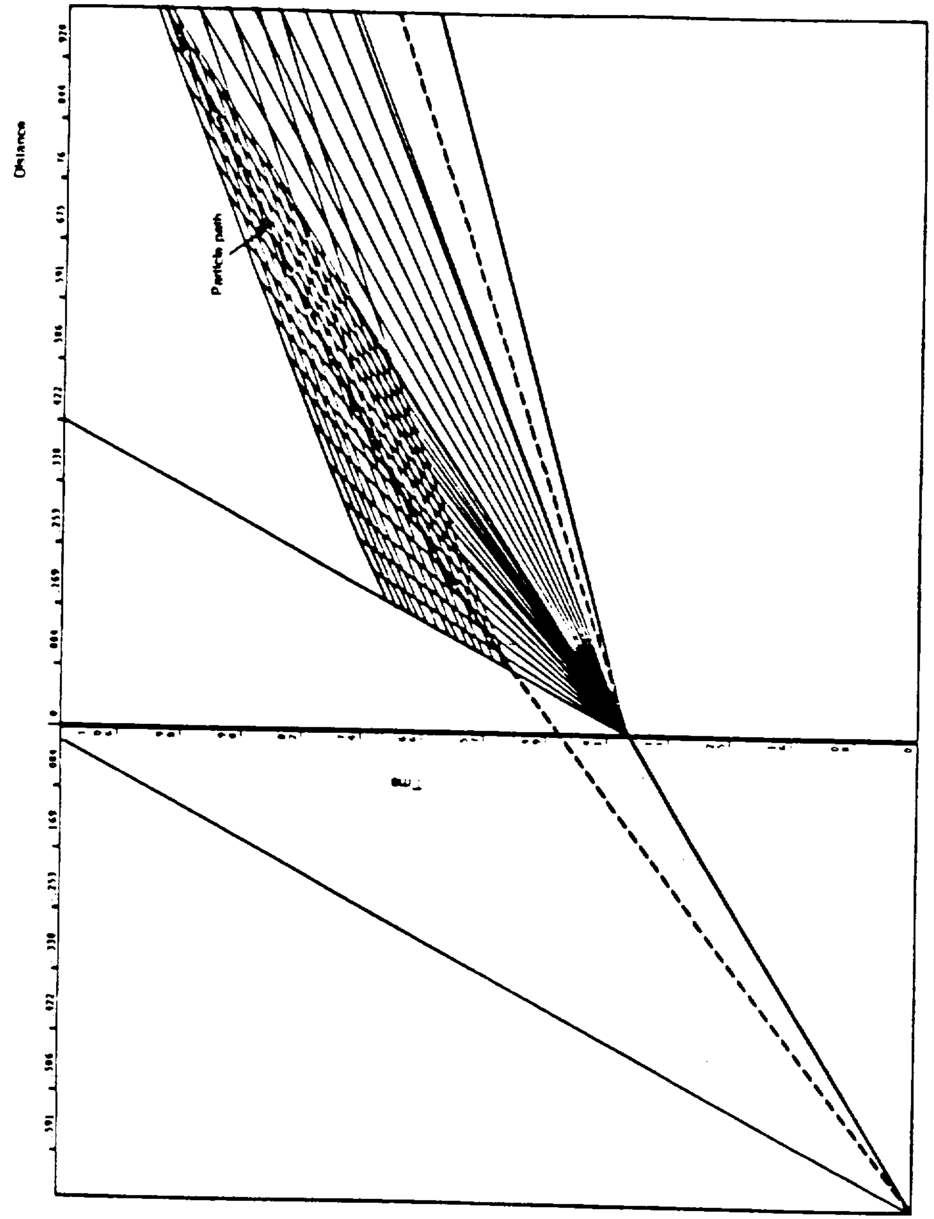
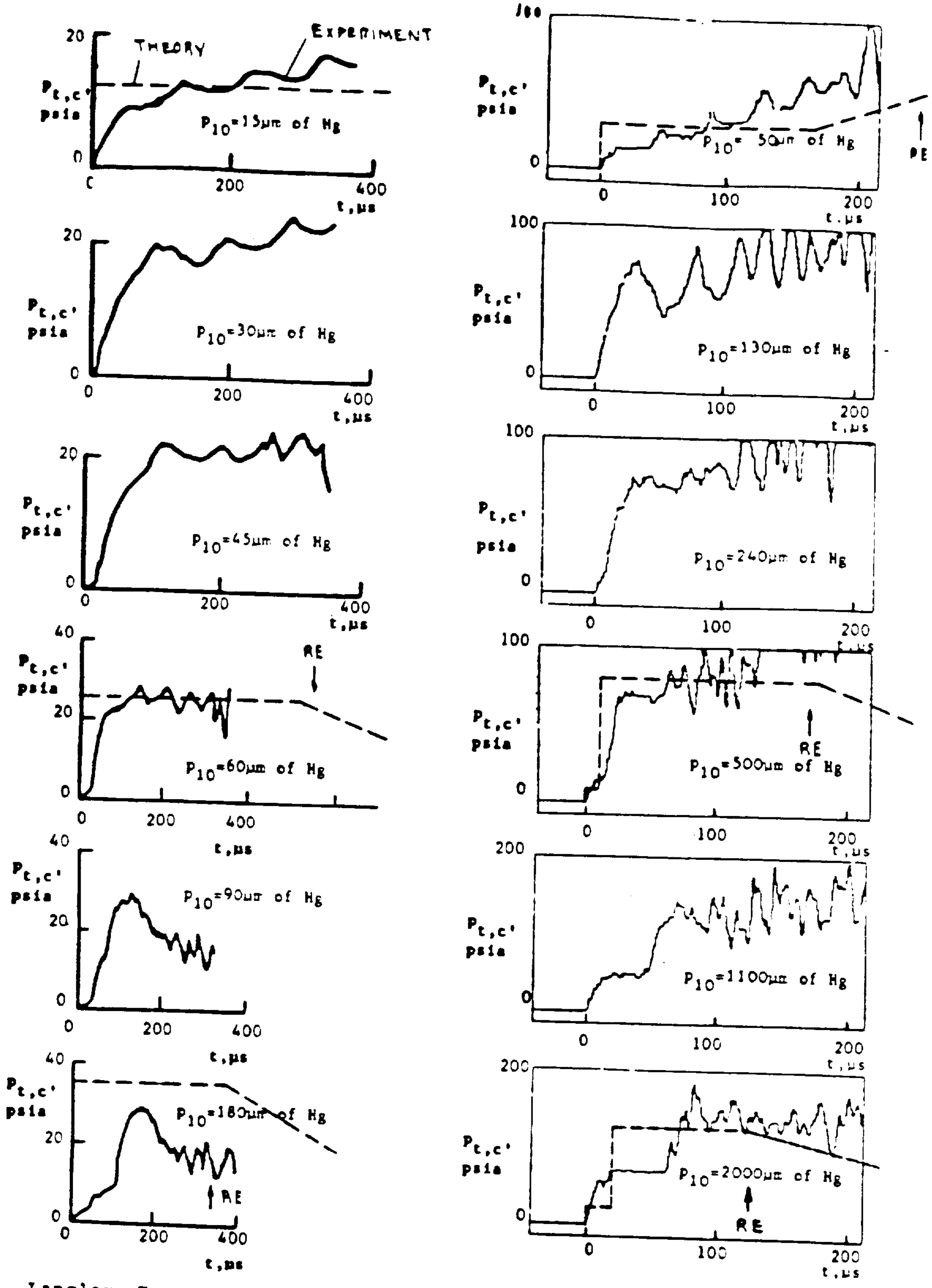


Figure 15: Analytical particle trajectory.



Langley Expansion Tube

P_1 = 3.4 kPa,

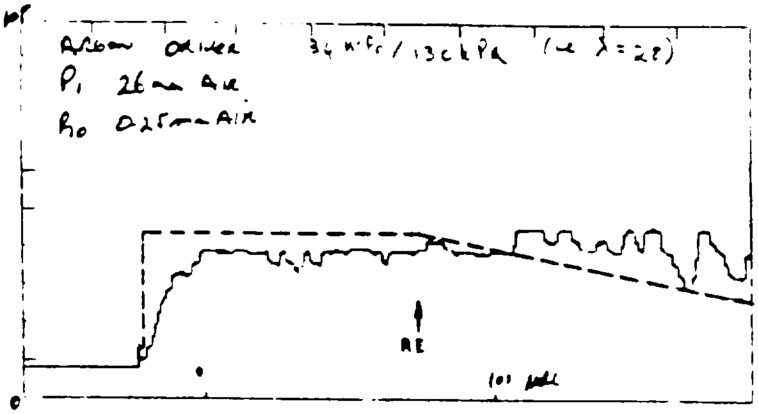
Helium driver.

TQ Expansion Tube

P_1 = 13.7 kPa,

Argon driver.

Figure 16 (a) & 17 (a): Langley (helium) and TQ (argon) pitot-pressures and predictions. (Note 'RE' = arrival time of reflected expansion).



'RE' = REFLECTED EXPANSION

Figure 16 (a) & 17 (a): Langley (helium) and TQ (argon) pitot-pressures and predictions. (Note 'RE' = arrival time of reflected expansion).

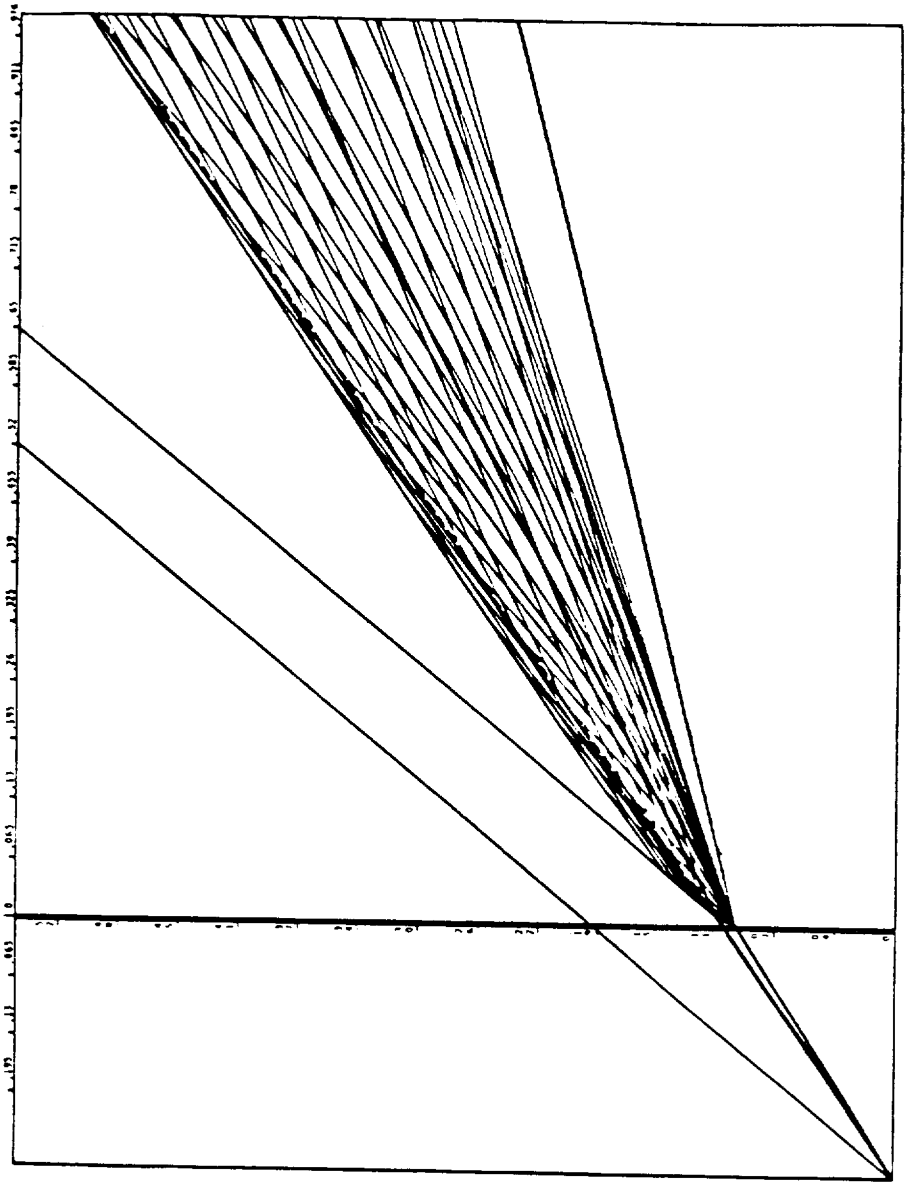


Figure 16 (b): Langley wave diagram, p_{10} = 15 μm Hg.

ORIGINAL PAGE IS
OF POOR QUALITY

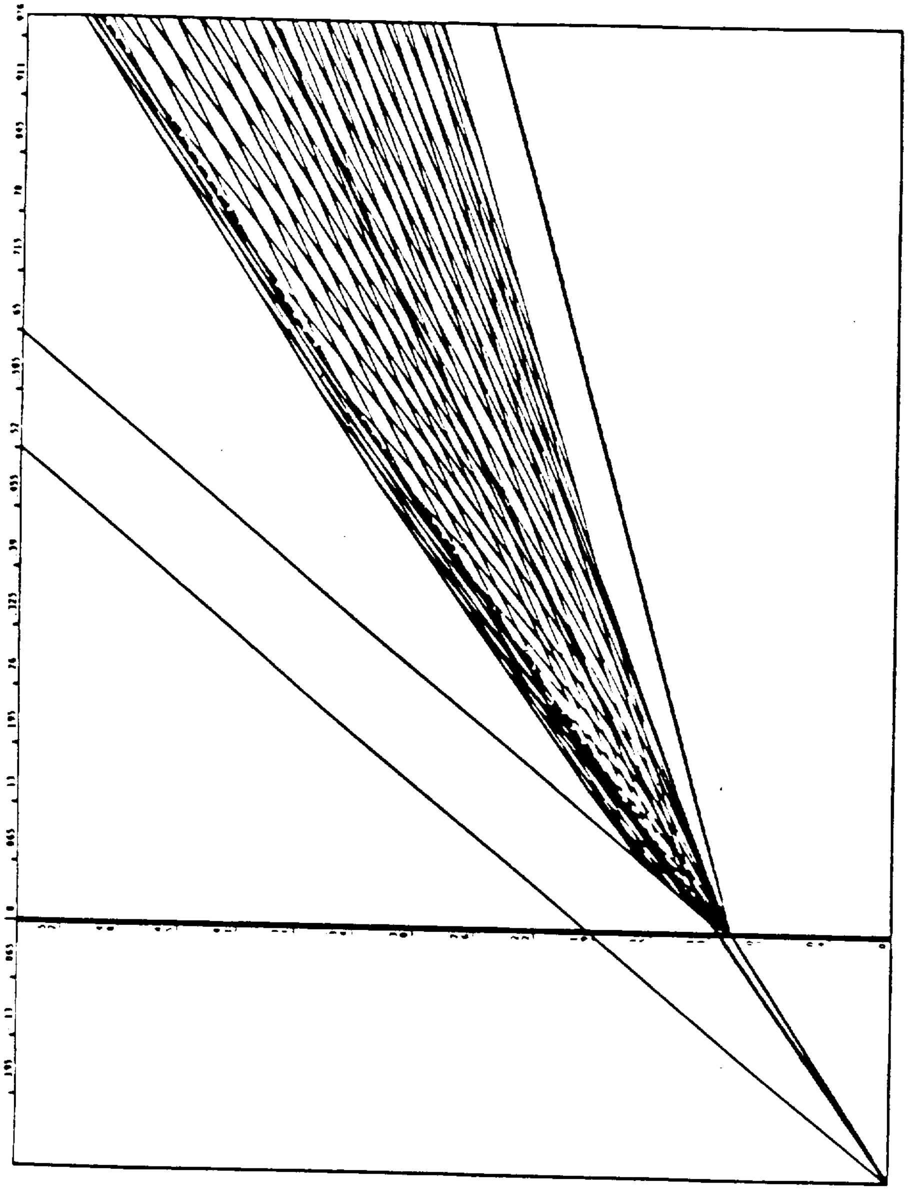


Figure 16 (c): Langley wave diagram, p_{1} = 60 μm Hg.

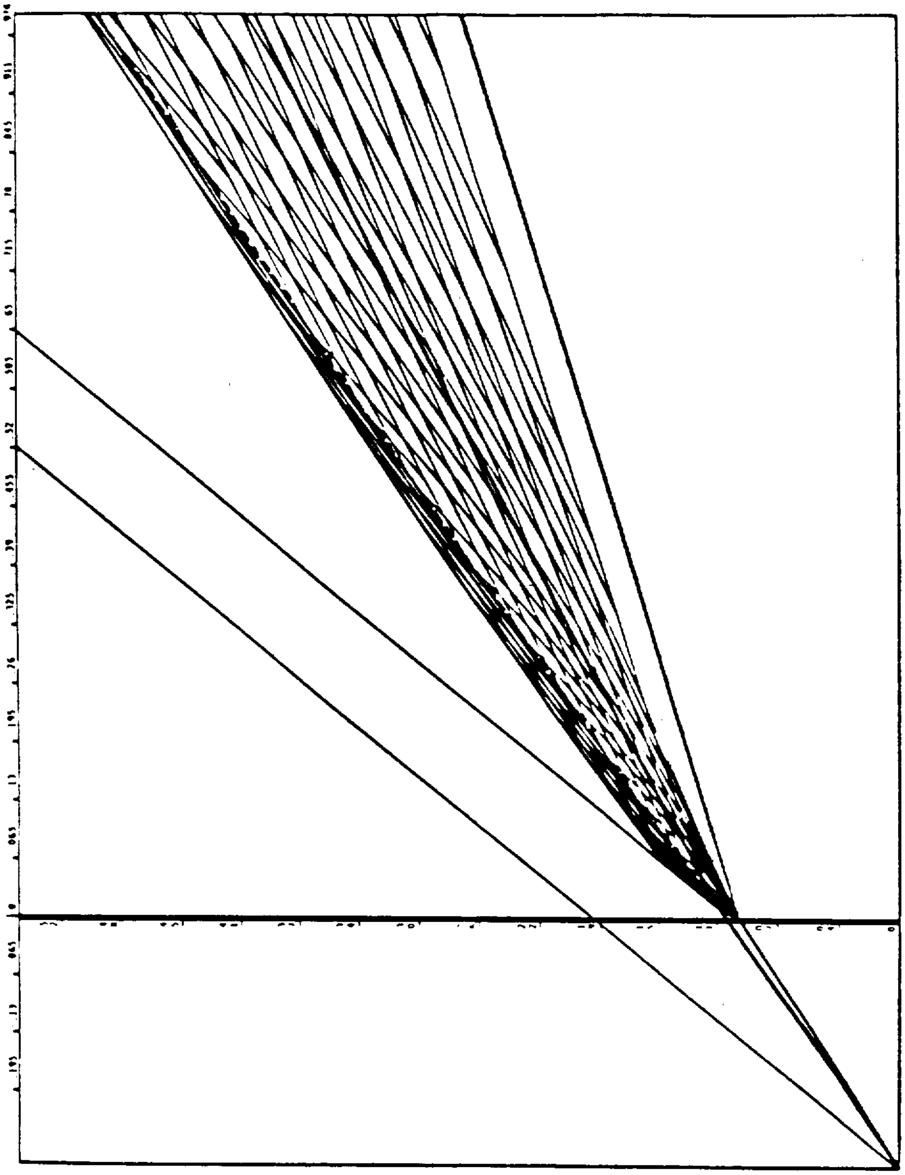


Figure 16 (d): Langley wave diagram, p_{1c} = 180 µm Hg.

ORIGINAL PAGE IS
OF POOR QUALITY

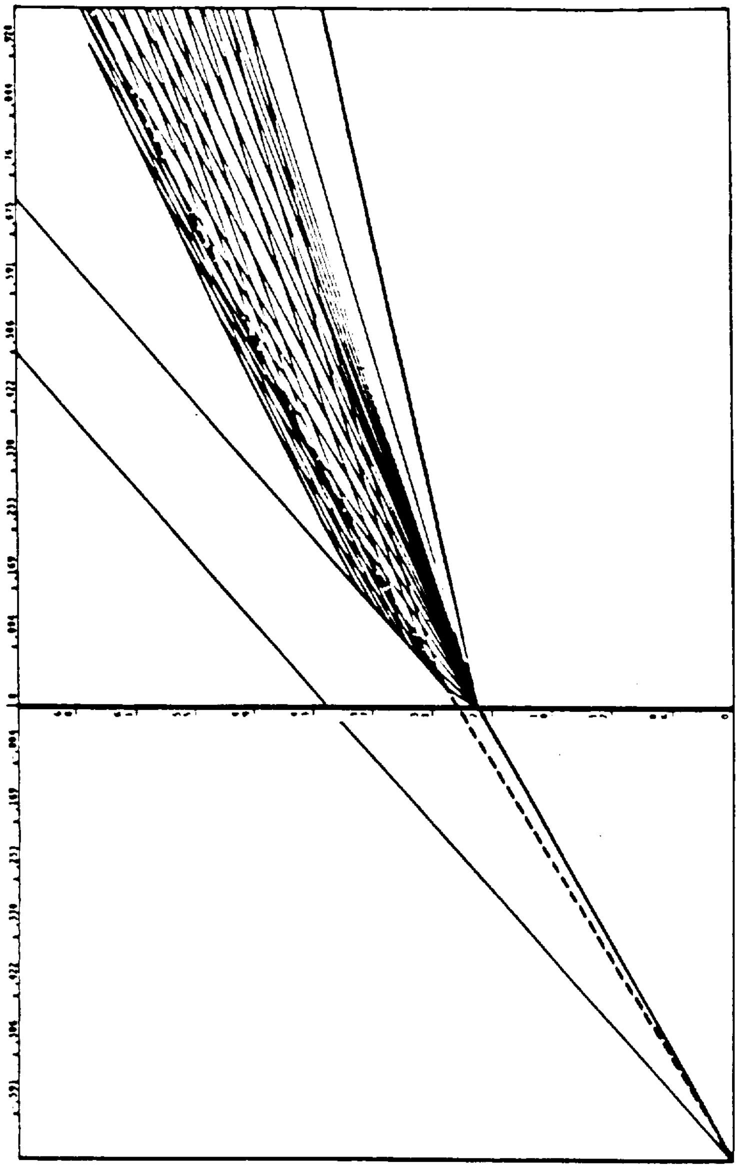


Figure 17 (b): TQ wave diagram, argon, p_1 = 13.7 kPa,

p_{10} = 50 μm Hg.

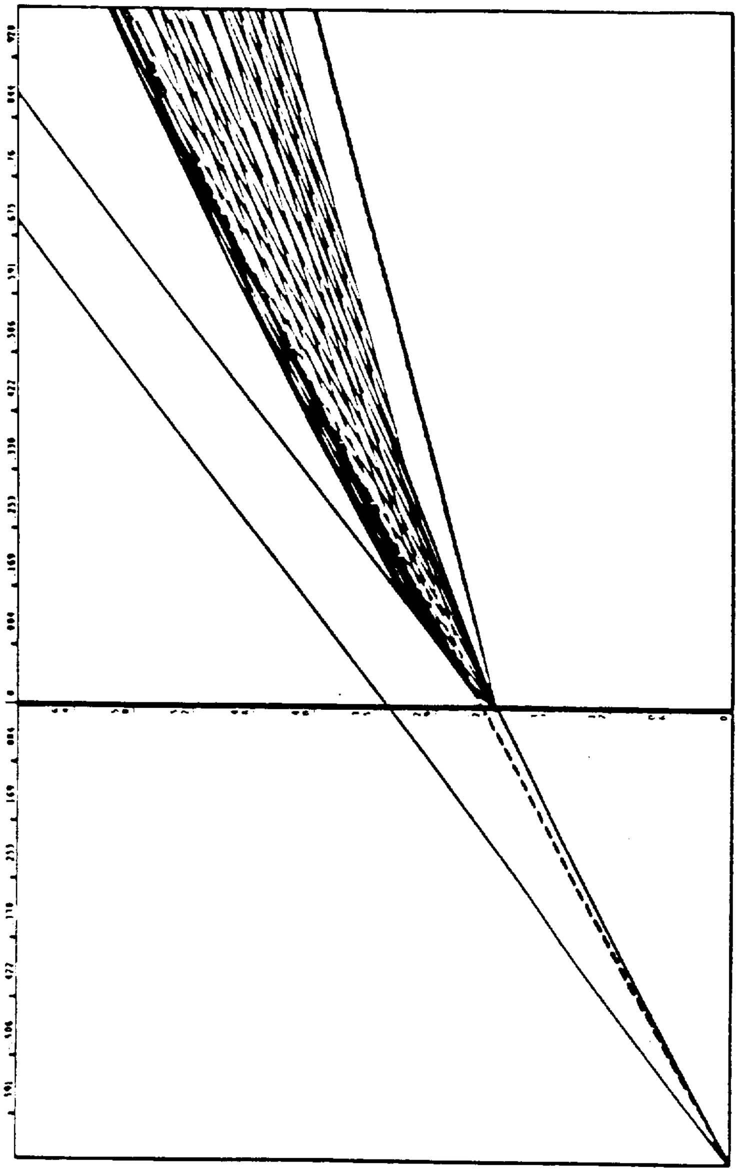


Figure 17 (c): TQ wave diagram, argon, p_1 = 3.5 kPa, p_{10} = 250 µm Hg.

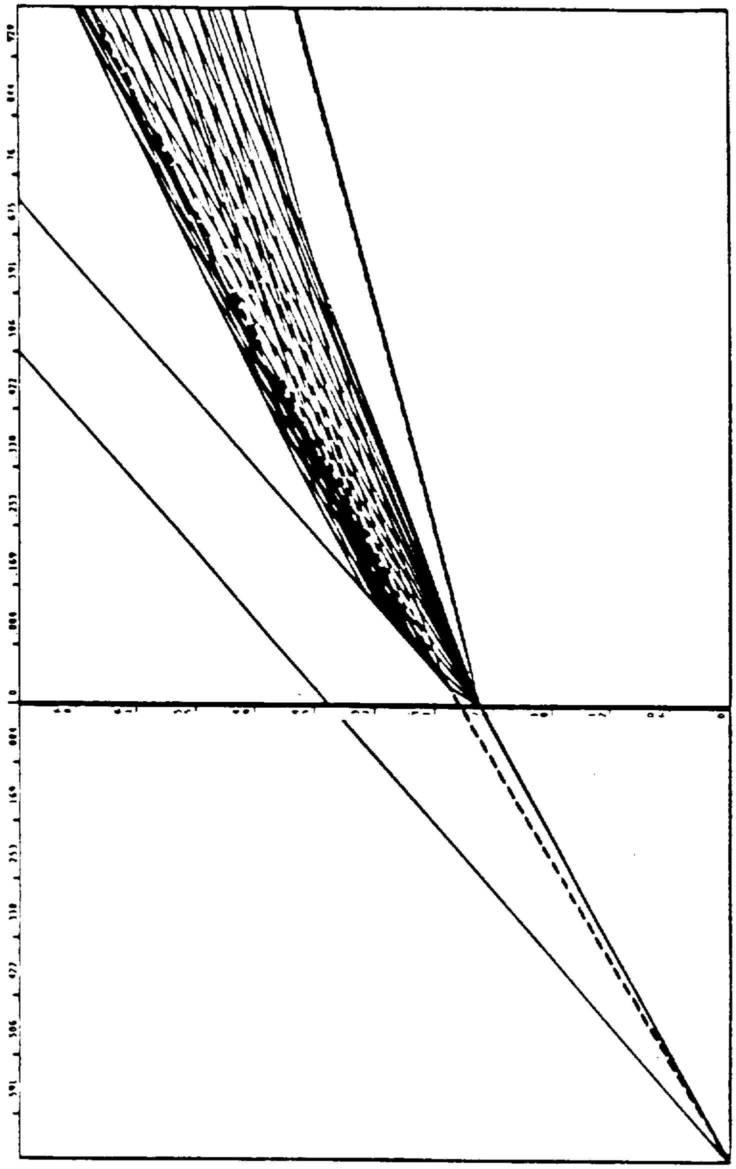


Figure 17 (d): TQ wave diagram, argon, p_1 = 13.7 kPa,

p_{10} = 500 μm Hg.

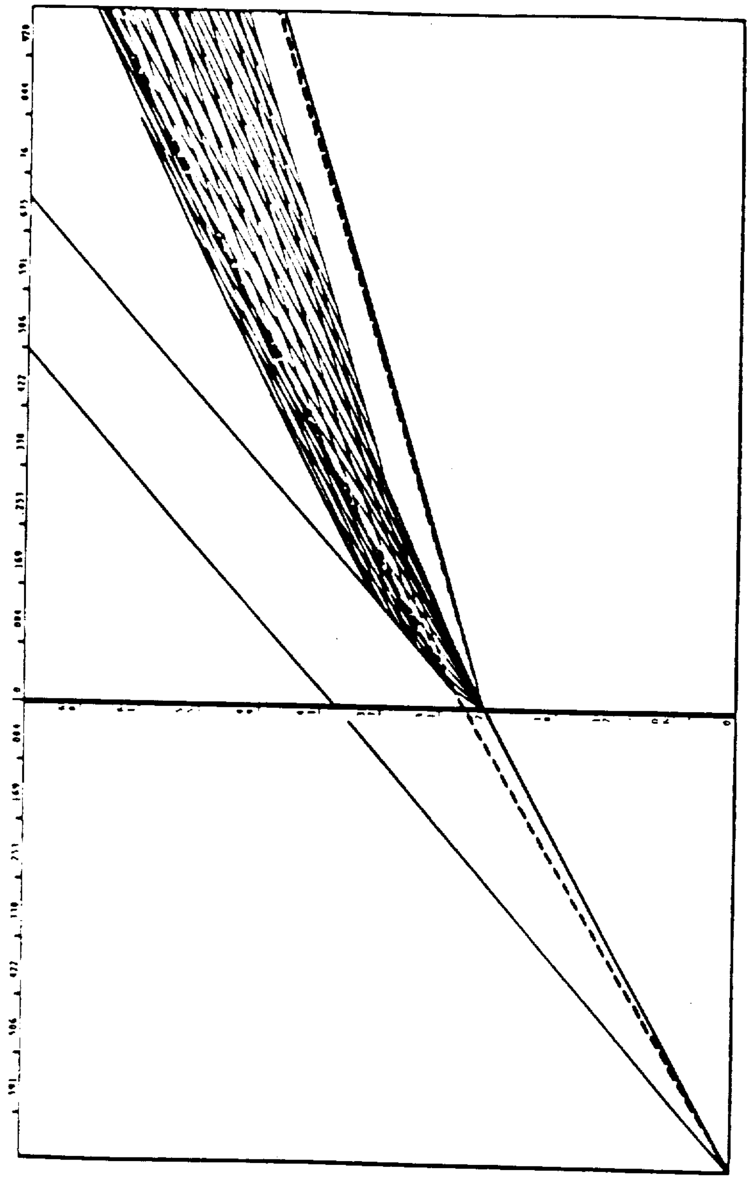
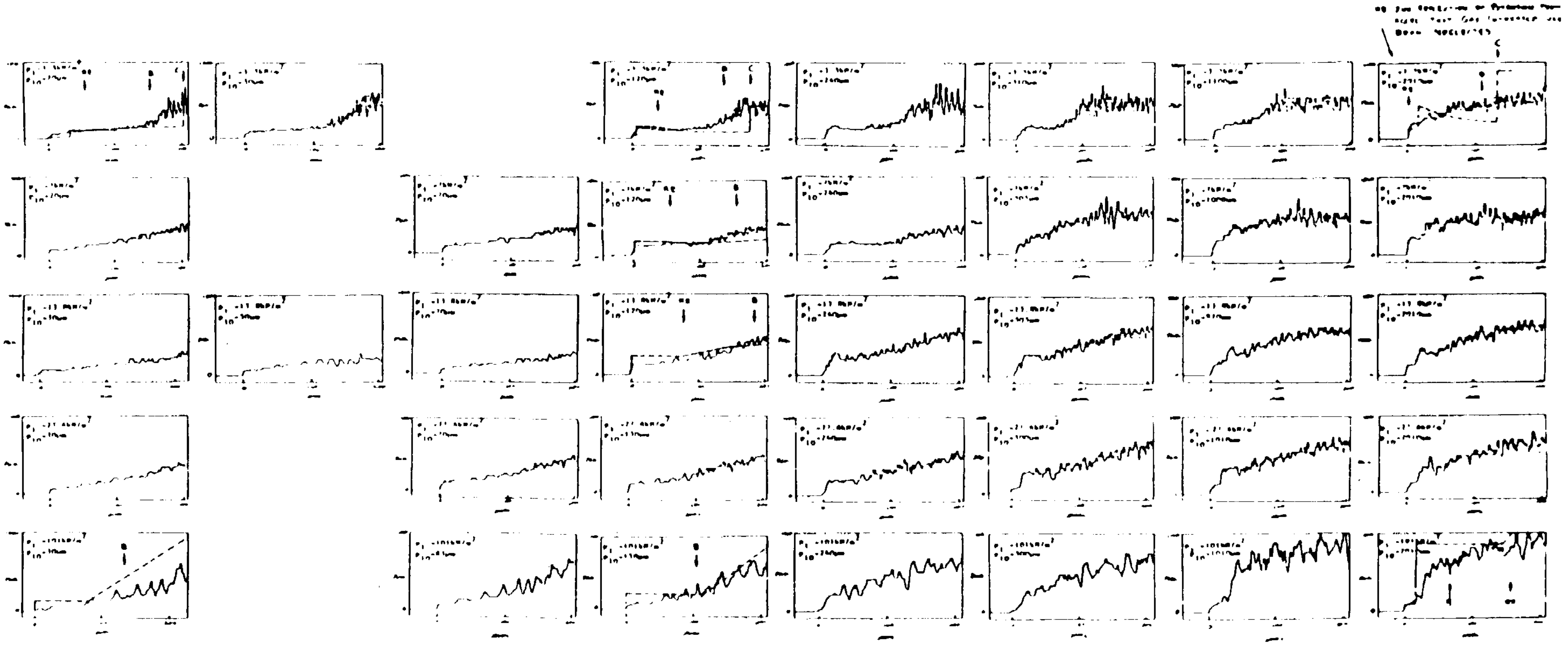


Figure 17 (e): TC wave diagram, argon, P_1 = 13.7 kPa,

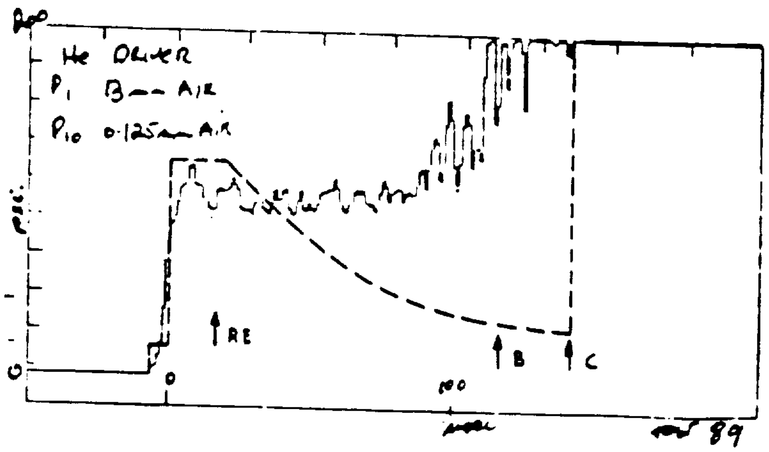
P_{10} = 2000 μm Hg.



Helium driver, λ = 29, P_r = 34.5 MPa.

RE = Reflected Expansion
C = Contact Surface
B = Bulk of Light Gas

Figure 18 (a): TQ pitot pressures and predictions for helium test gas.



'RE' = REFLECTED EXPANSION

'C' = CONTACT SURFACE

'B' = BLOB OF LIGHT GAS

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 18 (a): TQ pitot-pressures and predictions for helium test gas.

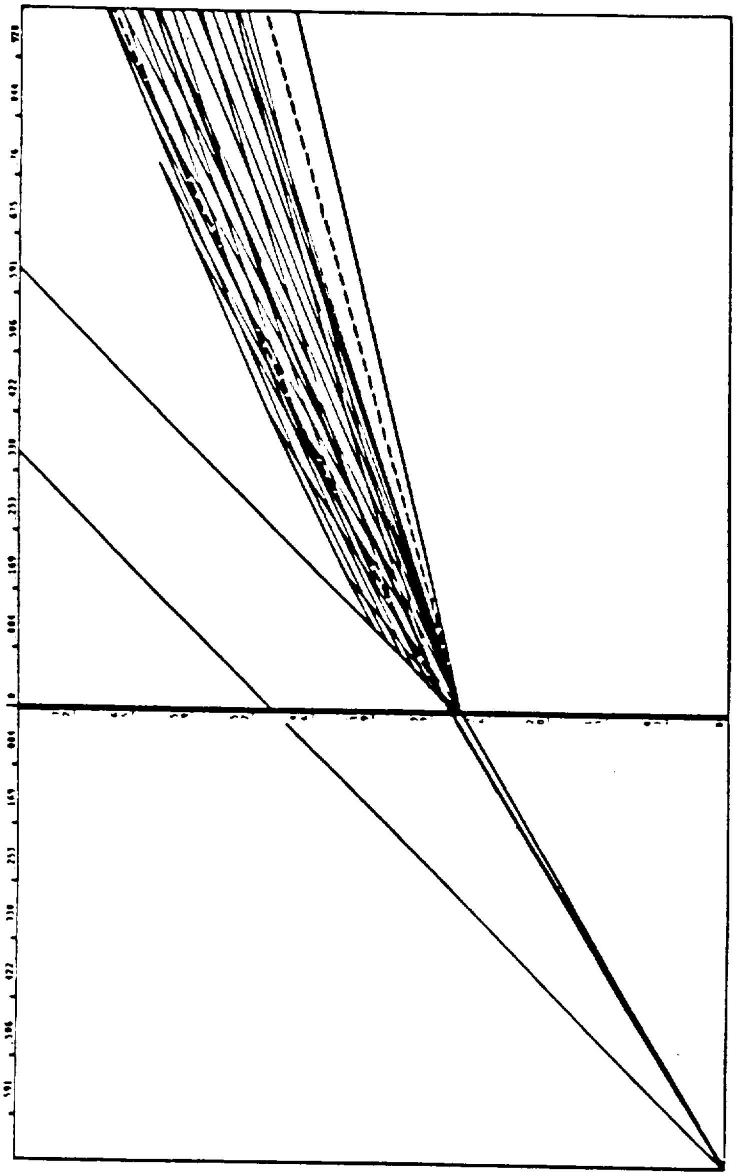


Figure 18 (b): TQ wave diagram, helium, p_1 = 3.5 kPa, p_{1c} = 20 µm Hg.

ORIGINAL PAGE IS
OF POOR QUALITY

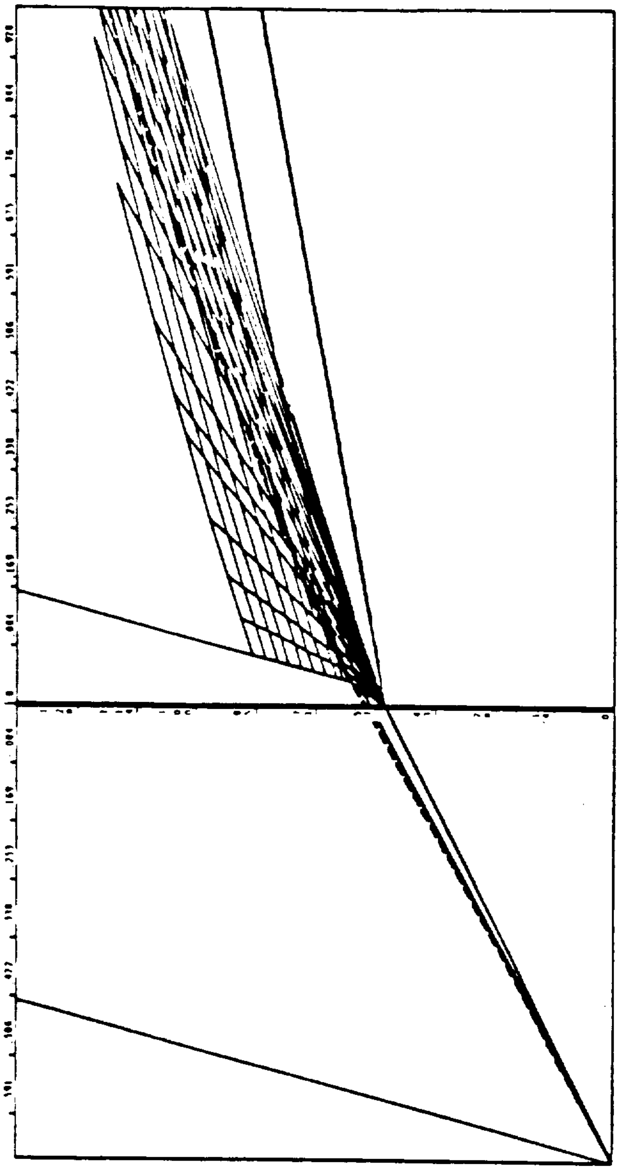


Figure 18 (c): TQ wave diagram, helium, p_1 = 101.0 kPa,

p_{10} = 30 μm Hg.

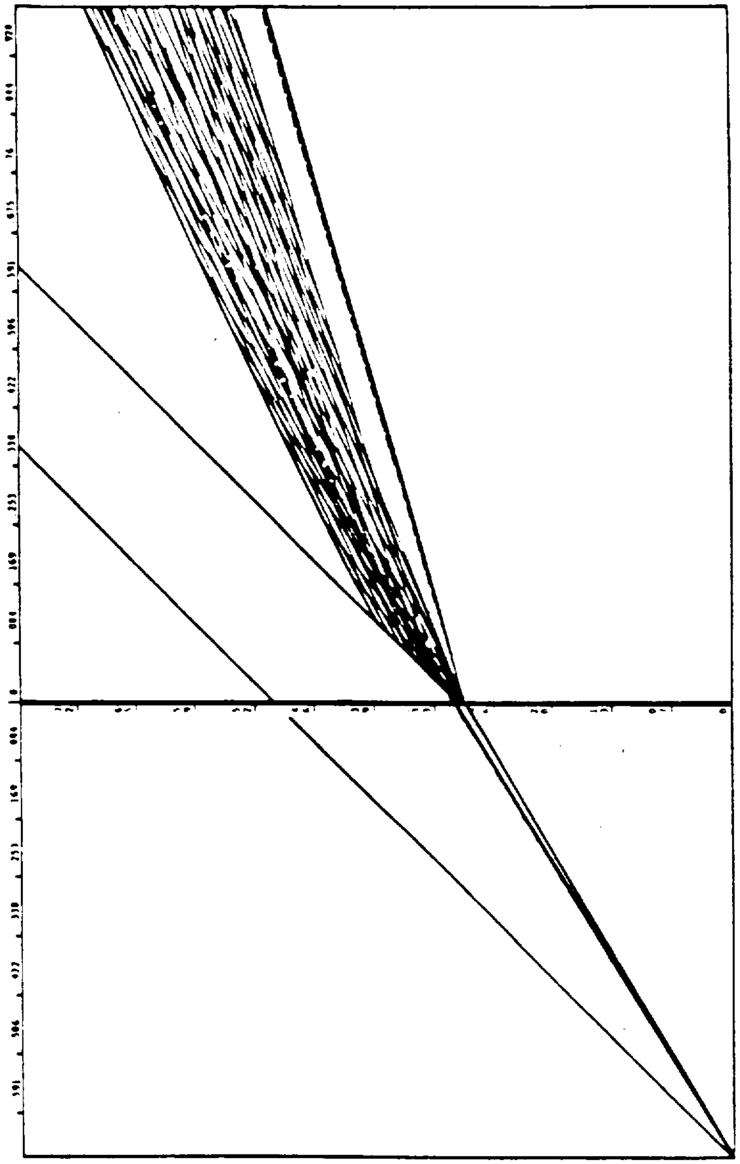


Figure 18 (e): TQ wave diagram, helium, p_1 = 3.5 kPa, p_{10} = 120 µm Hg.

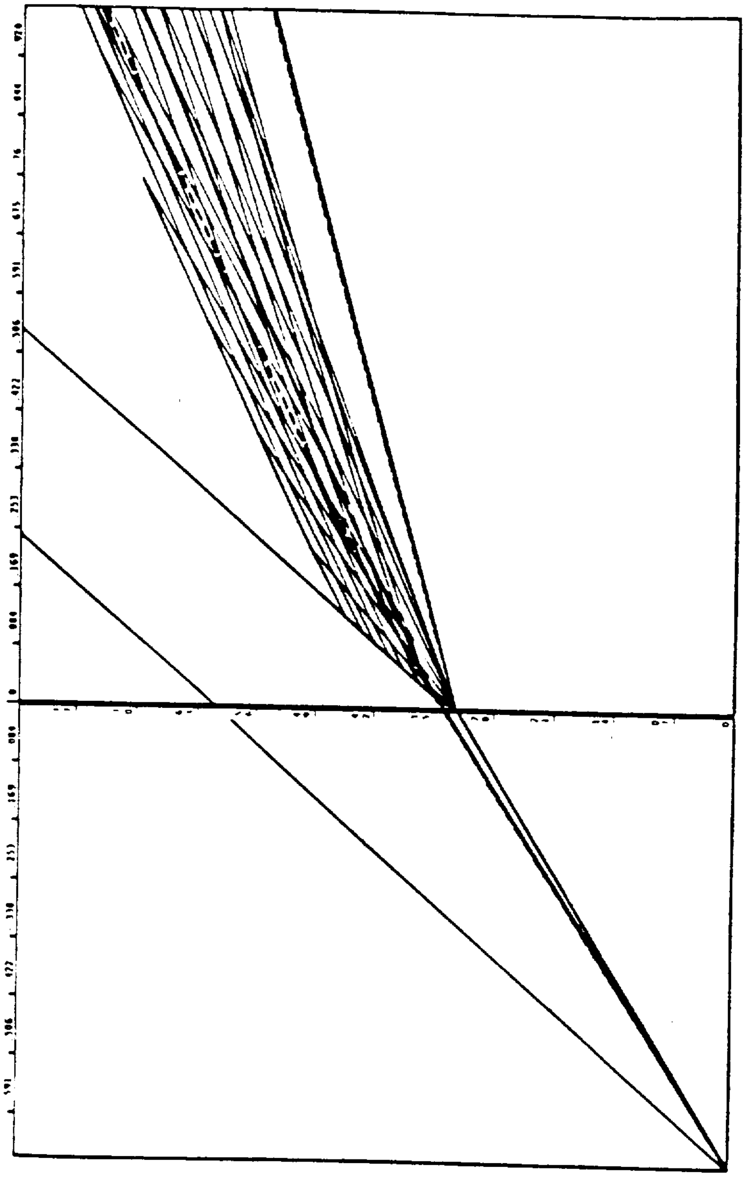


Figure 18 (f): TQ wave diagram, helium, p_1 = 7.0 kPa,

p_{10} = 120 μm Hg.

ORIGINAL PAGE IS
OF POOR QUALITY

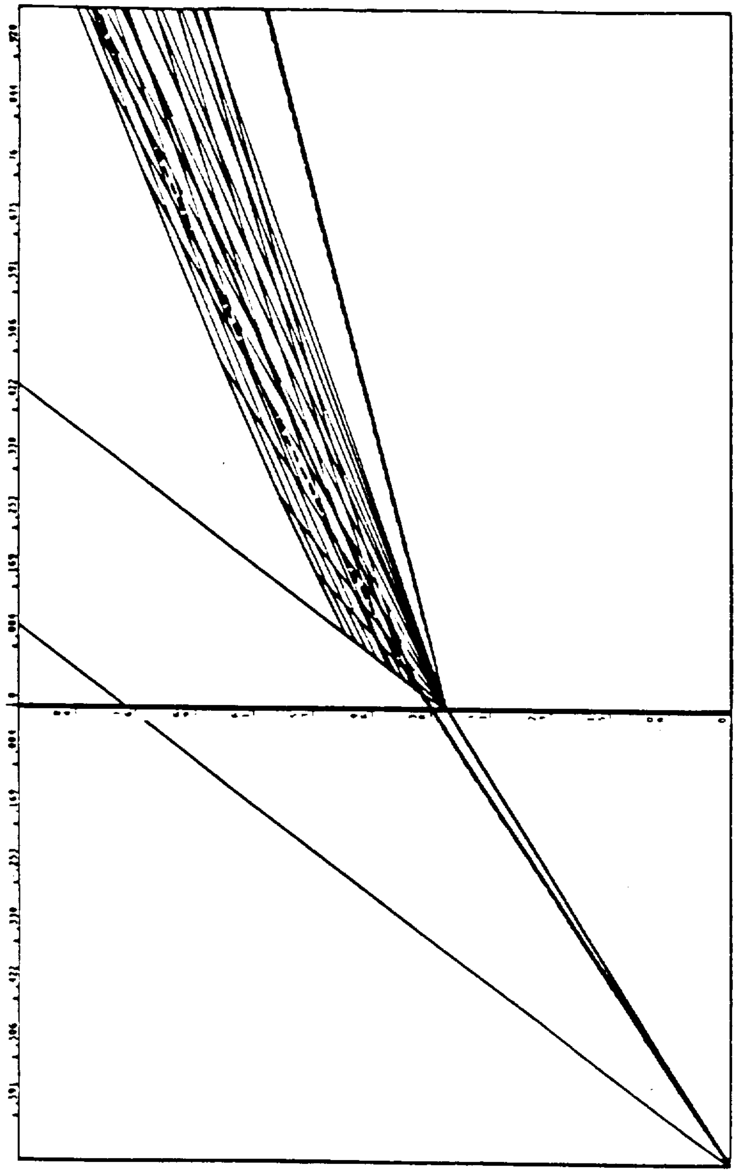


Figure 18 (g): TQ wave diagram, helium, p_1 = 13.8 kPa,

p_{10} = 120 µm Hg.

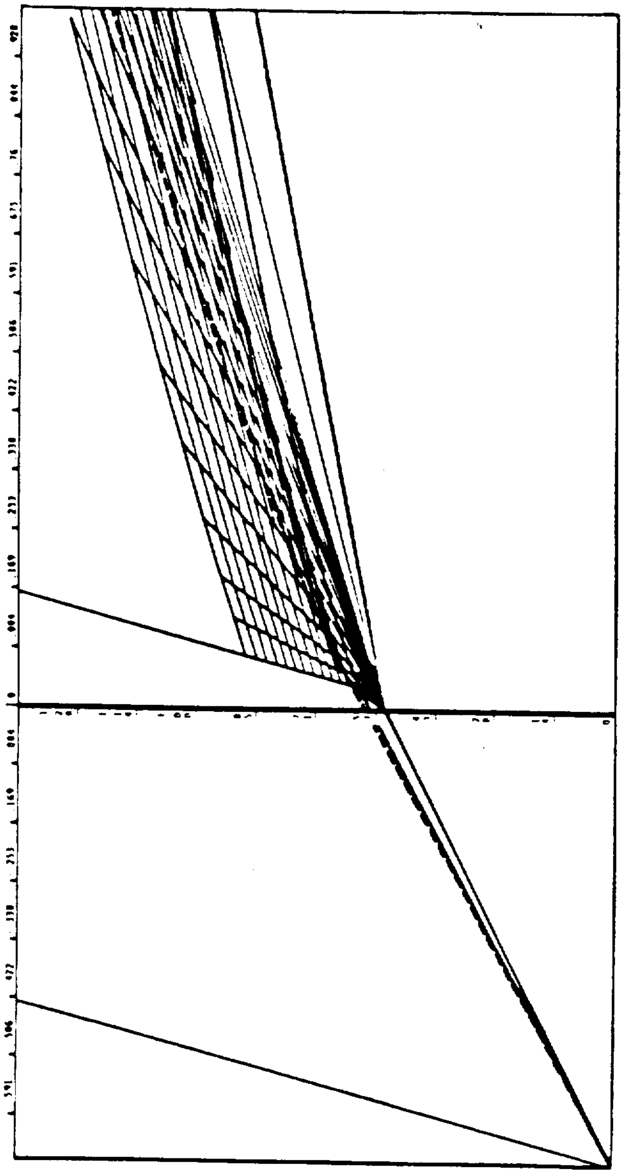


Figure 18 (b): TQ wave diagram, helium, p_1 = 101.0 kPa, p_{10} = 150 μm Hg.

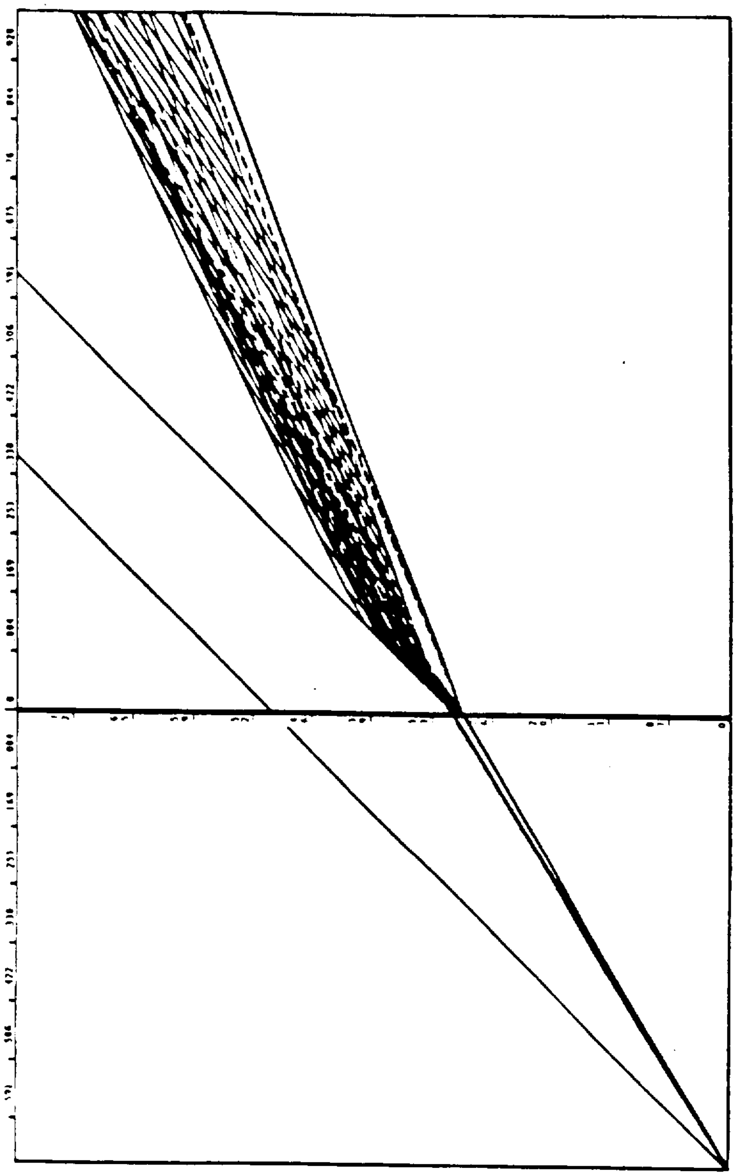


Figure 18 (i): TQ wave diagram, helium, p_4 = 3.5 kPa,

P_1 = 2010 μm Hg.

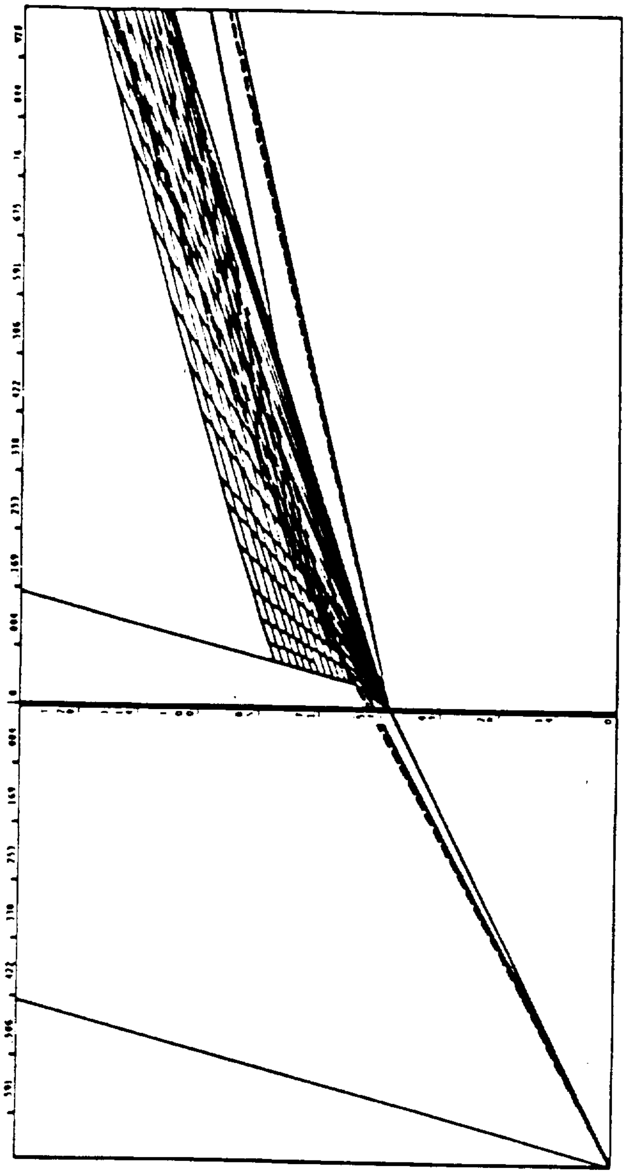
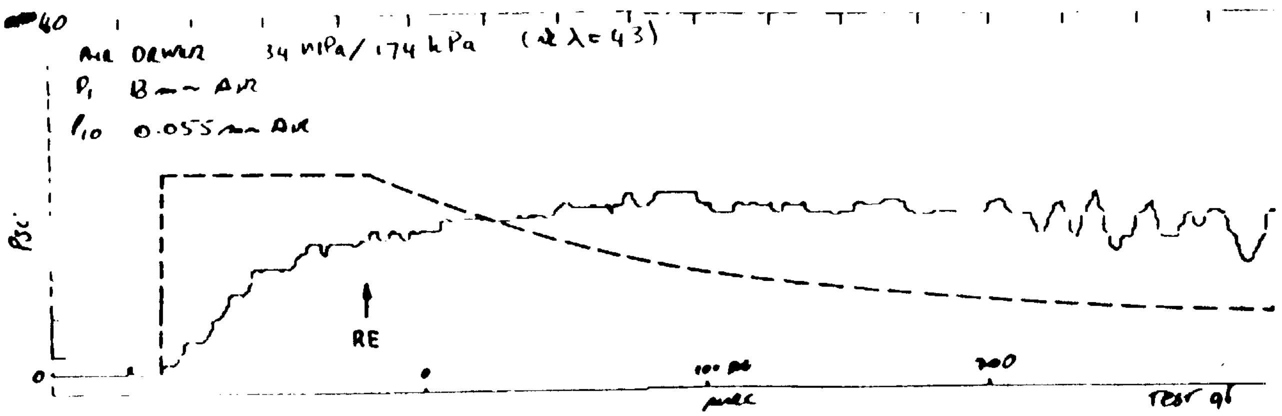


Figure 18 (j): TQ wave diagram, helium, p_1 = 101.0 kPa,

p_{10} = 2010 μm Hg.



'RE' = REFLECTED EXPANSION

Figure 19 (a): TQ pitot-pressures and predictions for air test gas.

ORIGINAL PAGE IS
OF POOR QUALITY

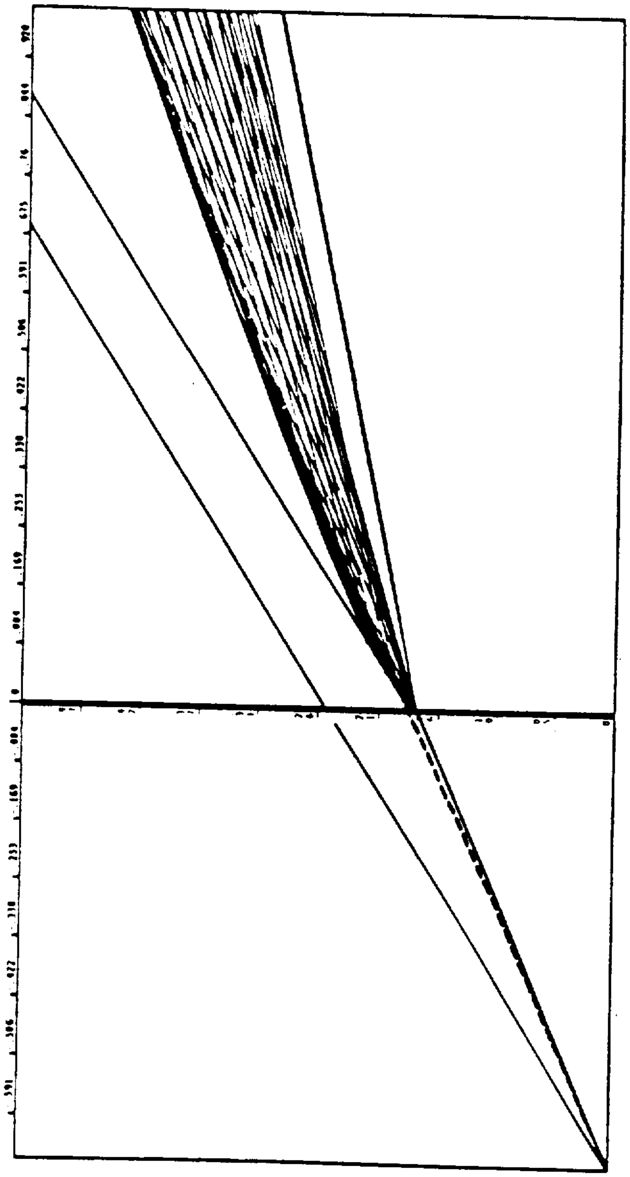


Figure 19 (b): TQ wave diagram, air.

APPENDICES

A. Complete Set of Finite Difference Equations

A.1 *Non-Dimensionalisation of Variables*

The reference conditiosn chosen for the wave diagram are the acceleration tube length, the diaphragm rupture pressure and the speed of sound in the driver gas prior to expansion.

A.2 *Equations for Ideal Expansion Tube Flow*

Shock Tube Section Flow

References: Stalker (1964)
Liepmann and Roskho (1957)

$$\frac{p_4}{p_1} = \frac{p_2}{p_1}\left[\sqrt{\frac{\gamma_4 + 1}{2}} - \frac{(\gamma_4 - 1)(A_1/A_4)(p_2/p_1 - 1)}{\sqrt{2\gamma_1}\sqrt{2\gamma_1 + (\gamma_1 + 1)(p_2/p_1 - 1)}}\right]^{\frac{-2\gamma_4}{\gamma_4 - 1}}$$

$$\frac{p_3}{p_4} = \frac{p_2/p_1}{p_4/p_1}$$

$$\frac{T_3}{T_4} = \left(\frac{p_3}{p_4}\right)^{\frac{\gamma_4 - 1}{\gamma_4}}$$

$$\frac{T_2}{T_1} = \frac{1 + \frac{\gamma_1 - 1}{\gamma_1 + 1}\frac{p_2}{p_1}}{1 + \frac{\gamma_1 - 1}{\gamma_1 + 1}\frac{p_1}{p_2}}$$

$$M_2 = \frac{1}{\gamma_1}\left(\frac{p_2}{p_1} - 1\right)\left[\frac{p_2}{p_1}\left(\frac{\gamma_1 + 1}{2\gamma_1} + \frac{\gamma_1 - 1}{2\gamma_1}\frac{p_2}{p_1}\right)\right]^{-\frac{1}{2}}$$

$$M_3 = \frac{2}{\gamma_4 - 1}\left[\left(\frac{p_4/p_1}{p_2/p_1}\right)^{\frac{\gamma_4 - 1}{2\gamma_4}} \cdot \sqrt{\frac{\gamma_4 + 1}{2}} - 1\right]$$

$$A_2 = \sqrt{\gamma_1 R_1 T_2}$$

$$A_3 = \sqrt{\gamma_4 R_4 T_3}$$

$$U_2 = M_2 a_2$$

$$U_3 = M_3 a_3$$

$$\frac{\rho_2}{\rho_1} = \frac{T_1}{T_2}\frac{p_2}{p_1}$$

$$\frac{\rho_3}{\rho_4} = \frac{T_4}{T_3}\frac{p_3}{p_4}$$

Expansion Tube Section Flow

Reference: Liepmann and Roskho (1957)

$$\frac{p_2}{p_{10}} = \frac{p_{20}}{p_{10}}\left[1 + \frac{\gamma_1 - 1}{2} M_2 - \frac{(\gamma_1 - 1)(A_{10}/A_2)(p_{20}/p_{10} - 1)}{\sqrt{2\gamma_1}\sqrt{2\gamma_1 + (\gamma_1 + 1)(p_{20}/p_{10} - 1)}}\right]^{\frac{-2\gamma_1}{\gamma_1 - 1}}$$

$$\frac{p_5}{p_2} = \frac{p_{20}/p_{10}}{p_2/p_{10}}$$

$$\frac{T_5}{T_2} = \left(\frac{p_5}{p_2}\right)^{\frac{\gamma_1 - 1}{\gamma_1}}$$

$$\frac{T_{20}}{T_{10}} = \frac{1 + \frac{\gamma_1 - 1}{\gamma_1 + 1}\frac{p_{20}}{p_{10}}}{1 - \frac{\gamma_1 - 1}{\gamma_1 + 1}\frac{p_{10}}{p_{20}}}$$

$$M_{20} = \frac{1}{\gamma_1}\left(\frac{p_{20}}{p_{10} - 1}\right)\left[\frac{p_{20}}{p_{10}}\left(\frac{\gamma_1 + 1}{2\gamma_1} + \frac{\gamma_1 - 1}{2\gamma_1}\frac{p_{20}}{p_{10}}\right)\right]^{-\frac{1}{2}}$$

$$M_5 = \frac{2}{\gamma_1 - 1}\left[\left(\frac{p_2/p_{10}}{p_{20}/p_{10}}\right)^{\frac{\gamma_1 - 1}{2\gamma_1}}\left(1 + \frac{\gamma_1 - 1}{2} M_2\right) - 1\right]$$

$$A_{20} = \sqrt{\gamma_1 R_1 T_{20}}$$

$$A_5 = \sqrt{\gamma_1 R_1 T_5}$$

$$U_{20} = M_{20} a_{20}$$

$$U_5 = M_5 a_5$$

$$\frac{\rho_{20}}{\rho_{10}} = \frac{T_{10}}{T_{20}}\frac{p_{20}}{p_{10}}$$

$$\frac{\rho_5}{\rho_2} = \frac{T_2}{T_5}\frac{p_5}{p_2}$$

A.3 *Mirels Effect for Laminar or Turbulent Boundary Layers*

Laminar

Reference: Mirels (1963)

The acceleration tube flow is laminar for TQ and partly laminar for Langley. Therefore assume that the maximum separation of the shock and the contact surface has been reached. This only has a cosmetic effect on the wave diagram in the acceleration tube region. It does not affect the results. The effect on the test gas, and blobs, is difficult to determine

ORIGINAL PAGE IS
OF POOR QUALITY

due to the expansion wave thickness and the complex nature of the boundary layer (see Mirels and Mullen, 1964).

$$\frac{l_{m2}}{L_2} = \left(\frac{1}{4\beta_1}\right)^2 \frac{p_{10}}{p_{20}} \frac{\rho_{20}/\rho_{10}}{\rho_{20}/\rho_{10} - 1} M_{s20} \frac{\rho_1 a_1}{\mu_1} \left(\frac{d}{L_2}\right)^2 \frac{p_{10}}{p_1} \frac{p_1}{p_4} \frac{p_4}{p_{st}}$$

$$M_{s0} = u_{s20} \sqrt{\frac{\gamma_4 R_4 T_4}{\gamma_1 R_1 T_1}}$$

M_s	β_1 (Air p_1 = 0.001 atm)
6	0.0283
8	0.0220
10	0.0188
12	0.0157
14	0.0129
16	0.0116
18	0.0104
20	0.0094

$$-\frac{X_2}{2} = \ln\left(1 - T_2^{\,n}\right) + T_2^{\,n}, \quad n = \frac{1}{2}$$

$$X_2 = \frac{u_{s20}t}{(\rho_{20}/\rho_{10})\, l_{m2}}$$

$$T_2 = \frac{l_2}{l_{m2}}$$

$$\ln\left\{1 - \left[\frac{u_{s20}\left(t_{G2} - \frac{1}{u_{s2}}\frac{L_1}{L_2}\right) - 1}{l_{m2}}\right]^{1/2}\right\} + \left[\frac{u_{s20}\left(t_{G2} - \frac{1}{u_{s2}}\frac{L_1}{L_2}\right) - 1}{l_{m2}}\right]^{1/2}$$

$$+ \frac{u_{s20}\left(t_{G2} - \frac{1}{u_{s2}}\frac{L_1}{L_2}\right)\rho_{10}}{2\ l_{m2}\ \rho_{20}} = 0$$

The limiting separation approximation used in the acceleration tube is given by,

$$t_{G2} = \frac{l_{m2} + 1}{u_{s20}} + \frac{1}{u_{s2}}\frac{L_1}{L_2}$$

Turbulent

Reference: Mirels (1964)

The shock tube flow is turbulent. The Mirels effect also effects the blob trajectory. The limiting separation is not reached in ths shock tube length.

$$\frac{l_m}{L_2} = \left(\frac{1}{4\beta_1}\right)^{5/4} \frac{p_1}{p_2} \frac{\rho_2/\rho_1}{\rho_2/\rho_1 - 1} M_s^{1/4} \left(\frac{\rho_1 a_1}{\mu_1}\right)^{1/4} \left(\frac{d}{L_2}\right)^{5/4} \left(\frac{p_1}{p_4} \frac{p_4}{p_{st}}\right)^{1/4}$$

$$M_s = u_{s2} \sqrt{\frac{\gamma_4 R_4 T_4}{\gamma_1 R_1 T_1}}$$

$$\beta_1 = \beta_0 \left(\frac{(\rho_2/\rho_1)^2 + 1.25(\rho_2/\rho_1) - 0.80}{(\rho_2/\rho_1)((\rho_2/\rho_1) - 1)}\right)$$

M_s	β_0 (Air
	p_1 = 0.5cm Hg)
6	0.0283
8	0.0220
10	0.0188
12	0.0157
14	0.0129
16	0.0116
18	0.0104
20	0.0094

$$-\frac{X}{2} = \frac{5}{8}\left(\ln\frac{1 - T^n}{1 + T^n} - 2\arctan T^n + 4T^n\right), \quad n = \frac{1}{5}$$

$$X = \frac{u_{s2} t}{(\rho_2/\rho_1)\, l_m}$$

$$T = \frac{l}{l_m}$$

$$\frac{5}{8}\left(\ln\left\{\frac{1 - \left[\frac{x_G}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}}{1 + \left[\frac{x_G}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}}\right\} - 2\arctan\left[\frac{x_G}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}\right.$$

$$\left. + 4\left[\frac{x_G}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}\right) + \left[\frac{\rho_1}{\rho_2}\frac{u_{s2}}{2l_m}\left(\frac{x_G}{U_2 - A_2} + \frac{1}{u_{s2}}\frac{L_1}{L_2}\right)\right] = 0$$

$$t_G = \frac{x_G}{U_2 - A_2} + \frac{1}{u_{s2}}\frac{L_1}{L_2}$$

A.4 Blob Trajectories including Mirels Effect

$$X = \frac{u_{s1} t^*}{(\rho_2/\rho_1)\, l_m} \left(\frac{3R}{2 + R}\right)^{\frac{1}{1-n}}$$

$$T = \frac{l^*}{l_m} \left(\frac{3R}{2 + R}\right)^{\frac{1}{1-n}}$$

$$R = \frac{\rho}{\rho_{min}}$$

$$v = \left(\frac{3R}{2 + R}\right)^{\frac{5}{4}}$$

l^* = mixing front separation from shock wave

$$\frac{5}{8}\left(\ln \left\{ \frac{1 - \left[\frac{x_{GB}\, v}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}}{1 + \left[\frac{x_{GB}\, v}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5}} \right\} - 2 \arctan \left[\frac{x_{GB}\, v}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5} \right.$$

$$\left. + 4 \left[\frac{x_{GB}\, v}{l_m}\left(\frac{u_{s2}}{U_2 - A_2} - 1\right)\right]^{1/5} \right) + \left[\frac{\rho_1}{\rho_2} \frac{u_{s2}\, v}{2 l_m}\left(\frac{x_{GB}}{U_2 - A_2} + \frac{1}{u_{s2}} \frac{L_1}{L_2}\right)\right] = 0$$

$$t_{GB} = \frac{x_{GB}}{U_2 - A_2} + \frac{1}{u_{s2}} \frac{L_1}{L_2}$$

A.5 The Unsteady Method of Characteristics

Reference: Ferri (1961)

$$\frac{\delta p}{\delta t} + u \frac{\delta p}{\delta x} + \rho a^2 \frac{\delta u}{\delta x} = 0$$

$$\frac{\delta p}{\delta x} + \rho u \frac{\delta u}{\delta x} + \rho \frac{\delta u}{\delta t} = 0$$

$$\Delta S = 0$$

equation of state

$$p = \rho R T$$

definition of speed of sound for a perfect gas

$$a^2 = \left(\frac{\delta p}{\delta \rho}\right)_S = \gamma R T$$

physical characteristics

along first family, $\frac{dx}{dt} = u + a$

along second family, $\frac{dx}{dt} = u - a$

state characteristics

along first family, $\frac{dp}{dt} + \rho a \frac{du}{dt} = 0$

along second family, $\frac{dp}{dt} - \rho a \frac{du}{dt} = 0$

Interior Points

$$x_3 = \frac{t_1 - t_2 + \frac{x_2(u_2 - a_2 + u_3 - a_3)}{2(u_2 - a_2)(u_3 - a_3)} - \frac{x_1(u_1 + a_1 + u_3 + a_3)}{2(u_1 + a_1)(u_3 + a_3)}}{\frac{u_2 - a_2 + u_3 - a_3}{2(u_2 - a_2)(u_3 - a_3)} - \frac{u_1 + a_1 + u_3 + a_3}{2(u_1 + a_1)(u_3 + a_3)}}$$

$$t_3 = \frac{x_1 - x_2 + \frac{2t_2(u_2 - a_2)(u_3 - a_3)}{u_2 - a_2 + u_3 - a_3} - \frac{2t_1(u_1 + a_1)(u_3 + a_3)}{u_1 + a_1 + u_3 + a_3}}{\frac{2(u_2 - a_2)(u_3 - a_3)}{u_2 - a_2 + u_3 - a_3} - \frac{2(u_1 + a_1)(u_3 + a_3)}{u_1 + a_1 + u_3 + a_3}}$$

$$u_3 = \frac{u_1 + a_1}{2} + \frac{a_1 - a_2}{\gamma - 1}$$

$$a_3 = \frac{(\gamma - 1)(u_1 - u_2)}{4} + \frac{a_1 + a_2}{2}$$

For driver point:

$$p_3 = p_1 \left(\frac{a_3}{a_1}\right)^{\frac{2\gamma_4}{\gamma_4 - 1}}$$

$$T_3 = a_3{}^2$$

$$\rho_3 = \frac{p_3}{T_3}$$

For test gas point:

$$p_3 = p_1 \left(\frac{a_3}{a_1}\right)^{\frac{2\gamma_1}{\gamma_1 - 1}}$$

$$T_3 = a_3{}^2 \frac{\gamma_4 R_4}{\gamma_1 R_1}$$

$$\rho_3 = \frac{p_3 R_4}{T_3 R_1}$$

Expansion Wave Points

$$c_3 = \frac{-(\theta + \mu)}{100} \left(\arctan\left(\frac{1}{U_2 - A_2}\right) - \arctan\left(\frac{1}{U_5 - A_5}\right) \right) + \arctan\left(\frac{1}{u_1 - a_1}\right)$$

$$u_2 = \frac{2}{(\gamma_1 + 1)\tan c_3} + \frac{(\gamma_1 - 1)U_2}{\gamma_1 + 1}$$

$$a_2 = \frac{(\gamma_1 - 1)(U_2 - u_2)}{2} + A_2$$

$$p_2 = p_1 \left(\frac{a_2}{a_1}\right)^{\frac{2\gamma_1}{\gamma_1 - 1}}$$

$$\rho_2 = \rho_1 \left(\frac{a_2}{a_1}\right)^{\frac{2}{\gamma_1 - 1}}$$

$$x_2 = 0$$

$$t_2 = \frac{x_2 - x_1}{u_1 + a_1} + t_1$$

$$x_3 = \frac{t_1 - t_2 + \dfrac{x_2(u_2 - a_2 + u_3 - a_3)}{2(u_2 - a_2)(u_3 - a_3)} - \dfrac{x_1(u_1 + a_1 + u_3 + a_3)}{2(u_1 + a_1)(u_3 + a_3)}}{\dfrac{u_2 - a_2 + u_3 - a_3}{2(u_2 - a_2)(u_3 - a_3)} - \dfrac{u_1 + a_1 + u_3 + a_3}{2(u_1 + a_1)(u_3 + a_3)}}$$

$$t_3 = \frac{x_1 - x_2 + \dfrac{2t_2(u_2 - a_2)(u_3 - a_3)}{u_2 - a_2 + u_3 - a_3} - \dfrac{2t_1(u_1 + a_1)(u_3 + a_3)}{u_1 + a_1 + u_3 + a_3}}{\dfrac{2(u_2 - a_2)(u_3 - a_3)}{u_2 - a_2 + u_3 - a_3} - \dfrac{2(u_1 + a_1)(u_3 + a_3)}{u_1 + a_1 + u_3 + a_3}}$$

$$p_3 = p_1 \left(\frac{a_3}{a_1}\right)^{\frac{2\gamma_1}{\gamma_1 - 1}}$$

$$T_3 = a_3{}^2 \frac{\gamma_4 R_4}{\gamma_1 R_1}$$

$$\rho_3 = \frac{p_3 R_4}{T_3 R_1}$$

Contact Surface Points (velocities and pressures equal)

$$(u_3)_1 = u_4$$

$$(a_{3t})_1 = \frac{(\gamma_1 - 1)(u_4 - u_2)}{2} + a_2$$

$$(a_{3d})_1 = (a_{3t})_1$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$(u_1)_1 = \frac{(u_3)_1 + u_4}{2} + \frac{(a_{3d})_1 - a_4}{\gamma_4 - 1}$$

$$(a_1)_1 = \frac{(\gamma_4 - 1)((u_3)_1 - u_4)}{4} + \frac{(a_{3d})_1 + a_4}{2}$$

$$(x_3)_I = \frac{t_4 - t_2 + \dfrac{x_2(u_2 - a_2 + (u_3)_{I-1} - (a_{3t})_{I-1})}{2(u_2 - a_2)((u_3)_{I-1} - (a_{3t})_{I-1})} - \dfrac{x_4((u_3)_{I-1} + u_4)}{2(u_3)_{I-1}u_4}}{\dfrac{u_2 - a_2 + (u_3)_{I-1} - (a_{3t})_{I-1}}{2(u_2 - a_2)((u_3)_{I-1} - (a_{3t})_{I-1})} - \dfrac{(u_3)_{I-1} + u_4}{2(u_3)_{I-1}u_4}}$$

$$(t_3)_I = \frac{x_4 - x_2 + \dfrac{2t_2(u_2 - a_2)((u_3)_{I-1} - (a_{3t})_{I-1})}{u_2 - a_2 + (u_3)_{I-1} - (a_{3t})_{I-1}} - \dfrac{2t_4(u_3)_{I-1}u_4}{(u_3)_{I-1} + u_4}}{\dfrac{2(u_2 - a_2)((u_3)_{I-1} - (a_{3t})_{I-1})}{u_2 - a_2 + (u_3)_{I-1} - (a_{3t})_{I-1}} - \dfrac{2(u_3)_{I-1}u_4}{(u_3)_{I-1} + u_4}}$$

$$(x_1)_I = \left\{(t_3)_I - t_4 + \frac{x_4(u_4 - a_{4d} + u_5 - a_5)}{2(u_4 - a_{4d})(u_5 - a_5)}\right.$$

$$\left. - \frac{(x_3)_I((u_1)_{I-1} + (a_1)_{I-1} + (u_3)_I + (a_{3d})_{I-1})}{2((u_1)_{I-1} + (a_1)_{I-1})((u_3)_{I-1} + (a_{3d})_{I-1})}\right\}$$

$$+ \left\{\frac{u_4 - a_{4d} + u_5 - a_5}{2(u_4 - a_{4d})(u_5 - a_5)} - \frac{(u_1)_{I-1} + (a_1)_{I-1} + (u_3)_{I-1} + (a_{3d})_{I-1}}{2((u_1)_{I-1} + (a_1)_{I-1})((u_3)_{I-1} + (a_{3d})_{I-1})}\right\}$$

$$(t_1)_I = \left\{(x_3)_I - x_4 + \frac{2t_4(u_4 - a_{4d})(u_5 - a_5)}{u_4 - a_{4d} + u_5 - a_5}\right.$$

$$\left. - \frac{2(t_3)_I((u_1)_{I-1} + (a_1)_{I-1})((u_3)_{I-1} + (a_{3d})_{I-1})}{(u_1)_{I-1} + (a_1)_{I-1} + (u_3)_{I-1} . (a_{3d})_{I-1}}\right\}$$

$$+ \left\{\frac{2(u_4 - a_{4d})(u_5 - a_5)}{u_4 - a_{4d} + u_5 - a_5} - \frac{2((u_1)_{I-1} + (a_1)_{I-1})((u_3)_{I-1} + (a_{3d})_{I-1})}{(u_1)_{I-1} + (a_1)_{I-1} + (u_3)_{I-1} + (a_{3d})_{I-1}}\right\}$$

$$(u_1)_I = \frac{u_4\sqrt{(t_5 - (t_1)_I)^2 + (x_5 - (x_1)_I)^2} + u_5\sqrt{(t_4 - (t_1)_I)^2 + (x_4 - (x_1)_I)^2}}{\sqrt{(t_5 - t_4)^2 + (x_5 - x_4)^2}}$$

$$(a_1)_I = \frac{a_{4d}\sqrt{(t_5 - (t_1)_I)^2 + (x_5 - (x_1)_I)^2} + a_5\sqrt{(t_4 - (t_1)_I)^2 + (x_4 - (x_1)_I)^2}}{\sqrt{(t_5 - t_4)^2 + (x_5 - x_4)^2}}$$

$$(p_1)_I = \frac{p_4\sqrt{(t_5 - (t_1)_I)^2 + (x_5 - (x_1)_I)^2} + p_5\sqrt{(t_4 - (t_1)_I)^2 + (x_4 - (x_1)_I)^2}}{\sqrt{(t_5 - t_4)^2 + (x_5 - x_4)^2}}$$

$$(\rho_1)_I = \frac{\rho_{4d}\sqrt{(t_5 - (t_1)_I)^2 + (x_5 - (x_1)_I)^2} + \rho_5\sqrt{(t_4 - (t_1)_I)^2 + (x_4 - (x_1)_I)^2}}{\sqrt{(t_5 - t_4)^2 + (x_5 - x_4)^2}}$$

$$(p_3)_I = p_4\left(\frac{(a_{st})_{I-1}}{a_{4t}}\right)^{\frac{2\gamma_1}{\gamma_1 - 1}}$$

$$(\rho_{3d})_I = \frac{(p_3)_{I-1}}{{(a_{3d})_{I-1}}^2}$$

$$(\rho_{3t})_I = \frac{\gamma_1 (p_3)_{I-1}}{\gamma_4 (a_{3t})_{I-1}}$$

$$(u_3)_I = \frac{p_1 - p_2 + \frac{u_1(\rho_1 a_1 + (\rho_{3d})_I (a_{3d})_{I-1})}{2} + \frac{u_2(\rho_2 a_2 + (\rho_{3t})_I (a_{3t})_{I-1})}{2}}{\frac{(\rho_1)_I (a_1)_I + (\rho_{3d})_I (a_{3d})_{I-1}}{2} + \frac{\rho_2 a_2 + (\rho_{3t})_I (a_{3t})_{I-1}}{2}}$$

$$(p_3)_I = \frac{u_1 - u_2 + \frac{2p_1}{\rho_1 a_1 + (\rho_{3d})_I (a_{3d})_{I-1}} + \frac{2p_2}{\rho_2 a_2 + (\rho_{3t})_I (a_{3t})_{I-1}}}{\frac{2}{\rho_1 a_1 + (\rho_{3d})_I (a_{3d})_{I-1}} + \frac{2}{\rho_2 a_2 + (\rho_{3t})_I (a_{3t})_{I-1}}}$$

$$(a_{3t})_I = a_{4t} \left(\frac{(p_3)_I}{p_4}\right)^{\frac{\gamma_1 - 1}{2\gamma_1}}$$

$$(a_{3d})_I = a_{4d} \left(\frac{(p_3)_I}{p_4}\right)^{\frac{\gamma_4 - 1}{2\gamma_4}}$$

$$T_{3t} = \frac{\gamma_4 R_4 a_{3t}^2}{\gamma_1 R_1}$$

$$T_{3d} = a_{3d}^2$$

Blob Point

$$(x_2)_1 = x_6$$

$$(x_2)_2 = x_3$$

$$(t_2)_I = \frac{(t_6 - t_3)(x_2)_{I-1} - x_6}{x_6 - x_3} + t_6$$

$$(u_2)_I = \frac{u_6 \sqrt{(t_3 - t_2)^2 + (x_3 - (x_2)_{I-1})^2} + u_3 \sqrt{(t_6 - t_2)^2 + (x_6 - (x_2)_{I-1})^2}}{\sqrt{(t_6 - t_3)^2 + (x_6 - x_3)^2}}$$

$$(\rho_2)_I = \frac{\rho_6 \sqrt{(t_3 - t_2)^2 + (x_3 - (x_2)_{I-1})^2} + \rho_3 \sqrt{(t_6 - t_2)^2 + (x_6 - (x_2)_{I-1})^2}}{\sqrt{(t_6 - t_3)^2 + (x_6 - x_3)^2}}$$

$$(W_2)_I = \frac{2((u_2)_I - u_1)}{2R + 1} + W_1$$

$$(x_2)_{I-1} = \frac{2W_1 (W_2)_I ((t_2)_I - t_1)}{W_1 + (W_2)_I} + x_1$$

Blob Expansion Point

$$c_3 = \frac{-(\theta + \mu)}{100} \left(\arctan\left(\frac{1}{U_2 - A_2}\right) - \arctan\left(\frac{1}{U_5 - A_5}\right) \right) + \arctan\left(\frac{1}{u_1 - a_1}\right)$$

$$u_3 = \frac{2}{(\gamma_1 + 1)\tan c_3} + \frac{(\gamma_1 - 1)U_2}{\gamma_1 + 1}$$

$$a_3 = \frac{(\gamma_1 - 1)(U_2 - u_3)}{2} + A_2$$

$$T_3 = a_3{}^2 \frac{\gamma_4 R_4}{\gamma_1 R_1}$$

$$\rho_3 = \rho_1 \left(\frac{T_3}{T_1}\right)^{\frac{1}{\gamma_1 - 1}}$$

$$W_3 = \frac{3(u_3 - u_1)}{\dfrac{4R}{\rho_1 + \rho_3} + 1} + W_1$$

$$x_3 = \frac{\dfrac{1}{u_{s2}} \dfrac{L_1}{L_2} - t_1 + \dfrac{x_1 (W_1 + W_3)}{2 W_1 W_3}}{\dfrac{W_1 + W_3}{2 W_1 W_3} - \dfrac{1}{u_3 - a_3}}$$

$$t_3 = \frac{x_3}{u_3 - a_3} \frac{1}{u_{s2}} \frac{L_1}{L_2}$$

Boundary Conditions

1. Moving piston - gas remains in contact with piston.
2. Supersonic outflow through open-ended duct - both families of characteristics travel in same direction and both exit, same as interior points.
3. Strong shock waves - patch solutions together.

A.6 The Pitot Pressure

$$\frac{p_1}{p_{02}} = \frac{\left(\dfrac{2\gamma}{\gamma + 1} M_1{}^2 - \dfrac{\gamma - 1}{\gamma + 1}\right)^{\frac{1}{\gamma - 1}}}{\left(\dfrac{\gamma + 1}{2} M_1{}^2\right)^{\frac{\gamma}{\gamma - 1}}}$$

B Program Listing

```
' The Unsteady Method of Characteristics
' by Chris M  Gourley
' University of Queensland
' St  Lucia 4067
' Queensland, Australia
' September, 1987
' Microsoft BASIC version 2.10.00 (Binary Math)
'
'________________________________________________
' INITIALISE PROGRAM AND SET UP EVENT TRAPPING

        WINDOW CLOSE 1
        CLEAR
    initialise
        DEFINT i-n : DEFDBL a-h : DEFDBL o-z
        MenuSelect = 0 : MenuItem = 0 : Activity = 0
        ButtonPressed = 0 : EFSelected = 0 : WSelected = 0 : WGoAway = 0
        WDirty = 0 : TRUE% = 1 : FALSE% = 0
        i = 0
        k = 1
        isize = 500 : jsize = 500 : max = 5 : iScrollH= 0 : iScrollT = 0 : iopenflag = 0 : inumberflag = 0 : icircleflag = 1 : iexpend = 0 : itestflag
            = 0 : ilangleyflag = 0
        infoflag = 0 : igridflag = 0 : imeshflag = 0 : iDirty = 1 : mouseflag = 0 : iwindowflag = 0 : igammaflag = 0 : ipitotpoint = 1 : ipitotflag =
            0
        pi = 4#*ATN(1#)
        tol = .0000001#
        P4P1 = 0# : p1p10 = 0# : T4T1 = 0# : L1L2# = 0# : magfactor# = 1! : rhom = 0#
        jjflag = 0 : firstinfo = 0
        DIM iwatch(33), icrosshair(33), igrey(3), icircle(33), igrey2(3), ibar(3), ihand(33), idash(3), p$(10), penLocation%(1)
        DIM PointLocx(500), PointLocy(500), PointType(500), iPoint(500), jPoint(500)
        RESTORE
        FOR ii = 0 TO 31
                READ iwatch(ii)
        NEXT ii
        SETCURSOR VARPTR (iwatch(0))
        RESTORE 1
        FOR ii = 0 TO 15
                READ icrosshair(ii)
        NEXT ii
        icrosshair(32) = 7 : icrosshair(33) = 8
        FOR ii = 0 TO 31
                READ icircle(ii)
        NEXT ii
        icircle(32) = 7 : icircle(33) = 8
        FOR ii = 0 TO 3
                igrey(ii) = -21931
        NEXT ii
    ' draw dashed wall pattern
        FOR ii = 0 TO 3
                idash(ii) = 16191
        NEXT ii
        FOR ii = 0 TO 31
                READ ihand(ii)
        NEXT ii
        ihand(32) = 7 : ihand(33) = 8
    ' set up info scroll bar
        PICTURE ON
    ' draw grey pattern
        FOR ii = 0 TO 3
                igrey2(ii) = -30806
        NEXT ii
        ibar(0) = 14 : ibar(1) = 94 : ibar(2) = 66 : ibar(3) = 110
    ' frame grey pattern
        FILLRECT (VARPTR(ibar(0))), (VARPTR(igrey2(0)))
        FRAMERECT (VARPTR(ibar(0)))
        LINE (94, 0) - (94, 80)
    ' draw up arrow
        MOVETO 102, 2
        CALL LINE (-6, 6) : CALL LINE (3, 0) : CALL LINE (0, 4) : CALL LINE (6, 0) : CALL LINE (0, -4) : CALL LINE (3, 0) : CALL LINE (-6, -6)
    ' draw down arrow
        MOVETO 99, 68
        CALL LINE (0, 4) : CALL LINE (-3, 0) : CALL LINE (6, 6) : CALL LINE (6, -6) : CALL LINE (-3 , 0) : CALL LINE (0, -4) : CALL LINE (-6, 0)
        PICTURE OFF
        ScrollBar$ = PICTURE$
        PICTURE ON
    ' draw scroll box
        LINE (95, 15) - (109, 30), 30, bf : LINE (95, 15) - (109, 30), , b
        PICTURE OFF
        ScrollBox$ = PICTURE$
    ' watch data
        DATA 2016, 2016, 2016, 2016, 2064, 4232, 4232, 4236
        DATA 5004, 4104, 4104, 2064, 2016, 2016, 2016, 2016
        DATA 2016, 2016, 2016, 2016, 4080, 8184, 8184, 8184
        DATA 8184, 8184, 8184, 4080, 2016, 2016, 2016, 2016
    ' crosshair data
1       DATA 0, 0, 256, 256, 256, 256, 256, 16376
        DATA 256, 256, 256, 256, 256, 0, 0, 0
    ' circle data
        DATA 0, 0, 0, 0, 896, 1088, 2080, 2080
        DATA 2080, 1088, 896, 0, 0, 0, 0, 0
        DATA 0, 0, 0, 0, 896, 1088, 2080, 2080
        DATA 2080, 1088, 896, 0, 0, 0, 0, 0
    ' hand data
        DATA 0, 0, 0, 0, 896, 3200, 4352, 8960
        DATA 25598, 17921, 19454, 29192, 17392, 16912, 25568, 16320
        DATA 0, 0, 0, 0, 896, 3968, 7936, 16128
        DATA 32766, 32767, 32766, 32760, 32752, 32752, 32736, 16320
2       DATA "UNSTEADY CHARACTERISTICS PROGRAM", "by Chris Gourley"
        DATA "Dept. of Mechanical Engineering", "University of Queensland, St. Lucia"
        DATA "Queensland 4067, Australia", "Does database file exist ?"
        DATA "Click OK to begin "
3       DATA "EXPANSION TUBE INITIAL CONDITIONS", "Driver Tube (4)", "Shock Tube (1)", "Accel Tube (10)", "Driver Gas =", "Test/Accel Gas ="
5       DATA "FLOWFIELD SETUP BOX", "Meshscale", "Magnification"
6       DATA "New Magnification"
200     DATA "PRINT SETUP BOX", "x increment", "y increment", "Magnification", "%"
241     DATA "Start", "Finish"
300     DATA "PITOT PLOT SETUP"
3000    DATA "Scroll %"
        MENU 1, 0, 1, "File"
            MENU 1, 1, 1, "Open..."
            MENU 1, 2, 0, "Close"
            MENU 1, 3, 0, "Save"
            MENU 1, 4, 0, "Print..."
            MENU 1, 5, 1, "Quit"
        MENU 3, 0, 0, "Display"
            MENU 3, 1, 2, "Mesh..."
            MENU 3, 2, 0, "Table..."
            MENU 3, 3, 0, "-"
            MENU 3, 4, 1, "Grid"
            MENU 3, 5, 0, "-"
            MENU 3, 6, 1, "Point Numbers"
            MENU 3, 7, 0, "-"
            MENU 3, 8, 2, "Point Circles"
            MENU 3, 9, 0, "-"
            MENU 3, 10, 1, "Pitot Plot..."
        MENU 4, 0, 0, "Point"
            MENU 4, 1, 0, "Contour..."
            MENU 4, 2, 0, "Expansion..."
            MENU 4, 3, 0, "Driver..."
            MENU 4, 4, 0, "Test..."
            MENU 4, 5, 0, "Blob..."
            MENU 4, 6, 0, "BlobExp..."
            MENU 4, 7, 0, "-"
            MENU 4, 8, 1, "Mesh Split"
            MENU 4, 9  0, "-"
            MENU 4, 10, 0, "Erase..."
        MENU 5, 0, 0, "Info"
            MENU 5, 1, 1, "Display Info"
            MENU 5, 2, 0, "Get Info..."
        MENU 6, 0, 0, "Scale"
            MENU 6, 1, 1, "Zoom..."
            MENU 6, 2, 1, "Scroll..."
        MENU 7, 0, 0, "Copy"
            MENU 7, 1, 1, "CopyFlow"
    Activate event handling
        ON DIALOG GOSUB DialogEvent
        ON MENU GOSUB MenuEvent
        ON BREAK GOSUB BreakEvent
        ON MOUSE GOSUB MouseEvent
        MENU        'turn off any highlighted menus
        INITCURSOR
        MENU ON : DIALOG ON

'-------------------------------------------
'     MAIN PROGRAM LOOP

Idle:
        WHILE TRUE% = 1                 'Endless loop
        WEND

'-------------------------------------------
' HANDLE EVENTS HERE

MenuEvent:
        DIALOG STOP : IF infoflag = 1 THEN MOUSE STOP
        SETCURSOR VARPTR (iwatch(0))
        MenuSelect = MENU(0)
        ON MenuSelect GOSUB FileMenu, , DisplayMenu, PointMenu, InfoMenu, ScaleMenu, CopyMenu
        MENU
        IF infoflag = 1 THEN MOUSE ON
        DIALOG ON : INITCURSOR
        RETURN

DialogEvent:
        MENU STOP : IF infoflag = 1 THEN MOUSE STOP
        Activity = DIALOG(0)
        ON Activity GOSUB ButtonEvent, EditEvent, Activate, GoAway, Refresh, ReturnEvent, TabEvent
        Activity = DIALOG(0) : IF infoflag = 1 THEN MOUSE ON
        MENU ON
        RETURN

BreakEvent:
        ibreak = 1 : iDirty = 0
        IF infoflag = 1 THEN GOSUB FindInfo2
        RETURN

MouseEvent:
        IF mouseflag = 1 THEN RETURN
        MENU STOP : DIALOG STOP
        MouseClick = MOUSE(0) + 4
        ON MouseClick GOSUB MouseReturn, MouseReturn, MousePosition, MouseReturn, MousePosition, MouseReturn, MouseReturn
        IF MOUSE(0) = -1 THEN GOTO MouseEvent
        MENU ON : DIALOG ON
        RETURN

'-------------------------------------------
' HANDLE THE MENU EVENTS HERE

FileMenu:
        MenuItem = MENU(1)
        ON MenuItem GOSUB OpenFile, CloseFile, SaveFile, PrintFile, Quit
        RETURN

OpenFile:
' Setup database interface for storage of information about a particular
' geometry and solution.
        GOSUB DrawDialog1
        MENU 1, 1, 0 : MENU 1, 2, 1 : MENU 1, 3, 1 : MENU 1, 4, 1
        RETURN

CloseFile:
        WINDOW CLOSE 3 : WINDOW CLOSE 2 : WINDOW CLOSE 4
        MENU 3, 1, 2 : MENU 3, 2, 0
        MENU 3, 8, 2 : MENU 3, 10, 1
        MENU 4, 1, 1 : MENU 4, 2, 1 : MENU 4, 3, 1 : MENU 4, 4, 1 : MENU 4, 5, 1 : MENU 4, 6, 1 : MENU 4, 8, 1 : MENU 4, 10, 1
        MENU 6, 1, 1 : MENU 6, 2, 1 : MENU 7, 1, 1
    ' Save status information
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 2 AS max$, 8 AS magfactor$, 2 AS ScrollX$, 2 AS ScrollY$, 8 AS xg$, 8 AS tg$, 8 AS xgblob$, 8 AS tgblob$, 8 AS tg2$, 2 AS iexpend$,
8 AS gammablob$
        LSET max$ = MKI$(max)
        LSET magfactor$ = MKD$(magfactor#)
        LSET ScrollX$ = MKI$(iScrollX)
        LSET ScrollY$ = MKI$(iScrollY)
        LSET xg$ = MKD$(xg)
        LSET tg$ = MKD$(tg)
        LSET xgblob$ = MKD$(xgblob)
        LSET tgblob$ = MKD$(tgblob)
        LSET tg2$ = MKD$(tg2)
        LSET iexpend$ = MKI$(iexpend)
        LSET gammablob$ = MKD$(gammablob)
        PUT #1, 1
        RESET
        MENU 1, 1, 1 : MENU 1, 2, 0 : MENU 1, 3, 0 : MENU 1, 4, 0 : MENU 7, 0, 0 : MENU 4, 1, 0 : MENU 4, 2, 0 : MENU 4, 3, 0 : MENU 4, 8, 1 : MENU 4
10, 0 : MENU 4, 5, 0
        MENU 5, 1, 1 : MENU 5, 2, 0 : MENU 3, 4, 1
        MENU 3, 1, 2 : MENU 3, 2, 0 : MENU 3, 8, 2
        MENU 3, 0, 0 : MENU 4, 0, 0 : MENU 5, 0, 0 : MENU 6, 0, 0 : MENU 4, 4, 0
        i = 0
        k = 1
        max = 5 : iScrollX = 0 : iScrollY = 0 : iopenflag = 0 : inumberflag = 0 : icircleflag = 1 : iexpend = 0 : ilangleyflag = 0 : iheatflag = 0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        infoflag = 0 : igridflag = 0 : imeshflag = 0 : iDirty = 1 : mouseflag = 0 : iwindowflag = 0 : igammaflag = 0 : ipitotflag = 0 : ipitotpoint =
            :
        P4P1 = 0# : p1p10 = 0# : T4T1 = 0# : L1L24 = 0# : magfactor# = 1! : rhom = 0#
        jjflag = 0 : firstinfo = 0
        RETURN

SaveFile:
      ' Save status information
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 2 AS max$, 8 AS magfactor$, 2 AS ScrollX$, 2 AS ScrollY$, 8 AS ag$, 8 AS tg$, 8 AS agblob$, 8 AS tgblob$, 8 AS tg2$, 2 AS iexpend$
           8 AS gammablob$
        LSET max$ = MKI$(max)
        LSET magfactor$ = MKD$(magfactor#)
        LSET ScrollX$ = MKI$(iScrollX)
        LSET ScrollY$ = MKI$(iScrollY)
        LSET ag$ = MKD$(ag)
        LSET tg$ = MKD$(tg)
        LSET agblob$ = MKD$(agblob)
        LSET tgblob$ = MKD$(tgblob)
        LSET tg2$ = MKD$(tg2)
        LSET iexpend$ = MKI$(iexpend)
        LSET gammablob$ = MKD$(gammablob)
        PUT #1, 1
      ' Set file up for point data
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 16 AS gg$, 2 AS i$, 2 AS j$, 8 AS x$, 8 AS t$, 8 AS u$, 8 AS a$, 8 AS p$, 8 AS TT$, 8 AS rho$
        RETURN

PrintFile:
        IF infoflag = 1 THEN WINDOW CLOSE 2 : WINDOW 4
        CALL HIDEPEN : LINE (0, 0) - (507, 317), , b
        PICTURE OFF : image$ = PICTURE$ : PICTURE ON : CALL SHOWPEN
        ' Draw box to input scaling information
        ixx = 50 : iYY = 10 : ss = 1
        WINDOW 4, "", (116, 80) - (396, 150), -2
        RESTORE 200 : TEXTFACE(1)
        READ as$ : MOVETO (224# - LEN(as$)*8.7)/2#, 20 : PRINT as$
        BUTTON 1, 1, "OK", (224, 6) - (274, 28), 1
        EDIT FIELD 1, STR$(ixx), (5, 40) - (45, 55), , 2
        READ as$ : MOVETO 50, 52 : PRINT as$
        EDIT FIELD 2, STR$(iYY), (5, 60) - (45, 75), , 2
        READ as$ : MOVETO 50, 72 : PRINT as$
        READ as$ : MOVETO 140# + (140# - LEN(as$)*7.4)/2#, 52 : PRINT as$ : EDIT FIELD 3, STR$(ss*100), (190, 58) - (230, 73), , 2
        READ as$ : MOVETO 240, 72 : PRINT as$
        LINE (5, 34) - (275, 34) : LINE (140, 34) - (140, 104)
        EDIT FIELD 1
        i = DIALOG(0) : INITCURSOR : tabfield = 1
        WHILE i <> 1 AND i <> 6
                IF i = 7 THEN IF tabfield < 3 THEN tabfield = tabfield + 1 ELSE tabfield = 1
                IF i = 7 THEN EDIT FIELD tabfield
                i = DIALOG(0)
        WEND
        ixx = VAL(EDIT$(1))
        iYY = VAL(EDIT$(2))
        ss = VAL(EDIT$(3))/100
        WINDOW CLOSE 4 : ibreak = 0
        ON ERROR GOTO ErrorFix
100     IF ibreak = 0 THEN OPEN "LPT1:PROMPT" FOR OUTPUT AS 2 : SETCURSOR VARPTR (iwatch(0)) : WINDOW OUTPUT #2 : PICTURE (0 + ixx, 0 + iYY) - ((508 +
            ixx)*ss, (318 + iYY)*ss), image$
    PrintEnd:
        ON ERROR GOTO 0
        CLOSE #2
        IF infoflag = 1 THEN WINDOW 2 : GOSUB FindInfo2
        RETURN

Quit:
        IF iopenflag = 0 THEN RESET : MENU RESET : SYSTEM
      ' Save status information
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 2 AS max$, 8 AS magfactor$, 2 AS ScrollX$, 2 AS ScrollY$, 8 AS ag$, 8 AS tg$, 8 AS agblob$, 8 AS tgblob$, 8 AS tg2$, 2 AS iexpend$
           8 AS gammablob$
        LSET max$ = MKI$(max)
        LSET magfactor$ = MKD$(magfactor#)
        LSET ScrollX$ = MKI$(iScrollX)
        LSET ScrollY$ = MKI$(iScrollY)
        LSET ag$ = MKD$(ag)
        LSET tg$ = MKD$(tg)
        LSET agblob$ = MKD$(agblob)
        LSET tgblob$ = MKD$(tgblob)
        LSET tg2$ = MKD$(tg2)
        LSET iexpend$ = MKI$(iexpend)
        LSET gammablob$ = MKD$(gammablob)
        PUT #1, 1
        RESET
        MENU RESET
        SYSTEM

DisplayMenu:
        MenuItem = MENU(1)
        ON MenuItem GOSUB MeshDisplay, TableDisplay, , GridDisplay, , NumberDisplay, , CircleDisplay, , PitotDisplay
        RETURN

MeshDisplay:
        MENU 7, 0, 1 : MENU 3, 1, 2 : MENU 3, 2, 0 : MENU 4, 0, 1 : MENU 6, 0, 1
        IF igridflag = 1 THEN MENU 3, 4, 2 ELSE MENU 3, 4, 1
        IF inumberflag = 1 THEN MENU 3, 6, 2 ELSE MENU 3, 6, 1
        IF icircleflag = 1 THEN MENU 3, 8, 2 ELSE MENU 3, 8, 1
        CLS : GOSUB RePlot
        IF infoflag = 1 THEN iDirty = 0 : ii = iiii : WINDOW 2, "", (400, 260) - (510, 340), 3 : GOSUB FindInfo
        RETURN

TableDisplay:
        PICTURE OFF
        MENU 7, 0, 0 : MENU 3, 1, 1 : MENU 3, 2, 2 : MENU 4, 0, 0 : MENU 6, 0, 0
        MENU 3, 4, 0 : MENU 3, 6, 0 : MENU 3, 8, 0 : CLS : iiii = ii : GOSUB DrawTable
        RETURN

GridDisplay:
        IF igridflag = 0 THEN igridflag = 1 : MENU 3, 4, 2 : GOSUB Grid : RETURN
        IF igridflag = 1 THEN igridflag = 0 : MENU 3, 4, 1 : CLS : GOSUB RePlot2 : RETURN

NumberDisplay:
        IF inumberflag = 0 THEN inumberflag = 1 : MENU 3, 6, 2 : GOSUB ShowNumber : RETURN
        IF inumberflag = 1 THEN inumberflag = 0 : MENU 3, 6, 1 : CLS : GOSUB RePlot2 : RETURN

CircleDisplay:
        IF icircleflag = 0 THEN icircleflag = 1 : MENU 3, 8, 2 : MENU 4, 0, 1 : MENU 5, 0, 1 : GOSUB RePlot2 : RETURN
        IF icircleflag = 1 THEN icircleflag = 0 : MENU 3, 8, 1 : MENU 4, 0, 0 : MENU 5, 0, 0 : CLS : GOSUB RePlot
        RETURN

PitotDisplay:
        MENU 3, 1, 0 : MENU 3, 2, 0 : MENU 3, 4, 0 : MENU 3, 6, 0 : MENU 3, 8, 0 : MENU 3, 10, 0
        MENU 4, 1, 0 : MENU 4, 2, 0 : MENU 4, 3, 0 : MENU 4, 4, 0 : MENU 4, 5, 0 : MENU 4, 6, 0 : MENU 4, 8, 0 : MENU 4, 10, 0
        MENU 5, 1, 0 : MENU 5, 2, 0 : MENU 6, 1, 0 : MENU 6, 2, 0 : MENU 7, 1, 0
        IF infoflag = 1 THEN MOUSE OFF
        mouseflag = 1 : ipitotpoint = 1 : GOSUB DrawWindow4 : ipitotflag = 1 : WINDOW CLOSE 2
        RETURN

PointMenu:
        MenuItem = MENU(1)
        ON MenuItem GOSUB ContactPoint, ExpPoint, InteriorPointd, InteriorPointt, Blob, BlobExpansion, MeshSplit, , ErasePoint
        RETURN

ContactPoint:
        inewflagc = 1
        IF infoflag = 1 THEN MOUSE OFF : WINDOW 3
        mouseflag = 1 : istopflag = 0
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectContact : IF ibreak = 1 THEN GOTO PointCEnd
        SETCURSOR VARPTR (iwatch(0))
        j = 112
        GOSUB FindPoint2 : GOSUB FindPoint5 : GOSUB FindPoint4t : GOSUB FindPoint4d
    ' Initial guess at properties
        u3 = u4
        a3t = (gamma1 - 1#)*(u4 - u2)/2# + a2
        a3d = a3t
        u1 = (u3 + u4)/2! + (a3d - a4)/(gamma4 - 1!)
        a1 = (gamma4 - 1!)*(u3 - u4)/4! + (a3d + a4)/2!
    ' Iterate for position and properties of contact surface point
    ContactLoop:
        u30 = u3
        a3t0 = a3t
        a3d0 = a3d
        GOSUB Contact
        IF ABS(u30 - u3)/u30 < tol AND ABS(a3t0 - a3t)/a3t0 < tol AND ABS(a3d0 - a3d)/a3d0 < tol THEN GOTO ContactCalcs ELSE GOTO ContactLoop
    ContactCalcs:
    ' calculate contact point remaining properties
        TT3t = (a3t^2)*(gamma4*R4)/(gamma1*R1)
        TT3d = a3d^2
    ' save point properties
        IF inewflagc = 0 THEN max = max - 1
        IF t1 > t5 THEN BEEP : BEEP : GOTO PointCEnd
        a3 = a3t : TT3 = TT3t : rho3 = rho3t : i = max + 2 : j = 112 : Grid$ = "Contactt"
        GOSUB SavePoint : GOSUB DrawPoint
        PointLocx(max) = x3 : PointLoct(max) = t3 : PointType(max) = 2 : iPoint(max) = i : jPoint(max) = j
        a3 = a3d : TT3 = TT3d : rho3 = rho3d : i = max + 2 : j = max : Grid$ = "Contactd"
        GOSUB SavePoint
        PointLocx(max) = x3 : PointLoct(max) = t3 : PointType(max) =11 : iPoint(max) = i : jPoint(max) = j
        GOSUB DrawMach
        GOSUB DrawContact
        GOSUB Rename3
        IF inewflagc <> 0 THEN GET #1, 115 : jj = max + 1 : LSET j$ = MKI$(jj) : PUT #1, 115 : jPoint(115) = max + 1
        Grid$ = "Interiorj"
        GOSUB SavePoint
        GOSUB DrawPoint
        PointLocx(max) = x3 : PointLoct(max) = t3 : PointType(max) = 4 : iPoint(max) = i : jPoint(max) = j
    PointCEnd:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR : inewflagc = 0
        RETURN

ExpPoint:
        IF infoflag = 1 THEN MOUSE OFF : WINDOW 3
        mouseflag = 1 : istopflag = 0 : iexpand2 = 0 : igammaflag = 0
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectExpansion : IF ibreak = 1 THEN GOTO PointEEnd
        SETCURSOR VARPTR (iwatch(0))
        i = 11 : GOSUB FindPoint1
        IF iexpand = 0 THEN expangle = -(ATN(1#/(uu2 - aa2)) - ATN(1#/(uu5 - aa5)))*expangle/100# + ATN(1#/(u1 - a1))
        IF iexpand = 0 THEN IF expangle < ATN(1#/(uu5 - aa5)) THEN expangle = ATN(1#/(uu5 - aa5)) : BEEP : iexpand2 = 1
        IF iexpand = 0 THEN Grid$ = "Expansion" : GOSUB Expansion : GOSUB Interior ELSE Grid$ = "Expansion" : u2 = uu5 : a2 = aa5 : GOSUB Interior
        IF iexpand = 0 THEN x2 = 0# : t2 = L1L2#/us2 ELSE x2 = x1 + expangle/100# : t2 = (x2 - x1)/(u1 + a1) + t1
        x3 = (t1 - t2 + x2*(u2 - a2 + u3 - a3)/(2#*(u2 - a2)*(u3 - a3)) - x1*(u1 + a1 + u3 + a3)/(2#*(u1 + a1)*(u3 + a3)))/((u2 - a2 + u3 -
            a3)/(2#*(u2 - a2)*(u3 - a3)) - (u1 + a1 + u3 + a3)/(2#*(u1 + a1)*(u3 + a3)))
        t3 = (x1 - x2 + t2*2#*(u2 - a2)*(u3 - a3)/(u2 - a2 + u3 - a3) - t1*2#*(u1 + a1)*(u3 + a3)/(u1 + a1 + u3 + a3))/(2#*(u2 - a2)*(u3 - a3)/(u2 -
            a2 + u3 - a3) - 2#*(u1 + a1)*(u3 + a3)/(u1 + a1 + u3 + a3))
        PointLocx(max + 1) = x3 : PointLoct(max + 1) = t3 : PointType(max + 1) = 1 : iPoint(max + 1) = i : IF iexpand = 0 THEN jPoint(max + 1) = -1
            ELSE jPoint(max + 1) = -2
        IF iexpand = 0 THEN j = -1 ELSE j = -2
        GOSUB Calculations
        IF iexpand2 = 1 THEN iexpand = 1
    PointEEnd:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR
        RETURN

InteriorPointd:
        igammaflag = 1 : GOTO NewInterior
InteriorPointt:
        igammaflag = 0
    NewInterior:
        IF infoflag = 1 THEN MOUSE OFF : WINDOW 3
        mouseflag = 1 : istopflag = 0
        IF imeshflag = 1 THEN GOTO Interior2
        inewflagi = 1
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectInterior : IF ibreak = 1 THEN PointIIEnd
        SETCURSOR VARPTR (iwatch(0))
        i = 111 : j = 112
        GOSUB FindPoint1 : GOSUB FindPoint2
        Grid$ = "Interior" : GOSUB Interior
        x3 = (t1 - t2 + x2*(u2 - a2 + u3 - a3)/(2#*(u2 - a2)*(u3 - a3)) - x1*(u1 + a1 + u3 + a3)/(2#*(u1 + a1)*(u3 + a3)))/((u2 - a2 + u3 -
            a3)/(2#*(u2 - a2)*(u3 - a3)) - (u1 + a1 + u3 + a3)/(2#*(u1 + a1)*(u3 + a3)))
        t3 = (x1 - x2 + t2*2#*(u2 - a2)*(u3 - a3)/(u2 - a2 + u3 - a3) - t1*2#*(u1 + a1)*(u3 + a3)/(u1 + a1 + u3 + a3))/(2#*(u2 - a2)*(u3 - a3)/(u2 -
            a2 + u3 - a3) - 2#*(u1 + a1)*(u3 + a3)/(u1 + a1 + u3 + a3))
        GOSUB Calculations
        PointLocx(max) = x3 : PointLoct(max) = t3 : PointType(max) = 0 : iPoint(max) = i : jPoint(max) = j
    PointIIEnd:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR : inewflagi = 0
        RETURN
    Interior2:
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectInterior : IF ibreak = 1 THEN GOTO PointI2End
        i = 111 : GOSUB FindPoint1
        IF PointType(i) = 2 OR PointType(i) = 10 OR PointType(i) = 11 THEN BEEP : BEEP : GOTO PointI2End
        sx1 = 490#*(x1*magfactor# + L1L2#)/(1# + L1L2#) + 14# + iScrollX
        st1 = 314# + iScrollT - 490#*t1*magfactor#*tstretch/(1# + L1L2#)
        j = 112 : GOSUB FindPoint2
        IF PointType(j) = 2 OR PointType(j) = 10 OR PointType(j) = 11 THEN BEEP : BEEP : GOTO PointI2End
        sx2 = 490#*(x2*magfactor# + L1L2#)/(1# + L1L2#) + 14# + iScrollX
        st2 = 314# + iScrollT - 490#*t2*magfactor#*tstretch/(1# + L1L2#)
        itimer = 0 : SETCURSOR VARPTR (icircle(0)) : MENU OFF : GOSUB SelectSplit : LINE (sx1, st1) - (sx2, st2), 33 : SETCURSOR VARPTR (iwatch(0)) :
          MENU ON : IF ibreak = 1 THEN GOTO MeshSplit
        GOSUB Average : x3 = xxx6 : t3 = ttt6 : GOSUB Renumber : GOSUB DrawPoint
        PointLocx(max) = x3 : PointLoct(max) = t3 : iPoint(max) = i : jPoint(max) = j
    Point13End:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR
        GOTO MeshSplit

Blob:
        IF infoflag = 1 THEN MOUSE OFF : WINDOW 3
        mouseflag = 1 : istopflag = 0
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectBlob : IF ibreak = 1 THEN GOTO BlobFinish
        SETCURSOR VARPTR(iwatch(0)) : i = ii
        GOSUB FindPoint1 : GOSUB FindPoint3 : GOSUB FindPoint6
            inoblob = 0
        GET #1, ii : IF gg$ = "Contact          " THEN GOSUB MakeBlob ELSE rhom = p1 : W1 = a1 : inoblob = 1
        IF inoblob = 0 THEN GOTO BlobFinish
            xlower1 = x6 : xupper1 = x3 : iflag = 2
        t2 = (t6 - t3)*(xlower1 - x6)/(x6 - x3) + t6
        u2 = (SQR((t3 - t2)^2 + (x3 - xlower1)^2)*u6 + SQR((t6 - t2)^2 + (x6 - xlower1)^2)*u3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        rho2 = (SQR((t3 - t2)^2 + (x3 - xlower1)^2)*rho6 + SQR((t6 - t2)^2 + (x6 - xlower1)^2)*rho3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        IF rhom > 1# THEN W2 = (u2 + u1)/2# ELSE W2 = (u2 - u1)*(3#/(2#*rhom + 1#)) + W1
        x2lower = (t2 - t1)*2#*W1*W2/(W1 + W2) + x1 - xlower1
        t2 = (t6 - t3)*(xupper1 - x6)/(x6 - x3) + t6
        u2 = (SQR((t3 - t2)^2 + (x3 - xupper1)^2)*u6 + SQR((t6 - t2)^2 + (x6 - xupper1)^2)*u3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        rho2 = (SQR((t3 - t2)^2 + (x3 - xupper1)^2)*rho6 + SQR((t6 - t2)^2 + (x6 - xupper1)^2)*rho3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        rho2 = (SQR((t3 - t2)^2 + (x3 - xupper1)^2)*TT6 + SQR((t6 - t2)^2 + (x6 - xupper1)^2)*TT3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        IF rhom > 1# THEN W2 = (u2 + u1)/2# ELSE W2 = (u2 - u1)*(3#/(2#*rhom + 1#)) + W1
        x2upper = (t2 - t1)*2#*W1*W2/(W1 + W2) + x1 - xupper1
        IF x2lower*x2upper > 0# THEN BEEP : BEEP : BEEP : GOTO BlobFinish
        CALL zeroin(xlower1, xupper1, tol, iflag, x) : x2 = x
        t2 = (t6 - t3)*(x2 - x6)/(x6 - x3) + t6
        u2 = (SQR((t3 - t2)^2 + (x3 - x2)^2)*u6 + SQR((t6 - t2)^2 + (x6 - x2)^2)*u3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        rho2 = (SQR((t3 - t2)^2 + (x3 - x2)^2)*rho6 + SQR((t6 - t2)^2 + (x6 - x2)^2)*rho3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        IF rhom > 1# THEN W2 = (u2 + u1)/2# ELSE W2 = (u2 - u1)*(3#/(2#*rhom + 1#)) + W1
        p1 = rho1*R1*TT1/R4
        aaa = SQR((t3 - t2)^2 + (x3 - x2)^2)
        bbb = SQR((t2 - t6)^2 + (x2 - x6)^2)
        p2 = (aaa*p6 + bbb*p3)/(aaa + bbb)
        rhom = (rhom*rho1/rho2)*(p2/p1)^(1#/gammablob)
        GOSUB Rename4 : Grid$ = "Blob"
        GOSUB SavePoint
        PointLocx(max) = x2 : PointLoct(max) = t2
        PointType(max) = 6
        iPoint(max) = ii
        jPoint(max) = ii
        GOSUB DrawPoint
        GOSUB DrawBlob
    BlobFinish:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR
        RETURN

MakeBlob:
        BEEP : BEEP : BEEP : RETURN
    ' find density minimum
        cpblob = 7#
        ii = i1 : GOSUB FindPoint
        aalpha = 5#*R4 - cpblob*R1
        balpha = cpblob*R1
        calpha = R4 - R1
        dalpha = R1
        ealpha = 5#*R4*TT - cpblob*R1*TT1
        falpha = cpblob*R1*TT1
        alphamin1 = -balpha/aalpha
        alphamin2 = (balpha/aalpha)^2 - balpha*dalpha/(aalpha*calpha) - balpha*falpha/(aalpha*ealpha) + dalpha*falpha/(calpha*ealpha)
        IF alphamin2 < 0# THEN IF rho < rho1 THEN rhom = rho : GOTO CloseMake ELSE BEEP : BEEP : BEEP : RETURN ELSE REM
        alphamin = alphamin1 - SQR(alphamin2)
        IF alphamin < 0# OR alphamin > 1# THEN alphamin = alphamin1 + SQR(alphamin2)
        IF alphamin < 0# OR alphamin > 1# THEN IF rho < rho1 THEN rhom = rho : GOTO CloseMake ELSE BEEP : BEEP : BEEP : RETURN ELSE REM
        rhom = R4*p*((5#*R4 - cpblob*R1)*alphamin + cpblob*R1)/(((R4 - R1)*alphamin + R1)*((5#*R4*TT - cpblob*R1*TT1)*alphamin + cpblob*R1*TT1
        IF rhom > rho1 THEN IF rho > rho1 THEN BEEP : BEEP : BEEP : RETURN ELSE rhom = rho : GOTO CloseMake ELSE REM
        IF rhom < rho1 THEN IF rho < rhom THEN rhom = rho
    CloseMake:
        rhom = rhom/rho1
        W1 = u1
            inoblob = 1
        RETURN

BlobExpansion:
        IF infoflag = 1 THEN MOUSE OFF : WINDOW 3
        mouseflag = 1 : istopflag = 0
        SETCURSOR VARPTR(icrosshair(0)) : GOSUB SelectExpansion : IF ibreak = 1 THEN GOTO Point8End
        SETCURSOR VARPTR(iwatch(0))
        i = ii : GOSUB FindPoint1 : rhom = p1 : W1 = a1
        a1 = SQR(TT1*gamma1*R1/(gamma4*R4))
        expangle = -(ATN(1#/(uu2 - aa2)) - ATN(1#/(uu5 - aa5)))*expangle/100# + ATN(1#/(u1 - a1))
        iescapeflag = 0
        IF expangle < ATN(1#/(uu5 - aa5)) THEN expangle = ATN(1#/(uu5 - aa5)) : iescapeflag = 1
        Grid$ = "Blob"
        u3 = 2#/((gamma1 + 1#)*TAN(expangle)) + (gamma1 - 1#)*uu2/(gamma1 + 1#) + 2#*aa2/(gamma1 + 1#)
        a3 = (gamma1 - 1#)*(uu2 - u3)/2# + aa2
        TT3 = a3^2*gamma4*R4/(gamma1*R1)
        rho3 = rho1*(TT3/TT1)^(1#/(gamma1 - 1#))
        IF rhom > 1# THEN W3 = (u3 + u1)/2# ELSE W3 = 3#*(u3 - u1)/(2#*rhom + 1#) + W1
        x3 = (L1L2#/us2 - t1 + x1*(W1 + W3)/(2#*W1*W3))/((W1 + W3)/(2#*W1*W3) - 1#/(u3 - a3))
        t3 = x3/(u3 - a3) + L1L3#/us2
        p1 = rho1*R1*TT1/R4
        p3 = rho3*R1*TT3/R4
        rhom = (rhom*rho1/rho3)*(p3/p1)^(1#/gammablob)
        a1 = W3 : p3 = rhom : i = ii : j = ii : x2 = x3 : t2 = t3
        GOSUB SavePoint
        PointLocx(max) = x3 : PointLoct(max) = t3
        PointType(max) = 6 : iPoint(max) = i : jPoint(max) = j
        GOSUB DrawPoint : GOSUB DrawBlob
        IF iescapeflag = 1 THEN GOSUB BlobEscape
    Point8End:
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR
        RETURN

MeshSplit:
        IF imeshflag = 1 THEN imeshflag = 0 : MENU 4, 8, 1 : MENU 4, 1, 1 : MENU 4, 2, 1 : MENU 4, 5, 1 : MENU 4, 6, 1 : MENU 4, 10, 1 : RETURN
        IF imeshflag = 0 THEN imeshflag = 1 : MENU 4, 8, 2 : MENU 4, 1, 0 : MENU 4, 2, 0 : MENU 4, 5, 0 : MENU 4, 6, 0 : MENU 4, 10, 0 : RETURN

ErasePoint:
        GOSUB LocateErase : IF ibreak = 1 THEN GOTO EPointEnd
        IF ii <= 7 THEN BEEP : BEEP : GOTO EPointEnd
        Grid$ = "Blank"
        LSET gg$ = Grid$ : PUT #1, ii
```

```
            IF ii = max THEN max = max - 1
            PointType(ii) = 5
            CLS : GOSUB RePlot
        EPointEnd:
            IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo
            mouseflag = 0
            RETURN

  InfoMenu:
            MenuItem = MENU(1)
            ON MenuItem GOSUB DisplayInfo, GetInfo
            RETURN

  DisplayInfo:
            IF infoflag = 1 THEN infoflag = 0 : MENU 5, 1, 1 : MENU 5, 2, 0 : WINDOW CLOSE 2 : WINDOW 3 : MOUSE OFF ELSE infoflag = 1 : MENU 5, 1, 2 :
               MENU 5, 2, 1 : GOSUB DrawInfoBox
            RETURN

  GetInfo:
            GOSUB LocateInfo
            RETURN

  ScaleMenu:
            MenuItem = MENU(1)
            ON MenuItem GOSUB Zoom, ScrollScreen
            RETURN

  Zoom:
        ' draw dialog box to enter new magnification
            WINDOW 1, "", (186, 130) - (326, 190), -2
            RESTORE 6 : CALL TEXTSIZE(12) : CALL TEXTFACE(0) : CALL TEXTFONT(3)
            READ as$ : MOVETO (140# - LEN(as$)*6.7)/2#, 20 : PRINT as$
            BUTTON 1, 1, "OK", (94, 32) - (134, 47)
            EDIT FIELD 1, STR$(magfactor#), (14, 32) - (54, 47), , 2
            DIALOG OFF : s = DIALOG(0) : INITCURSOR
        Zoom2:
            WHILE s <> 1 AND s <> 6 : s = DIALOG(0) : WEND
            magfactor# = VAL(EDIT$(1)) : IF magfactor# < 1# THEN BEEP : GOTO Zoom2
            SETCURSOR VARPTR(iwatch(0))
            WINDOW CLOSE 1
            CLS
            GOSUB RePlot
            DIALOG ON
            RETURN

  ScrollScreen:
            WINDOW 1, "", (186, 130) - (326, 190), -2
            RESTORE 3000 : CALL TEXTSIZE(12) : CALL TEXTFACE(0) : CALL TEXTFONT(3)
            READ as$ : MOVETO (140# - LEN(as$)*6.7)/2#, 20 : PRINT as$
            BUTTON 1, 1, "OK", (94, 32) - (134, 47)
            EDIT FIELD 1, "10", (14, 32) - (54, 47), , 2
            DIALOG OFF : s = DIALOG(0) : INITCURSOR
        Scroll2:
            WHILE s <> 1 AND s <> 6 : s = DIALOG(0) : WEND
            ifirstScroll = CINT((VAL(EDIT$(1))/100#)*490#*magfactor#/(1# + L1L2))
            IF ifirstScroll < 0 THEN GOTO Scroll2
            WINDOW CLOSE 1
            DIALOG ON
        ScrollScreen2:
            GOSUB LocateScroll
            IF ibreak = 1 THEN GOTO ScrollScreenEnd
            GOSUB GetScroll
            CLS : GOSUB RePlot
            GOTO ScrollScreen2
        ScrollScreenEnd:
            IF infoflag = 1 THEN GOSUB FindInfo2
            mouseflag = 0
            INITCURSOR
            RETURN

 CopyMenu:
            MenuItem = MENU(1)
            ON MenuItem GOSUB CopyFlow
            RETURN

 CopyFlow:
            CALL HIDEPEN : LINE (0, 0) - (508, 318), , b
            PICTURE OFF : Image$ = PICTURE$
            OPEN "CLIP:PICTURE" FOR OUTPUT AS 2 : SETCURSOR VARPTR (iwatch(0)) : PRINT #2, PICTURE$
            CLOSE #2
            PICTURE ON : CALL SHOWPEN : PICTURE, Image$ : LINE (0, 0) - (508, 318), 30, b
            RETURN

'---------------------------------------------
' HANDLE THE VARIOUS DIALOG EVENTS

ButtonEvent:
            ButtonPressed = DIALOG(1)
            IF WINDOW(0) = 1 AND ButtonPressed = 1 THEN BUTTON 1, 2 : BUTTON 2, 1 : as$ = "n" : RETURN
            IF WINDOW(0) = 1 AND ButtonPressed = 2 THEN BUTTON 1, 1 : BUTTON 2, 2 : as$ = "y" : RETURN
            IF WINDOW(0) = 1 AND ButtonPressed = 3 THEN GOTO CloseWindow1
            IF WINDOW(0) = 2 AND ButtonPressed = 1 THEN GOSUB InData : IF jjflag = 1 THEN GOTO CloseWindow2
            IF WINDOW(0) = 2 AND ButtonPressed = 2 THEN BUTTON 2, 2 : BUTTON 3, 1 : ilangleyflag = 0 : RETURN
            IF WINDOW(0) = 2 AND ButtonPressed = 3 THEN BUTTON 2, 1 : BUTTON 3, 2 : ilangleyflag = 1 : RETURN
            IF WINDOW(0) = 2 AND ButtonPressed = 4 THEN IF iheatflag = 1 THEN iheatflag = 0 : BUTTON 4, 1 ELSE iheatflag = 1 : BUTTON 4, 2 ELSE REM
            IF WINDOW(0) = 4 AND ButtonPressed = 1 THEN ipitotpoint = 1 : BUTTON 1, 2 : BUTTON 2, 1 : BUTTON 3, 1 : BUTTON 4, 1
            IF WINDOW(0) = 4 AND ButtonPressed = 2 THEN ipitotpoint = 2 : BUTTON 1, 1 : BUTTON 2, 2 : BUTTON 3, 1 : BUTTON 4, 1
            IF WINDOW(0) = 4 AND ButtonPressed = 3 THEN ipitotpoint = 3 : BUTTON 1, 1 : BUTTON 2, 1 : BUTTON 3, 2 : BUTTON 4, 1
            IF WINDOW(0) = 4 AND ButtonPressed = 4 THEN ipitotpoint = 4 : BUTTON 1, 1 : BUTTON 2, 1 : BUTTON 3, 1 : BUTTON 4, 2
            IF WINDOW(0) = 4 AND ButtonPressed = 5 THEN GOTO DrawPitot
            IF WINDOW(0) = 4 AND ButtonPressed = 6 THEN GOTO CloseWindow4
            RETURN

EditEvent:
            IF WINDOW(0) = 2 THEN k = DIALOG(2) : EDIT FIELD k : j = k : i = 0 : GOTO InData
            RETURN

Activate:
            IF ipitotflag = 1 THEN WINDOW 3 : GOSUB RePicture : SETCURSOR VARPTR(icrosshair(0)) : GOTO SelectPitot
            RETURN

GoAway:
            WGoAway = DIALOG(4)
            IF WGoAway = 4 THEN GOTO CloseWindow4
            RETURN

Refresh:
            iWDirty = DIALOG(5)
            IF iDirty = 1 THEN IF iWDirty = 3 THEN SETCURSOR VARPTR (iwatch(0)) : GOSUB RePicture
            IF iDirty = 1 THEN IF iWDirty = 2 THEN SETCURSOR VARPTR (iwatch(0)) : GOSUB RePicture : GOSUB FindInfo2
            iDirty = 1 : INITCURSOR
        RETURN

ReturnEvent:
        IF WINDOW(0) = 1 AND iwindowflag = 0 THEN GOTO CloseWindow1
        IF WINDOW(0) = 2 THEN GOSUB InData : IF jjflag = 1 THEN GOTO CloseWindow2
        RETURN

TabEvent:
        IF WINDOW(0) = 2 THEN j = k + 1
        IF WINDOW(0) = 2 THEN IF j = 7 THEN EDIT FIELD 1 : i = 1 - k : GOTO InData
        IF WINDOW(0) = 2 THEN EDIT FIELD j + 1 : i = i + 1 : GOTO InData
        RETURN

'-------------------------------------------------
' HANDLE THE VARIOUS MOUSE EVENTS

MousePosition:
        WINDOW OUTPUT 2
'        find mouse position if on scroll bar and scroll window 2 accordingly
        xxx = MOUSE(1) : ttt = MOUSE(2) : MouseSub = 0
        IF xxx > 94 AND ttt < 14 THEN MouseSub = 1
        IF xxx > 94 AND ttt > 14 AND ttt < 66 THEN MouseSub = 2
        IF xxx > 94 AND ttt > 66 THEN MouseSub = 3
        IF MouseSub = 0 THEN WINDOW OUTPUT 4 : RETURN
        ON MouseSub GOSUB UpArrow, GreyArea, DownArrow
'        draw scroll bar with scroll box at appropriate position
        PICTURE, ScrollBar$
        iTT = INT(34*(itop - 1)/10)
        PICTURE (0, iTT), ScrollBox$
        WINDOW OUTPUT 3
        RETURN

MouseReturn:
        RETURN

UpArrow:
        IF itop = 1 THEN GOSUB PrintInfo : RETURN ELSE itop = itop - 1
        GOSUB PrintInfo
        RETURN

GreyArea:
        IF ttt > iTT + 14 AND ttt < iTT + 29 THEN GOSUB Drag : GOSUB PrintInfo : RETURN
        IF ttt < iTT + 14 THEN IF itop < 5 THEN itop = 1 ELSE itop = itop - 4 ELSE REM
        IF ttt > iTT + 29 THEN IF itop > 7 THEN itop = 6 ELSE itop = itop + 4 ELSE REM
        GOSUB PrintInfo
        RETURN

Drag:
        ittold = iTT
        PENNORMAL : PENMODE 10
        WHILE MouseClick < 5
                MOVETO 95, iTT + 15 : CALL LINE (14, 0) : CALL LINE (0, 15) : CALL LINE (-14, 0) : CALL LINE (0, -15)
                MouseClick = MOUSE(0) + 4
                ClickX = MOUSE(5) : ClickY = MOUSE(6)
                IF ClickY < 30 THEN ClickY =22
                IF ClickY > 50 THEN ClickY = 56
                MOVETO 95, iTT + 15 : CALL LINE (14, 0) : CALL LINE (0, 15) : CALL LINE (-14, 0) : CALL LINE (0, -15)
                iTT = ClickY - 22
        WEND
        PENMODE 8
        IF ClickX < 94 THEN iTT = ittold : RETURN
        iTT = ClickY - 22
        itop = INT(iTT*5/34 + 1)
        RETURN

DownArrow:
        IF itop = 6 THEN GOSUB PrintInfo : RETURN ELSE itop = itop + 1
        GOSUB PrintInfo
        RETURN

PrintInfo:
        LINE (0, 15) - (93, 80), 30, bf
        istart = 15 : ipos = 0
        ON itop GOTO Print1, Print2, Print3, Print4, Print5, Print6
    Print1:
        ipos = ipos + istart + 13 : MOVETO 5, ipos : PRINT "i"; PTAB(40); "="; i : istart = 0
    Print2:
        ipos = ipos + istart + 13 : MOVETO 5, ipos : PRINT "j"; PTAB(40); "="; j : istart = 0
    Print3:
        ipos = ipos + istart + 13 : aa = x : GOSUB Shorten : MOVETO 5, ipos : PRINT "x"; PTAB(40); "="; as$ : istart = 0
    Print4:
        ipos = ipos + istart + 13 : aa = t : GOSUB Shorten : MOVETO 5, ipos : PRINT "t"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN RETURN
    Print5:
        ipos = ipos + istart + 13 : aa = u : GOSUB Shorten : MOVETO 5, ipos : PRINT "u"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN RETURN
    Print6:
        ipos = ipos + istart + 13 : aa = a : GOSUB Shorten : MOVETO 5, ipos : PRINT "a"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN RETURN
        ipos = ipos + istart + 13 : aa = p : GOSUB Shorten : MOVETO 5, ipos : PRINT "p"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN RETURN
        ipos = ipos + istart + 13 : aa = TT : GOSUB Shorter : MOVETO 5, ipos : PRINT "TT"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN RETURN
        ipos = ipos + istart + 13 : aa =rho  : GOSUB Shorten : MOVETO 5, ipos : PRINT "rho"; PTAB(40); "="; as$ : istart = 0 : IF ipos = 67 THEN
RETURN
        RETURN

Shorten:
        IF ABS(aa) < .001# THEN as$ = ".000" + MID$(STR$(aa*100000#), 2, 1) : RETURN
        IF ABS(aa) < .01# THEN as$ = ".00" + MID$(STR$(aa*10000#), 2, 2) : RETURN
        IF ABS(aa) < .1# THEN as$ = ".0" + MID$(STR$(aa*10000#), 2, 3) : RETURN
        IF ABS(aa) < 1# THEN as$ = LEFT$(STR$(aa), 6) ELSE as$ = LEFT$(STR$(aa), 7)
        RETURN

'-------------------------------------------------
' MISCELLANEOUS ROUTINES

ErrorFix:
        IF (ERR=64) AND (ERL=7) THEN RESUME Recover
        IF (ERL=100) THEN RESUME PrintEnd
        END

Recover:
        ON ERROR GOTO 0
        MENU 1, 1, 1 : MENU 1, 2, 0 : MENU 1, 3, 0 : MENU 1, 4, 0
        RETURN

InData:
        jjflag = 1
        P4P1 = VAL(EDIT$(1)) : IF P4P1 <= 0# THEN jjflag = 0
        p1p10 = VAL(EDIT$(2)) : IF p1p10 <= 0# THEN jjflag = 0
        T4T1 = VAL(EDIT$(3)) : IF T4T1 <= 0# THEN jjflag = 0
        L1L2# = VAL(EDIT$(4)) : IF L1L2# <= 0# THEN jjflag = 0
        dL2 = VAL(EDIT$(5)) : IF dL2 <= 0# THEN jjflag = 0
        Gas$ = UCASE$(EDIT$(6)) : GOSUB CheckGas : IF igasflag = 1 THEN Driver$ = Gas$ : GOSUB GetDriver ELSE jjflag = 0
        Gas$ = UCASE$(EDIT$(7)) : GOSUB CheckGas : IF igasflag = 1 THEN Test$ = Gas$ : GOSUB GetTest ELSE jjflag = 0
        RETURN

CheckGas:
        IF Gas$ = "AIR" OR Gas$ = "ARGON" OR Gas$ = "HELIUM" OR Gas$ = "CO2" OR Gas$ = "NITROGEN" THEN igasflag = 1 ELSE igasflag = 0
        RETURN

GetDriver:
        IF Driver$ = "AIR" THEN gamma4 = 1.4# : R4 = 287#
        IF Driver$ = "ARGON" THEN gamma4 = 1.667# : R4 = 208.13#
        IF Driver$ = "HELIUM" THEN gamma4 = 1.667# : R4 = 2077.03#
        IF Driver$ = "CO2" THEN gamma4 = 1.29# : R4 = 188.92#
        IF Driver$ = "NITROGEN" THEN gamma4 = 1.4# : R4 = 296.8#
        RETURN

GetTest:
        IF Test$ = "AIR" THEN gamma1 = 1.4# : R1 = 287#
        'IF Test$ = "ARGON" THEN gamma1 = 1.667# : R1 = 208.13#
        'IF Test$ = "HELIUM" THEN gamma1 = 1.667# : R1 = 2077.03#
        'IF Test$ = "CO2" THEN gamma1 = 1.29# : R1 = 188.92#
        'IF Test$ = "NITROGEN" THEN gamma1 = 1.4# : R1 = 296.8#
        RETURN

DrawTable:
      ' get data from file
     IF infoflag = 1 THEN WINDOW CLOSE 2
        WINDOW 1, "", (186, 130) - (326, 190), -2
        RESTORE 241 : CALL TEXTSIZE(12) : CALL TEXTFACE(0) : CALL TEXTFONT(3)
        EDIT FIELD 1, "1", (8, 6) - (42, 21), , 2
        READ as$ : MOVETO 50, 18 : PRINT as$
        EDIT FIELD 2, "30", (8, 39) - (42, 54), , 2
        READ as$ : MOVETO 50, 51 : PRINT as$ : EDIT FIELD 1 : ifield = 1
        BUTTON 1, 1, "OK", (94, 22) - (134, 37)
        DIALOG OFF : INITCURSOR
     Table2:
        s = DIALOG(0)
        WHILE s <> 1 AND s <> 6 AND s <> 7 : s = DIALOG(0) : WEND
        IF s = 7 THEN IF ifield = 2 THEN EDIT FIELD 1 : ifield = 1 ELSE EDIT FIELD 2 : ifield = 2 ELSE REM
        IF s = 7 THEN GOTO Table2
        icommence = VAL(EDIT$(1))
        iconclude = VAL(EDIT$(2))
        SETCURSOR VARPTR(iwatch(0))
        WINDOW CLOSE 1
        CLS : DIALOG ON
        ii = -1
        PICTURE ON
        CALL PENSIZE(1, 1)
        CALL TEXTFONT(4) : CALL TEXTSIZE(12) : CALL TEXTFACE(1)
        MOVETO 5, 5 : LINETO 463, 5 : MOVETO 5, 5 : LINETO 5, 34
        MOVETO 14, 25 : PRINT "n" : MOVETO 33, 5 : LINETO 33, 34
        MOVETO 40, 25 : PRINT "n" ;: CALL TEXTSIZE(9) : MOVE 1, 3 : PRINT "i-1" : CALL TEXTSIZE(12) : MOVETO 72, 5 : LINETO 72, 34
        MOVETO 77, 25 : PRINT "n" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "j-1" : CALL TEXTSIZE(12) : MOVETO 108, 5 : LINETO 108, 34
        MOVETO 120, 25 : PRINT "x" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 153, 5 : LINETO 153, 34
        MOVETO 167, 25 : PRINT "t" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 199, 5 : LINETO 199, 34
        MOVETO 213, 25 : PRINT "u" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 251, 5 : LINETO 251, 34
        MOVETO 270, 25 : PRINT "a" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 305, 5 : LINETO 305, 34
        MOVETO 323, 25 : PRINT "p" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 360, 5 : LINETO 360, 34
        MOVETO 379, 25 : PRINT "T" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12) : MOVETO 412, 5 : LINETO 413, 34
        MOVETO 429, 25 : PRINT "r" ;: MOVE 1, 3 : CALL TEXTSIZE(9) : PRINT "n" : CALL TEXTSIZE(12)
        MOVETO 463, 5 : LINETO 463, 34 : MOVETO 5, 34 : LINETO 463, 34 : MOVETO 5, 34
        CALL TEXTFACE(0) : CharSize = 7#
        WHILE ii + icommence < iconclude
                ii = ii + 1
                GET #1, ii + 5 + icommence - 1
                IF EOF(1) THEN GOTO TableEnd
                i2$ = STR$(CVI(i$)) : j2$ = STR$(CVI(j$))
                aa = CVD(x$) : GOSUB Shorten : x$ = as$
                aa = CVD(t$) : GOSUB Shorten : t$ = as$
                aa = CVD(u$) : GOSUB Shorten : u$ = as$
                aa = CVD(a$) : GOSUB Shorten : a$ = as$
                aa = CVD(p$) : GOSUB Shorten : p$ = as$
                aa = CVD(TT$) : GOSUB Shorten : TT$ = as$
                aa = CVD(rho$) : GOSUB Shorten : rho$ = as$
                Grid$ = gg$ : iblank = INSTR(Grid$, " ") : Grid$ = LEFT$(Grid$, iblank - 1)
                IF Grid$ = "Blank" THEN GOTO TableEnd2
      ' print data to table and to clipboard
                IF ii > 8 THEN SCROLL (0, 0) - (508, 310), 0, -29 : MOVETO 5, 237
                CALL LINE (0, 29)
                MOVE 4, -9 : PRINT STR$(ii + icommence);
                MOVE 24 - CharSize*LEN(STR$(ii + icommence)), -20 : CALL LINE(0, 29) : MOVE 9, -9 : IF Grid$ <> "Contact"  AND Grid$ <> "Interior2"
                        THEN PRINT i2$; ELSE i2$ = " -" : PRINT i2$;
                MOVE 30 - CharSize*LEN(i2$), -20 : CALL LINE(0, 29) : MOVE 8, -9 : IF Grid$ <> "Expansion" AND Grid$ <> "Interior1" THEN PRINT j2$
                        ELSE j2$ = " -" : PRINT j2$;
                MOVE 28 - CharSize*LEN(j2$), -20 : CALL LINE(0, 29) : MOVE 2, -9 : PRINT x$;
                MOVE 43 - CharSize*LEN(x$), -20 : CALL LINE(0, 29) : MOVE 1, -9 : PRINT t$;
                MOVE 45 - CharSize*LEN(t$), -20 : CALL LINE(0, 29) : MOVE 1, -9 : PRINT u$;
                MOVE 51 - CharSize*LEN(u$), -20 : CALL LINE(0, 29) : MOVE 1, -9 : PRINT a$;
                MOVE 53 - CharSize*LEN(a$), -20 : CALL LINE(0, 29) : MOVE 1, -9 : PRINT p$;
                MOVE 54 - CharSize*LEN(p$), -20 : CALL LINE(0, 29) : MOVE 2, -9 : PRINT TT$;
                MOVE 51 - CharSize*LEN(TT$), -20 : CALL LINE(0, 29) : MOVE 1, -9 : PRINT rho$;
                MOVE 49 - CharSize*LEN(rho$), -20 : CALL LINE (0, 29) : MOVE -458, 0
        TableEnd2:
        WEND
    TableEnd:
        CALL LINE (458, 0)
        PICTURE OFF : Image$ = PICTURE$
        OPEN "CLIP:PICTURE" FOR OUTPUT AS 2
        PRINT #2, Image$
        CLOSE #2
        PICTURE ON
        RETURN

Old:
        MENU 7, 0, 1 : MENU 4, 1, 1 : MENU 4, 2, 1 : MENU 4, 3, 1 : MENU 4, 10, 1 : MENU 4, 5, 1 : MENU 4, 4, 1 : MENU 4, 6, 1
        RETURN

Rename3:
        i = 0
        j = i4
        x3 = x1
        t3 = t1
        u3 = u1
        a3 = a1
        p3 = p1
        TT3 = TT1
        rho3 = rho1
        RETURN

Rename4:
        i = ii1
        j = ii1
        x1 = x2
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        t3 = t2
        u3 = u2
        a3 = W2      ' Blob velocity
        p3 = rhom     ' Blob density
        TT3 = TT2
        rho3 = rho2
        RETURN

FindInfo:
        GOSUB FindPoint
    FindInfo2:
        WINDOW 2
         draw scroll bar with scroll box at top position
        PICTURE, ScrollBar$
        PICTURE, ScrollBox$
         print info in window 2
        CALL TEXTFONT (1) : CALL TEXTSIZE (12) : CALL TEXTFACE (1)
        aaa$ = "Point Info" : MOVETO 12, 11 : PRINT aaa$
        LINE (0, 14) - (94, 14)
        CALL TEXTFONT (4) : CALL TEXTSIZE (9) : CALL TEXTFACE (32)
        MOVETO 5, 28 : PRINT "i"; PTAB(40); "="; i
        MOVETO 5, 41 : PRINT "j"; PTAB(40); "="; j
        aa = x : GOSUB Shorten : MOVETO 5, 54 : PRINT "x"; PTAB(48); "="; as$
        aa = t : GOSUB Shorten : MOVETO 5, 67 : PRINT "t"; PTAB(48); "="; as$
        WINDOW OUTPUT 3 : itop = 1
        RETURN

GetScroll:
        start = TIMER : WHILE (TIMER - start) =< 1 : WEND
        s = ABS(s) + ABS(MOUSE(0)) - 1
        IF s = 1 THEN iScrolling = ifirstScroll
        IF s = 2 THEN iScrolling = 2*ifirstScroll
        IF s = 3 THEN iScrolling = 3*ifirstScroll
        IF xxx < 169 AND ttt < 106 THEN iScrollX = iScrollX + iScrolling : iScrollT = iScrollT + iScrolling : GOTO GetScrollEnd
        IF xxx < 169 AND ttt => 106 AND ttt =< 212 THEN iScrollX = iScrollX + iScrolling : GOTO GetScrollEnd
        IF xxx < 169 AND ttt > 212 THEN iScrollX = iScrollX + iScrolling : iScrollT = iScrollT - iScrolling : GOTO GetScrollEnd
        IF xxx => 169 AND xxx =< 339 AND ttt < 106 THEN iScrollT = iScrollT + iScrolling : GOTO GetScrollEnd
        IF xxx => 169 AND xxx =< 339 AND ttt > 106 AND ttt < 212 THEN iScrollX = 0# : iScrollT = 0# : GOTO GetScrollEnd
        IF xxx => 169 AND xxx =< 339 AND ttt > 212 THEN iScrollT = iScrollT - iScrolling : GOTO GetScrollEnd
        IF xxx > 339 AND ttt < 106 THEN iScrollX = iScrollX - iScrolling : iScrollT = iScrollT + iScrolling : GOTO GetScrollEnd
        IF xxx > 339 AND ttt => 106 AND ttt =< 212 THEN iScrollX = iScrollX - iScrolling : GOTO GetScrollEnd
        IF xxx > 339 AND ttt > 212 THEN iScrollX = iScrollX - iScrolling : iScrollT = iScrollT - iScrolling
    GetScrollEnd:
        RETURN

ShowNumber:
        CALL TEXTFONT(4) : CALL TEXTSIZE(9) : CALL TEXTFACE(32) : CALL TEXTMODE(1)
        FOR i = 0 TO max
                x = PointLocx(i)
                t = PointLoct(i)
                ii = iPointFile(i)
                IF PointType(i) = 5 THEN GOTO NumberLoop
                MOVETO 490#*(x*magfactor# + L1L1#)/(1# + L1L2#) + 15 + iScrollX, 318# + iScrollT - 490#*t*magfactor#*tstretch/(1# + L1L2#)
                PRINT ii + 7
        NumberLoop:
            NEXT i
        RETURN

CloseField3:
        EDIT FIELD 1 : nextfield = 2
        LINE (126, 82) - (234, 94), 30, bf
        EDIT FIELD CLOSE 3
        RETURN

DrawPitot:
        SETCURSOR VARPTR(iwatch(0))
        WINDOW CLOSE 4
        WINDOW 4, "Pitot Plot", (40, 40) - (470, 338), 1
        PICTURE ON : CALL SHOWPEN
        pitotmax! = 0# : iii = 6
        FOR ii = 6 TO max
                IF iii = 6 THEN iidash = ii
                IF iii = 6 AND PointType(ii) = 8 THEN iii = ii
                IF iii = 6 AND PointType(ii) = 9 THEN iii = ii
                IF PointType(ii) = 7 THEN GET #1, ii : pitotmax = CVD(TT$) : IF pitotmax > pitotmax! THEN iii = ii : pitotmax! = pitotmax : GOTO
     DPEnd1 ELSE GOTO DPEnd1 ELSE REM
                IF ii = max AND PointType(ii) <> 8 AND PointType(ii) <> 9 THEN BEEP : BEEP : BEEP : GOTO CloseWindows
         DPEnd1:
        NEXT ii
        GET #1, iii
        tmax = CVD(t$) : pitotmax = CVD(TT$)
        spitot! = (p5p2*p2p1*.84/P4P1)*((gamma! + 1!)*(MS#*.88)^2/2!)^(gamma!/(gamma! - 1!))/((2!*gamma!/(gamma! + 1!)*(MS#*.88)^2 - (gamma! -
           1!)/(gamma! + 1!))^(1!/(gamma! - 1!))
        IF spitot! > pitotmax THEN pitotmax = spitot!
        GET #1, iidash
        tdash = CVD(t$) : pitotdash = CVD(TT$)
        TEXTFONT 4 : TEXTSIZE 9 : TEXTFACE 32
        LINE (28, 18) - (408, 261), , b : it = 28 : MOVETO 16, 271 : PRINT CINT((L1L2#/us2 + 1#/us20 - (tdash - (L1L2#/us2 + 1#/us20))/8#)*1000)/1000
        FOR iii = 0 TO 8
                it = it + 40 : MOVETO it, 261 : CALL LINE(0, -8)
                MOVE -12, 18 : PRINT CINT(((tdash - (L1L2#/us2 + 1#/us20))*iii/8 + L1L2#/us2 + 1#/us20)*1000)/1000
         DPEnd2:
        NEXT iii
        MOVETO 180, 284 : PRINT "t a4/ L2"
        ip = 261 : MOVETO 28, 261 : CALL LINE(8, 0) : MOVETO 8, 261 : PRINT " 0"
        FOR iii = 1 TO 6
                ip = ip - 40 : MOVETO 28, ip : CALL LINE(8, 0)
                ps$ = STR$(CINT(pitotmax*iii*100!/5!))
                ps1$ = " " + MID$(ps$, 2, 1)
                ps2$ = " " + MID$(ps$, 3, 1)
                MOVETO 13, ip -10 : PRINT ".";
                IF LEN(ps$) = 4 THEN ps3$ = " " + MID$(ps$, 4, 1) : MOVETO 8, ip - 14 : PRINT ps1$;
                IF LEN(ps$) = 2 THEN MOVETO 8, ip -1 : PRINT " 0"; : MOVETO 8, ip + 8 : PRINT ps1$; : GOTO DPEnd3
                MOVETO 8, ip -1 : IF LEN (ps$) = 4 THEN PRINT ps2$; ELSE PRINT ps1$;
                MOVETO 8, ip + 8 : IF LEN(ps$) = 4 THEN PRINT ps3$; ELSE PRINT ps2$;
         DPEnd3:
        NEXT iii
        MOVETO 5, 145 : PRINT "p" : MOVETO 5, 156 : PRINT "-"
        MOVETO 5, 162 : PRINT "p" : MOVETO 5, 174 : PRINT "4"
        IF PointType(iidash) <> 8 AND PointType(iidash) <> 9 THEN MOVETO 388, 261 - pitotdash*200/pitotmax
        IF PointType(iidash) = 8 OR PointType(iidash) = 9 THEN  GET #1, iPoint(iidash) : t = CVD(t$) : pitot = CVD(TT$) : MOVETO 68 + (t - (L1L2#/us2
           + 1#/us20))*320/(tdash - (L1L2#/us2 + 1#/us20)), 261 - pitot*200/pitotmax
        ii = iidash
        tcontact = 0# : Tblob = 0# : PENSIZE 2, 2
        WHILE ii <= max
                GET #1, ii
                IF PointType(ii) = 8 THEN tcontact = CVD(t$) : ii = iPoint(ii) : GOTO DPEnd4
                IF PointType(ii) = 9 THEN Tblob = CVD(t$) : ii = iPoint(ii) : GOTO DPEnd4
                t = CVD(t$) : pitot = CVD(TT$) : ii = iPoint(ii)
                LINETO 68 + (t - (L1L2#/us2 + 1#/us20))*320/(tdash - (L1L2#/us2 + 1#/us20)), 261 - pitot*200/pitotmax
         DPEnd4:
```

```
        NEXT
        IF iexpend = 0 THEN LINETO 60 + ((10/(uu3 - aa3) - 10/us20)*320/(tdash - (L1L20/us2 + 10/us20)), 261 - spitot1*200/pitotmax
        LINETO 60 + (tg2 - (L1L20/us2 + 10/us20))*320/(tdash - (L1L20/us2 + 10/us20)), 261 - spitot1*200/pitotmax
        spitot2 = (p20p10*.80/(p1p10*P4P)))*((gamma1 + 1))*(M20*.80)^2/2))^(gamma1/(gamma1 - 1)))/((2^*gamma1/(gamma1 + 1))*(M20*.80)^2 - (gamma1 -
            1))/(gamma1 + 1)))^(1/(gamma1 - 1)))
        LINETO 60 + (tg2 - (L1L20/us2 + 10/us20))*320/(tdash - (L1L20/us2 + 10/us20)), 261 - spitot2*200/pitotmax
        LINETO 60, 261 - spitot2*200/pitotmax
        LINETO 60, 261
        PENSIZE 1, 1
        IF Tblob <= 0! THEN GOTO pitotjump1
        MOVETO 60 + (Tblob - (L1L20/us2 + 10/us20))*320/(tdash - (L1L20/us2 + 10/us20)), 17
        PENSIZE 2, 2 : CALL MOVE(0, 1) : CALL LINE(0, 20) : MOVE -10, -25
        PENSIZE 1, 1 : PRINT "Blob"
    pitotjump1:
        IF tcontact <= 0! THEN GOTO pitotjump2
        MOVETO 60 + (tcontact - (L1L20/us2 + 10/us20))*320/(tdash - (L1L20/us2 + 10/us20)), 17
        PENSIZE 2, 2 : CALL MOVE(0, 1) : CALL LINE(0, 20) : MOVE -10, -30
        PENSIZE 1, 1 : PRINT "Contact"
    pitotjump2:
        PICTURE OFF : CALL HIDEPEN : Image$ = PICTURE$
        OPEN "CLIP:PICTURE" FOR OUTPUT AS 2
        PRINT #2, Image$
        CLOSE #2
        INITCURSOR
        RETURN

'---------------------------------------------------
' DIALOG BOX AND WINDOW DRAWING ROUTINES

DrawDialog1:
        WINDOW 1, "", (105, 50) - (400, 270), -2
        TEXTFACE (1)
        RESTORE 2
        FOR i = 1 TO 7
                READ as$
                IF i = 1 THEN MOVETO (2950 - LEN(as$)*8.7)/20, i*17 ELSE IF i <= 5 THEN MOVETO (2950 - LEN(as$)*7.3)/20, i*20
                IF i = 6 THEN MOVETO (2200 - LEN(as$)*7.3)/20,145
                IF i = 7 THEN MOVETO (2100 - LEN(as$)*7.3)/20,194
                PRINT as$
        NEXT i
        LINE (5,112) - (290,112)
        BUTTON 1, 2, "no", (220, 120) - (280, 140), 3
        BUTTON 2, 1, "yes", (220, 140) - (280, 160), 3
        BUTTON 3, 1, "OK", (210, 178) - (270, 202), 1
        as$ = "n"
        RETURN

DrawDialog2:
' setup variable input box
        WINDOW 2, "", (70, 30) - (442, 260), -2
        TEXTFACE(1)
        RESTORE 3
        READ as$
        MOVETO (288 - INT(LEN(as$)*8.3))/2), 25
        PRINT as$ : BUTTON 1, 1, "OK", (288, 10) - (352, 30), 1
        TEXTFACE 0
        MOVETO 10, 50 : LINETO 130, 50 : MOVETO 10, 50 : LINETO 10, 90 : LINETO 130, 90 : MOVETO 130, 50 : LINETO 130, 90
        MOVETO 130, 60 : LINETO 360, 60 : MOVETO 130, 80 : LINETO 360, 80 : MOVETO 250, 60 : LINETO 250, 80
        READ as$ : MOVETO 100 + (1200 - LEN(as$)*6.6)/20, 75 : PRINT as$
        READ as$ : MOVETO 1300 + (1200 - LEN(as$)*6.6)/20, 75 : PRINT as$
        READ as$ : MOVETO 2500 + (1100 - LEN(as$)*6.6)/20, 75 : PRINT as$
        MOVETO 130, 92 : LINETO 130, 112 : MOVETO 250, 82 : LINETO 250, 112 : MOVETO 360, 82 : LINETO 360, 112
        p%(0) = 22 : p%(1) = 107 : p%(2) = 130 : p%(3) = 113 : p%(4) = 140 : p%(5) = 110 : p%(6) = 130 : p%(7) = 113 : p%(8) = 140 : p%(9) = 107 :
            p%(10) = 140
        PAINTPOLY VARPTR(p%(0))
        p%(2) = 240 : p%(4) = 250 : p%(6) = 250 : p%(8) = 240 : p%(10) = 240
        PAINTPOLY VARPTR(p%(0))
        p%(2) = 250 : p%(4) = 260 : p%(6) = 250 : p%(8) = 260 : p%(10) = 260
        PAINTPOLY VARPTR(p%(0))
        p%(2) = 350 : p%(4) = 360 : p%(6) = 360 : p%(8) = 350 : p%(10) = 350
        PAINTPOLY VARPTR (p%(0))
        MOVETO 187, 107 : PRINT "L" : MOVETO 195, 109 : TEXTFONT(4) : TEXTSIZE (9) : PRINT "1" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 303, 107 : PRINT "L" : MOVETO 311, 109 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "2" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 130, 110 : LINETO 360, 110
        BUTTON 2, 2, "UQ", (10, 95) - (49, 110), 3
        BUTTON 3, 1, "Langley", (52, 95) - (120, 110), 3
        BUTTON 4, 1, "E.T.", (10, 130) - (50, 145), 2
        MOVETO 77, 137 : PRINT "p" : MOVETO 85, 139 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "4" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 77, 150 : PRINT "p" : MOVETO 85, 152 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "1" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 76, 140 : LINETO 90, 140
        MOVETO 96, 143 : PRINT "="
        EDIT FIELD 1, "10130", (110, 132) - (170, 148)
        MOVETO 197, 137 : PRINT "p" : MOVETO 205, 139 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "1" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 197, 150 : PRINT "p" : MOVETO 204, 152 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "10" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 196, 140 : LINETO 210, 140
        MOVETO 216, 143 : PRINT "="
        EDIT FIELD 2, "209.6", (230, 132) - (290, 148)
        MOVETO 19, 177 : PRINT "T" : MOVETO 27, 179 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "4" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 19, 191 : PRINT "T" : MOVETO 27, 193 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "1" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 18, 180 : LINETO 32, 180
        MOVETO 38, 183 : PRINT "="
        EDIT FIELD 3, "9.43", (52, 172) - (112, 188)
        MOVETO 143, 177 : PRINT "L" : MOVETO 151, 179 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "1" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 143, 191 : PRINT "L" : MOVETO 151, 193 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "2" : TEXTFONT(1) : TEXTSIZE(12)
        MOVETO 142, 180 : LINETO 156, 180
        MOVETO 162, 183 : PRINT "="
        EDIT FIELD 4, "0.6541", (176, 172) - (236, 188)
        MOVETO 267, 177 : PRINT "d" : MOVETO 267, 191 : PRINT "L" : MOVETO 275, 193 : TEXTFONT(4) : TEXTSIZE(9) : PRINT "2" : TEXTFONT(1) :
            TEXTSIZE(12)
        MOVETO 266, 180 : LINETO 280, 180
        MOVETO 286, 183 : PRINT "="
        EDIT FIELD 5, "0.01164", (300, 172) - (360, 188)
        READ as$ : MOVETO 23, 224 : PRINT as$
        EDIT FIELD 6, "Helium", (110, 212) - (170, 228), , 2
        READ as$ : MOVETO 180, 224 : PRINT as$
        EDIT FIELD 7, "Air", (298, 212) - (360, 228), , 2
        EDIT FIELD 1
        i = 0 : k = 1
        INITCURSOR
        RETURN

DrawWindow3:
  Set up flow field window - window 3
        WINDOW 3, "", (2, 22) - (510, 340), 3
        PICTURE ON : CALL SHOWPEN
        MENU 7, 0, 1 : MENU 3, 0, 1 : MENU 4, 0, 1 : MENU 5, 0, 1 : MENU 6, 0, 1
        IF iiflag = 1 THEN RETURN
' Draw expansion tube (including scales)
        GOSUB Tube : LINE (0, 0) - (500, 10) , 30, bf : LINE (0, 0) - (10, 318), 30, bf
        GOSUB Scales
        x3 = xg : t3 = tg : GOSUB DrawPoint
        x3 = xgblob : t3 = tgblob : GOSUB DrawPoint
        MENU 4, 1, 1 : MENU 4, 2, 1 : MENU 4, 3, 1 : MENU 4, 10, 1 : MENU 4, 5, 1 : MENU 4, 6, 1 : MENU 4, 4, 1
        INITCURSOR
        RETURN

DrawInfoBox:
        firstinfo = 1 : GOSUB LocateInfo
        WINDOW 2, "", (400, 260) - (510, 340), 3
        GOSUB FindInfo
        firstinfo = 0
        RETURN

DrawWindow4:
        WINDOW 4, "", (20, 40) - (200, 160), 3
        TEXTFACE(1)
        RESTORE 300
        READ aa$
        MOVETO (180 - LEN(aa$)*8.7)/2., 20 : PRINT aa$
        BUTTON 1, 2, "Test", (10, 30) - (80, 50), 3
        BUTTON 2, 1, "Contact", (100, 30) - (170, 50), 3
        BUTTON 3, 1, "Driver", (10, 60) - (80, 80), 3
        BUTTON 4, 1, "Blob", (100, 60) - (170, 80), 3
        BUTTON 5, 1, "Plot", (10, 90) - (80, 110), 1
        BUTTON 6, 1, "Exit", (100, 90) - (170, 110), 1
        RETURN

'___________________________________________
' WINDOW CLOSING ROUTINES

CloseWindow1:
        SETCURSOR VARPTR (iwatch(0))
        TEXTFACE(0)
        WINDOW CLOSE 1
'     Ask for name of database
        IF as$ = "n" THEN F$ = FILES$(0,"Type in name of new file ?") : iiflag = 0 ELSE IF as$ = "y" THEN F$ = FILES$(1,"TEXT") : iiflag = 1
        SETCURSOR VARPTR (iwatch(0))
        G$ = F$
        ON ERROR GOTO ErrorFix
?       OPEN F$ AS #1 LEN = 76
        ON ERROR GOTO 0
        iopenflag = 1
'     If old - read in variables and draw picture
        IF iiflag = 1 THEN GOSUB OldFile : GOSUB DrawWindow3 : GOSUB ReDraw : GOSUB Old : INITCURSOR : RETURN
        GOTO DrawDialog2

CloseWindow2:
        SETCURSOR VARPTR (iwatch(0))
        WINDOW CLOSE 2
        TEXTFACE(0)
        GOSUB NewCalcs
        IF Test$ <> "AIR" THEN BEEP : BEEP : BEEP : STOP
        GOSUB Mirels
        GOSUB Newfile
    'test gas point
        Grid$ = "FirstPoint" : PointType(max + 1) = 10
        i = max + 2 : iPoint(max + 1) = i
        j = 0 : jPoint(max + 1) = j
        slower1 = 0# : supper1 = .999999#*lm#/(us2/(uu2 - aa2) - 1#) : iflag = 3
        glower=(5#/8#)*(LOG(((1#-(slower1*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(slower1*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((slower1*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(slower1*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2/(rho2rho1*2#*lm#))*(slower1/(uu2-aa2)+L1L2#/us2)
        gupper=(5#/8#)*(LOG(((1#-(supper1*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(supper1*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((supper1*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(supper1*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2/(rho2rho1*2#*lm#))*(supper1/(uu2-aa2)+L1L2#/us2)
        IF glower*gupper > 0# THEN BEEP : BEEP : BEEP : STOP
        CALL zeroin(slower1, supper1, tol, iflag, s) : xg = s
        tg = xg/(uu2 - aa2) + L1L2#/us2
        x3 = xg : t3 = tg
        PointLocx(max + 1) = x3
        PointLoct(max + 1) = t3
            u3 = uu2
        a3 = aa2
        p3 = p2p1/P4P1
        TT3 = (a3^2)*(gamma4*R4)/(gamma1*R1)
        rho3 = p3*R4/(TT3*R1)
        GOSUB SavePoint
    'driver gas point
        PointType(max + 1) = 10
        i = 0 : iPoint(max + 1) = i
        j = max : jPoint(max + 1) = j
        PointLocx(max + 1) = x3
        PointLoct(max + 1) = t3
        a3 = aa3
        p3 = p3p4
        TT3 = a3^2
        rho3 = p3/TT3
        GOSUB SavePoint
    ' find density minimum
        xgblob = -1# : tgblob = -1#
        cpblob = 7#
        aalpha = 5#*R4 - cpblob*R1
        balpha = cpblob*R1
        calpha = R4 - R1
        dalpha = R1
        ealpha = 5#*R4*T3T4 - cpblob*R1*T2T1/T4T1
        falpha = cpblob*R1*T2T1/T4T1
        alphamin1 = -balpha/aalpha
        alphamin2 = (balpha/aalpha)^2 - balpha*dalpha/(aalpha*calpha) - balpha*falpha/(aalpha*ealpha) - dalpha*falpha/(calpha*ealpha)
        IF alphamin2 < 0# THEN IF p3p4/T3T4 < p3p4*R4*T4T1/(T2T1*R1) THEN rhom = p3p4/T3T4 : GOTO CloseW3 ELSE BEEP : BEEP : BEEP : GOTO EndW3 ELSE
            REM
        alphamin = alphamin1 - SQR(alphamin2)
        IF alphamin < 0# OR alphamin > 1# THEN alphamin = alphamin1 + SQR(alphamin2)
        IF alphamin < 0# OR alphamin > 1# THEN IF p3p4/T3T4 < p3p4*R4*T4T1/(T2T1*R1) THEN rhom = p3p4/T3T4 : GOTO CloseW3 ELSE BEEP : BEEP : BEEP :
            GOTO EndW3 ELSE REM
        rhom = R4*p3p4*((5#*R4 - cpblob*R1)*alphamin + cpblob*R1)/(((R4 - R1)*alphamin + R1)*((5#*R4*T3T4 - cpblob*R1*T2T1/T4T1)*alphamin +
            cpblob*R1*T2T1/T4T1))
        IF rhom > p3p4*R4*T4T1/(T2T1*R1) THEN IF p3p4/T3T4 > p3p4*R4*T4T1/(T2T1*R1) THEN BEEP : BEEP : BEEP : GOTO EndW3 ELSE rhom = p3p4/T3T4 : GOTO
            CloseW3 ELSE REM
        IF rhom < p3p4*R4*T4T1/(T2T1*R1) THEN IF p3p4/T3T4 < rhom THEN rhom = p3p4/T3T4
  CloseW3:
        IF rhom = p3p4/T3T4 THEN alphamin = 1#
        gammablob = (alphamin*5#*R4 + cpblob*(1# - alphamin)*R1)/(alphamin*3#*R4 + (cpblob - 2#)*(1# - alphamin)*R1)
    ' adjust blob density to come into thermal equilibrium with test gas
        IF iheatflag = 1 THEN Tblob = (alphamin*5#*R4*T3T4 + cpblob*(1# - alphamin)*R1*T2T1/T4T1)/(alphamin*5#*R4 + cpblob*(1# - alphamin)*R1)
        IF iheatflag = 1 THEN rhom = rhom*Tblob*T4T1/T2T1
    ' calculate blob trajectory
        ss = p3p4*R4*T4T1/(T2T1*R1*rhom)
        sl1 = 0# : vv = (2#*ss/(2# + ss))^(5#/4#)
        su1 = .999999#*lm#/(vv*(us2/(uu2 - aa2) - 1#))
        iflag = 4
        glower=(5#/8#)*(LOG(((1#-(sl1*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(sl1*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((sl1*vv*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(sl1*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2*vv/(rho2rho1*2#*lm#))*(sl1/(uu2-aa2)+L1L2#/us2
        gupper=(5#/6#)*(LOG((1#-(sul*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(sul*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((sul*vv*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(sul*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2*vv/(rho2rho1*2#*lm#))*(sul/(uu2-aa2)+L1L2#/us2)
        IF glower*gupper > 0# THEN BEEP : BEEP : BEEP : STOP
        CALL zeroin(sll, sul, tol, iflag, s) : xgblob = s
        tgblob = xgblob/(uu2 - aa2) + L1L2#/us2
      ' first blob point
        Grid$ = "Blob" : PointType(max + 1) = 6
        i = 0 : iPoint(max + 1) = 0
        j = 0 : jPoint(max + 1) = 0
        x3 = xgblob : PointLocx(max + 1) = x3
        t3 = tgblob : PointLoct(max + 1) = t3
        u3 = uu2
        a3 = uu2*(1# - 2#*ss*((tgblob*us2 - L1L2# - xgblob)/lm#)^.8#/(2# + ss)) + us2
        p3 = (rhom*P4P1*R1)/(rho2rho1*R4*T4T1) : TT3 = T2T1/T4T1 : rho3 = p3p4*R4*T4T1/(T2T1*R1)
        GOSUB SavePoint
    EndM3:
    ' Acceleration tube Mirels
        tg2 = (lm2# + 1#)/us20 + L1L2#/us2
        IF tg2 > L1L2#/us2 + 1#/uu20 THEN GOSUB LaminarMirels
        GOTO DrawWindow3

LaminarMirels:
        sll = L1L2#/us2 + 1#/us20
        sul = L1L2#/us2 + 1#/uu20
        iflag = 5
        glower = LOG(1# - SQR((us20*(sll - L1L2#/us2) - 1#)/lm2#)) + SQR((us20*(sll - L1L2#/us2) - 1#)/lm2#) + us20*(sll -
            L1L2#/us2)/(2#*lm2#*rho20rho10)
        gupper = LOG(1# - SQR((us20*(sul - L1L2#/us2) - 1#)/lm2#)) + SQR((us20*(sul - L1L2#/us2) - 1#)/lm2#) + us20*(sul -
            L1L2#/us2)/(2#*lm2#*rho20rho10)
        IF glower*gupper > 0# THEN BEEP : BEEP : BEEP : STOP
        CALL zeroin(sll, sul, tol, iflag, s) : tg2 = s
        RETURN

CloseWindow4:
        SETCURSOR VARPTR(iwatch(0))
        WINDOW CLOSE 4
        WINDOW 3
        GOSUB RePlot2
        IF infoflag = 1 THEN iDirty = 0 : GOSUB FindInfo2
        mouseflag = 0 : INITCURSOR : ipitotflag = 0
        MENU 3, 1, 2 : MENU 3, 2, 0
        IF igridflag = 1 THEN MENU 3, 4, 2 ELSE MENU 3, 4, 1
        IF inumberflag = 1 THEN MENU 3, 6, 2 ELSE MENU 3, 6, 1
        MENU 3, 8, 2 : MENU 3, 10, 1
        MENU 4, 1, 1 : MENU 4, 2, 1 : MENU 4, 3, 1 : MENU 4, 4, 1 : MENU 4, 5, 1 : MENU 4, 6, 1 : MENU 4, 8, 1 : MENU 4, 10, 1
        IF infoflag = 1 THEN MENU 5, 1, 2 : MENU 5, 2, 1 ELSE MENU 5, 1, 1
        MENU 6, 1, 1 : MENU 6, 2, 1 : MENU 7, 1, 1
        INITCURSOR
        RETURN

'______________________________________________
' FILE HANDLING ROUTINES

NewFile:
      ' Save initial conditions and geometry
        FIELD #1, 8 AS P4P1$, 8 AS p1p10$, 8 AS gamma4$, 8 AS gamma1$, 8 AS L1L2$, 8 AS R4$, 8 AS R1$, 8 AS T4T1$
        LSET P4P1$ = MKD$(P4P1)
        LSET p1p10$ = MKD$(p1p10)
        LSET gamma4$ = MKD$(gamma4)
        LSET gamma1$ = MKD$(gamma1)
        LSET L1L2$ = MKD$(L1L2#)
        LSET R4$ = MKD$(R4)
        LSET R1$ = MKD$(R1)
        LSET T4T1$ = MKD$(T4T1)
        PUT #1, 2
      ' Save derived variables
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 8 AS p2p1$, 8 AS p3p4$, 8 AS T3T4$, 8 AS T2T1$, 8 AS M2$, 8 AS M3$, 8 AS aa2$, 8 AS aa3$, 8 AS uu2$
        LSET p2p1$ = MKD$(p2p1)
        LSET p3p4$ = MKD$(p3p4)
        LSET T3T4$ = MKD$(T3T4)
        LSET T2T1$ = MKD$(T2T1)
        LSET M2$ = MKD$(M2#)
        LSET M3$ = MKD$(M3#)
        LSET aa2$ = MKD$(aa2)
        LSET aa3$ = MKD$(aa3)
        LSET uu2$ = MKD$(uu2)
        PUT #1, 3
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 8 AS uu3$, 8 AS rho2rho1$, 8 AS rho3rho4$, 8 AS lm$, 8 AS p20p10$, 8 AS p5p2$, 8 AS T5T2$, 8 AS T20T10$, 8 AS us2$
        LSET uu3$ = MKD$(uu3)
        LSET rho2rho1$ = MKD$(rho2rho1)
        LSET rho3rho4$ = MKD$(rho3rho4)
        LSET lm$ = MKD$(lm#)
        LSET p20p10$ = MKD$(p20p10)
        LSET p5p2$ = MKD$(p5p2)
        LSET T5T2$ = MKD$(T5T2)
        LSET T20T10$ = MKD$(T20T10)
        LSET us2$ = MKD$(us2)
        PUT #1, 4
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 8 AS M20$, 8 AS M5$, 8 AS aa20$, 8 AS aa5$, 8 AS uu20$, 8 AS uu5$, 8 AS rho20rho10$, 8 AS rho5rho2$, 8 AS us20$
        LSET M20$ = MKD$(M20#)
        LSET M5$ = MKD$(M5#)
        LSET aa20$ = MKD$(aa20)
        LSET aa5$ = MKD$(aa5)
        LSET uu20$ = MKD$(uu20)
        LSET uu5$ = MKD$(uu5)
        LSET rho20rho10$ = MKD$(rho20rho10)
        LSET rho5rho2$ = MKD$(rho5rho2)
        LSET us20$ = MKD$(us20)
        PUT #1, 5
      ' Setup file for point data
        CLOSE #1
        OPEN F$ AS #1 LEN = 76
        FIELD #1, 16 AS GU$, 2 AS i$, 2 AS j$, 8 AS x$, 8 AS t$, 8 AS u$, 8 AS a$, 8 AS p$, 8 AS TT$, 8 AS rho$
        RETURN

OldFile:
      ' Get status information
        FIELD #1, 2 AS max$, 8 AS magfactor$, 2 AS ScrollX$, 2 AS ScrollT$, 8 AS xg$, 8 AS tg$, 8 AS xgblob$, 8 AS tgblob$, 8 AS tg2$, 2 AS iexpand$,
            8 AS gammablob$
        GET #1, 1
        max = CVI(max$)
        magfactor# = CVD(magfactor$)
        iScrollX = CVI(ScrollX$)
        iScrollT = CVI(ScrollT$)
        xg = CVD(xg$)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
    tg = CVD(tg$)
    agblob = CVD(agblob$)
    tgblob = CVD(tgblob$)
    tg2 = CVD(tg2$)
    lexpend = CVI(lexpend$)
    gammablob = CVD(gammablob$)
' Get initial conditions
    CLOSE #1
    OPEN F$ AS #1 LEN = 76
    FIELD #1, 8 AS P4P1$, 8 AS p1p10$, 8 AS gamma4$, 8 AS gamma1$, 8 AS L1L2$  8 AS R4$, 8 AS R1$, 8 AS T4T1$
    GET #1, 2
    P4P1 = CVD(P4P1$)
    p1p10 = CVD(p1p10$)
    gamma4 = CVD(gamma4$)
    gamma1 = CVD(gamma1$)
    L1L2# = CVD(L1L2$)
    R4 = CVD(R4$)
    R1 = CVD(R1$)
    T4T1 = CVD(T4T1$)
' Get derived variables
    CLOSE #1
    OPEN F$ AS #1 LEN = 76
    FIELD #1, 8 AS p2p1$, 8 AS p3p4$, 8 AS T3T4$, 8 AS T2T1$, 8 AS M2$, 8 AS M3$, 8 AS aa2$  8 AS aa3$, 8 AS uu2$
    GET #1, 3
    p2p1 = CVD(p2p1$)
    p3p4 = CVD(p3p4$)
    T3T4 = CVD(T3T4$)
    T2T1 = CVD(T2T1$)
    M2# = CVD(M2$)
    M3# = CVD(M3$)
    aa2 = CVD(aa2$)
    aa3 = CVD(aa3$)
    uu2 = CVD(uu2$)
    CLSE #1
    OPEN F$ AS #1 LEN = 76
    FIELD #1, 8 AS uu3$, 8 AS rho2rho1$, 8 AS rho3rho4$, 8 AS lm$, 8 AS p20p10$, 8 AS p5p2$  8 AS T5T2$, 8 AS T20T10$, 8 AS ue2$
    GET #1, 4
    uu3 = CVD(uu3$)
    rho2rho1 = CVD(rho2rho1$)
    rho3rho4 = CVD(rho3rho4$)
    lm# = CVD(lm$)
    p20p10 = CVD(p20p10$)
    p5p2 = CVD(p5p2$)
    T5T2 = CVD(T5T2$)
    T20T10 = CVD(T20T10$)
    ue2 = CVD(ue2$)
    CLOSE #1
    OPEN F$ AS #1 LEN = 76
    FIELD #1, 8 AS M20$, 8 AS M5$, 8 AS aa20$, 8 AS aa5$, 8 AS uu20$, 8 AS uu5$, 8 AS rho20rho10$  8 AS rho5rho2$, 8 AS ue20$
    GET #1, 5
    M20# = CVD(M20$)
    M5# = CVD(M5$)
    aa20 = CVD(aa20$)
    aa5 = CVD(aa5$)
    uu20 = CVD(uu20$)
    uu5 = CVD(uu5$)
    rho20rho10 = CVD(rho20rho10$)
    rho5rho2 = CVD(rho5rho2$)
    ue20 = CVD(ue20$)
  ' Setup file for point data
    CLOSE #1
    OPEN F$ AS #1 LEN = 76
    FIELD #1, 16 AS gg$, 2 AS i$, 2 AS j$, 8 AS x$, 8 AS t$, 8 AS u$, 8 AS a$, 8 AS p$, 8 AS TT$, 8 AS rho$
    RETURN

SavePoint:
        max = max + 1
        LSET gg$ = Grid$
        LSET i$ = MKI$(i)
        LSET j$ = MKI$(j)
        LSET x$ = MKD$(x3)
        LSET t$ = MKD$(t3)
        LSET u$ = MKD$(u3)
        LSET a$ = MKD$(a3)
        LSET p$ = MKD$(p3)
        LSET TT$ = MKD$(TT3)
        LSET rho$ = MKD$(rho3)
        PUT #1, max
        RETURN

FindPoint:
        GET #1, ii
        i = CVI(i$)
        j = CVI(j$)
        x = CVD(x$)
        t = CVD(t$)
        u = CVD(u$)
        a = CVD(a$)
        p = CVD(p$)
        TT = CVD(TT$)
        rho = CVD(rho$)
        RETURN

FindPoint1:
        GET #1, i
        i1 = CVI(i$)
        j1 = CVI(j$)
        x1 = CVD(x$)
        t1 = CVD(t$)
        u1 = CVD(u$)
        a1 = CVD(a$)
        p1 = CVD(p$)
        TT1 = CVD(TT$)
        rho1 = CVD(rho$)
        RETURN

FindPoint2:
        GET #1, j
        i2 = CVI(i$)
        j2 = CVI(j$)
        x2 = CVD(x$)
        t2 = CVD(t$)
        u2 = CVD(u$)
        a2 = CVD(a$)
        p2 = CVD(p$)
        TT2 = CVD(TT$)
        rho2 = CVD(rho$)
        RETURN

FindPoint3:
        GET #1, ii3
        x3 = CVD(x$)
        t3 = CVD(t$)
        u3 = CVD(u$)
        a3 = CVD(a$)
        p3 = CVD(p$)
        TT3 = CVD(TT$)
        rho3 = CVD(rho$)
        RETURN

FindPoint4t:
        GET #1, i2
        i4 = CVI(i$)
        x4 = CVD(x$)
        t4 = CVD(t$)
        u4 = CVD(u$)
        a4t = CVD(a$)
        p4 = CVD(p$)
        TT4t = CVD(TT$)
        rho4t = CVD(rho$)
        RETURN

FindPoint4d:
        GET #1, i4
        x4 = CVD(x$)
        t4 = CVD(t$)
        u4 = CVD(u$)
        a4d = CVD(a$)
        p4 = CVD(p$)
        TT4d = CVD(TT$)
        rho4d = CVD(rho$)
        RETURN

FindPoint5:
        GET #1, ii5
        i5 = CVI(i$)
        j5 = CVI(j$)
        x5 = CVD(x$)
        t5 = CVD(t$)
        u5 = CVD(u$)
        a5 = CVD(a$)
        p5 = CVD(p$)
        TT5 = CVD(TT$)
        rho5 = CVD(rho$)
        RETURN

FindPoint6:
        GET #1, ii6
        x6 = CVD(x$)
        t6 = CVD(t$)
        u6 = CVD(u$)
        a6 = CVD(a$)
        p6 = CVD(p$)
        TT6 = CVD(TT$)
        rho6 = CVD(rho$)
        RETURN

Renumber:
        IF i1 = ii2 OR i2 = ii1 THEN GOTO Renumberi
    Renumberj:
        IF j1 = ii2 THEN GET#1, ii1 : j = max + 1 : LSET j$ = MKI$(j) : PUT #1, ii1 : j = ii2 : jPoint(ii1) = max + 1 ELSE GET #1, ii2 : j = max + 1 :
          LSET j$ = MKI$(j) : PUT #1, ii2 : j = ii1 : jPoint(ii2) = max + 1
        Grid$ = "Interiorj" : i = 0
        GOSUB SavePoint
        PointType(max) = 4
        RETURN
    Renumberi:
        IF i1 = ii2 THEN GET#1, ii1 : i = max + 1 : LSET i$ = MKI$(i) : PUT #1, ii1 : i = ii2 : iPoint(ii1) = max + 1 ELSE GET #1, ii2 : i = max + 1
          LSET i$ = MKI$(i) : PUT #1, ii2 : i = ii1 : iPoint(ii2) = max + 1
        Grid$ = "Interiori" : j = 0
        GOSUB SavePoint
        PointType(max) = 3
        RETURN

'-----------------------------------------------
' CALCULATIONS

NewCalcs:
    ' Calculate shock tube and acceleration tube properties
        slower1 = 1!
        IF ilangleyflag = 0 THEN supper1 = .9*(SQR(gamma1*(gamma1 + 1!)*(gamma4 + 1!)*gamma4*R4*T4T1/(gamma1*R1))/(gamma4 - 1!))^2 ELSE supper1 =
          .9*(SQR(gamma1*(gamma1 + 1!)*2#*gamma4*R4*T4T1/(gamma1*R1))/(gamma4 - 1!))^2
        iflag = 0
        IF ilangleyflag = 0 THEN yy1 = slower1*(SQR((gamma4 + 1!)/2!) - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(slower1 -
          1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(slower1 - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        IF ilangleyflag = 1 THEN yy1 = slower1*(1# - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(slower1 - 1#))/(SQR((2#*gamma1)*(2#*gamma1 +
          (gamma1 + 1#)*(slower1 - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        IF ilangleyflag = 0 THEN yy2 = supper1*(SQR((gamma4 + 1!)/2!) - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(supper1 -
          1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(supper1 - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        IF ilangleyflag = 1 THEN yy2 = supper1*(1# - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(supper1 - 1#))/(SQR((2#*gamma1)*(2#*gamma1 +
          (gamma1 + 1#)*(supper1 - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        IF yy1*yy2 > 0! THEN BEEP : BEEP : STOP
        CALL zeroin(slower1, supper1, tol, iflag, s) : p2p1 = s
        p3p4 = p2p1/P4P1
        T3T4 = (p3p4)^((gamma4 - 1#)/gamma4)
        T2T1 = (1# + (gamma1 - 1#)*p2p1/(gamma1 + 1#))/(1# + (gamma1 - 1#)/(p2p1*(gamma1 + 1#)))
        M2# = (1#/gamma1)*(p2p1 - 1#)/SQR(p2p1*((gamma1 + 1#)/(2#*gamma1)) + (gamma1 - 1#)/(2#*gamma1)*p2p1))
        IF ilangleyflag = 0 THEN M3# = (2#/(gamma4 - 1#))*((1#/p3p4)^((gamma4 - 1#)/(2#*gamma4))*SQR((gamma4 + 1#)/2#) - 1#)
        IF ilangleyflag = 1 THEN M3# = (2#/(gamma4 - 1#))*((1#/p3p4)^((gamma4 - 1#)/(2#*gamma4)) - 1#)
        aa2 = SQR(T2T1*gamma1*R1/(T4T1*gamma4*R4))
        uu2 = M2#*aa2
        aa3 = SQR(T3T4)
        uu3 = M3#*aa3
        rho2rho1 = p2p1/T2T1
        rho3rho4 = p3p4/T3T4
        us2 = SQR(((gamma1 - 1#)/(2#*gamma1) + ((gamma1 + 1#)/(2#*gamma1))*p2p1)*gamma1*R1/(T4T1*gamma4*R4))
        slower2 = 1!
        supper2 = .9*(SQR(2!*gamma1*(gamma1 + 1!)*T2T1)*(1! + (gamma1 - 1!)*M2#/2!)/(gamma1 - 1!))^2
        iflag = 1
        yy1 = slower2*(1# + (gamma1 - 1!)*M2#/2! - ((gamma1 - 1#)*SQR(1#/T2T1)*(slower2 - 1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(slower2 -
          1#)))))^(-2#*gamma1/(gamma1 - 1#)) - p2p1*p1p10
        yy2 = supper2*(1# + (gamma1 - 1!)*M2#/2! - ((gamma1 - 1#)*SQR(1#/T2T1)*(supper2 - 1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(supper2 -
          1#)))))^(-2#*gamma1/(gamma1 - 1#)) - p2p1*p1p10
        IF yy1*yy2 > 0! THEN BEEP : BEEP : STOP
        CALL zeroin(slower2, supper2, tol, iflag, s) : p20p10 = s
        p5p2 = p20p10/(p1p10*p2p1)
        T5T2 = (p5p2)^((gamma1 - 1#)/gamma1)
        T20T10 = (1# + (gamma1 - 1#)*p20p10/(gamma1 + 1#))/(1# + (gamma1 - 1#)/(p20p10*(gamma1 + 1#)))
        M20# = (1#/gamma1)*(p20p10 - 1#)/SQR(p20p10*((gamma1 + 1#)/(2#*gamma1)) + (gamma1 - 1#)*p20p10/(2#*gamma1)))
        M5# = (2#/(gamma1 - 1#))*((1#/p5p2)^((gamma1 - 1#)/(2#*gamma1))*(1# + (gamma1 - 1#)*M2#/2#) - 1#)
        aa20 = SQR(T20T10*gamma1*R1/(T4T1*gamma4*R4))
        us20 = M20#*aa20
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        aa5 = SQR(T5T2*T2T1*gamma1*R1/(T4T1*gamma4*R4))
        uu5 = M5#*aa5
        rho20rho10 = p20p10/T20T10
        rho5rho2 = p5p2/T5T2
        us20 = SQR(((gamma1 - 1#)/(2#*gamma1)) + ((gamma1 + 1#)/(2#*gamma1#))*p20p10)*gamma1*R1/(T4T1*gamma4*R4))
        RETURN

Interior:
        IF igammaflag = 1 THEN gamma = gamma4 : R = R4 ELSE gamma = gamma1 : R = R1
        u3 = .5#*(u1 + u2) + (a1 - a2)/(gamma - 1#)
        a3 = (gamma - 1#)*(u1 - u2)/4# + (a1 + a2)/2#
        RETURN

Expansion:
        u2 = 2#/((gamma1 + 1#)*TAN(expangle)) + (gamma1 - 1#)*uu2/(gamma1 + 1#) + 2#*aa2/(gamma1 + 1#)
        a2 = (gamma1 - 1#)*(uu2 - u2)/2# + aa2
        p2 = p1*(a2/a1)^(2#*gamma1/(gamma1 - 1#))
        rho2 = rho1*(a2/a1)^(2#/(gamma1 - 1#))
        igammaflag = 0
        RETURN

Contact:
      Calculate contact surface point position
        x3 = (t4 - t2 + x2*(u2 - a2 + u3 - a3t)/(2#*(u2 - a2)*(u3 - a3t)) - x4*(u3 + u4)/(2#*u3*u4))/((u2 - a2 + u3 - a3t)/(2#*(u2 - a2)*(u3 - a3t
            (u3 + u4)/(2#*u3*u4))
        t3 = (x4 - x2 + t2*2#*(u2 - a2)*(u3 - a3t)/(u2 - a2 + u3 - a3t) - t4*2#*u3*u4/(u3 + u4))/(2#*(u2 - a2)*(u3 - a3t)/(u2 - a2 + u3 - a3t) - 
            2#*u3*u4/(u3 + u4))
    ' Calculate driver gas point position
        x1 = (t3 - t4 + x4*(u4 - a4d + u5 - a5)/(2#*(u4 - a4d)*(u5 - a5)) - x3*(u1 + a1 + u3 + a3d)/(2#*(u1 + a1)*(u3 + a3d)))/((u4 - a4d + u5 -
            a5)/(2#*(u4 - a4d)*(u5 - a5)) - (u1 + a1 + u3 + a3d)/(2#*(u1 + a1)*(u3 + a3d)))
        t1 = (x3 - x4 + t4*2#*(u4 - a4d)*(u5 - a5)/(u4 - a4d + u5 - a5) - t3*2#*(u1 + a1)*(u3 + a3d)/(u1 + a1 + u3 + a3d))/(2#*(u4 - a4d)*(u5 -
            a5)/(u4 - a4d + u5 - a5) - 2#*(u1 + a1)*(u3 + a3d)/(u1 + a1 + u3 + a3d))
    ' Calculate driver gas point properties
        c = SQR((t4 - t1)^2 + (x4 - x1)^2)
        b = SQR((t5 - t1)^2 + (x5 - x1)^2)
        cb = SQR((t5 - t4)^2 + (x5 - x4)^2)
        u1 = (b*u4 + c*u5)/cb : a1 = (b*a4d + c*a5)/cb
    ' Calculate other point properties
        p1 = (b*p4 + c*p5)/cb
        rho1 = (b*rho4d + c*rho5)/cb
        p3 = p4*(a3t/a4t)^(2#*gamma1/(gamma1 - 1#))
        rho3d = p3/a3d^2
        rho3t = gamma1*p3/((a3t^2)*gamma4)
    ' Calculate Contact surface point properties
        u3 = ((rho1*a1 + rho3d*a3d)*u1/2# + (rho2*a2 + rho3t*a3t)*u2/2# + p1 - p2)/((rho1*a1 + rho3d*a3d)/2# + (rho2*a2 + rho3t*a3t)/2#)
        p3 = (2#*p1/(rho1*a1 + rho3d*a3d) + 2#*p2/(rho2*a2 + rho3t*a3t) + u1 - u2)/(2#/(rho1*a1 + rho3d*a3d) + 2#/(rho2*a2 + rho3t*a3t))
        a3t = a4t*(p3/p4)^((gamma1 - 1#)/(2#*gamma1))
        a3d = a4d*(p3/p4)^((gamma4 - 1#)/(2#*gamma4))
        RETURN

Calculations:
        IF igammaflag = 1 THEN temp = 1! : dens = 1! ELSE temp = (gamma4*R4)/(gamma1*R1) : dens = R4/R1
        IF igammaflag = 1 THEN p3 = p1*(a3/a1)^(2#*gamma4/(gamma4 - 1#)) ELSE p3 = p1*(a3/a1)^(2#*gamma1/(gamma1 - 1#))
        TT3 = (a3^2)*temp
        rho3 = (p3/TT3)*dens
'   Store results of calculations and store point type
        GOSUB SavePoint
'   Draw output mesh
        GOSUB DrawPoint
'   Draw characteristic lines
        GOSUB DrawMach
        RETURN

Average:
        aaa = SQR((t2 - tttd)^2 + (x2 - xxxd)^2)
        bbb = SQR((tttd - t1)^2 + (xxxd - x1)^2)
        u3 = (aaa*u1 + bbb*u2)/(aaa + bbb)
        a3 = (aaa*a1 + bbb*a2)/(aaa + bbb)
        p3 = (aaa*p1 + bbb*p2)/(aaa + bbb)
        TT3 = (aaa*TT1 + bbb*TT2)/(aaa + bbb)
        rho3 = (aaa*rho1 + bbb*rho2)/(aaa + bbb)
        RETURN

PitotCalcs:
        i = ii1 : GOSUB FindPoint1
        j = ii2 : GOSUB FindPoint2
        x3 = 1#
        t3 = (t2 - t1)*(x3 - x1)/(x2 - x1) + t1
        IF ipitotpoint = 2 OR ipitotpoint = 4 THEN GOTO PitotCalcs2
        aaa = SQR((t2 - t3)^2 + (x2 - x3)^2)
        bbb = SQR((t3 - t1)^2 + (x3 - x1)^2)
        u3 = (aaa*u1 + bbb*u2)/(aaa + bbb)
        a3 = (aaa*a1 + bbb*a2)/(aaa + bbb)
        p3 = (aaa*p1 + bbb*p2)/(aaa + bbb) : MOC# = .8#*u3/a3
        IF ipitotpoint = 1 THEN gamma = gamma1
        IF ipitotpoint = 3 THEN gamma = gamma4
        pitot = .8#*p3*((gamma + 1!)*MOC#^2/2!)^(gamma/(gamma - 1!))/(2!*gamma*MOC#^2/(gamma + 1!) - (gamma - 1!)/(gamma + 1!))^(1! /(gamma - 1
        Grid$ = "Pitot" : TT3 = pitot
    PitotCalcs2:
        i = max + 2 : j = max + 2
        IF ipitotpoint = 2 THEN Grid$ = "ContactTime"
        IF ipitotpoint = 4 THEN Grid$ = "BlobTime"
        GOSUB SavePoint
        iPoint(max) = max + 1 : jPoint(max) = max + 1
        IF ipitotpoint <> 2 AND ipitotpoint <> 4 THEN PointLocx(max) = x3 : PointLoct(max) = t3 : PointType(max) = 7
        IF ipitotpoint = 2 THEN PointType(max) = 8
        IF ipitotpoint = 4 THEN PointType(max) = 9
        WINDOW 4 : BUTTON 5, 1
        RETURN

Mirels:
        ms# = us2*SQR(gamma4*R4*T4T1/(gamma1*R1))
      ' ms2# = us20*SQR(gamma4*R4*T4T1/(gamma1*R1))
        GOSUB GetBeta
        GOSUB GetBeta2
        lm# = (dL2/(4#*beta))^(5#/4#)*(ms#*206100000#*354554#/(P4P1*1010#))^(1#/4#)*(rho2rho1)/((rho2rho1 - 1#)*p2p1)
        lm2# = (dL2/(4#*beta2))^2*ms2#*206100000#*354554#*rho20rho10/(P4P1*101#*p20p10*(rho20rho10 - 1#)*p1p10)
        RETURN

GetBeta:
        IF ms# < 8# THEN beta = -.00315#*(ms# - 6#) + .0283# : GOTO BetaEnd
        IF ms# < 10# THEN beta = -.0016#*(ms# - 8#) + .022# : GOTO BetaEnd
        IF ms# < 12# THEN beta = -.00155#*(ms# - 10#) + .0188# : GOTO BetaEnd
        IF ms# < 14# THEN beta = -.0014#*(ms# - 12#) + .0157# : GOTO BetaEnd
        IF ms# < 16# THEN beta = -.00065#*(ms# - 14#) + .0129# : GOTO BetaEnd
        IF ms# < 18# THEN beta = -5.999999999999999D-04*(ms# - 16#) + .0116# : GOTO BetaEnd
        beta = -.0005#*(ms# - 18#) + .0104#
    BetaEnd:
        beta = beta*((rho2rho1^2 + 1.25#*rho2rho1 - .88)/(rho2rho1*(rho2rho1 - 1#)))^(4#/5#)
        RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
GetBeta2:
        IF ms2# < 60 THEN beta2 = -.095#*(ms2# - 40) + 1.66# : GOTO Beta2End
        IF ms2# < 80 THEN beta2 = -8.500000000000001D-03*(ms2# - 60) + 1.47# : GOTO Beta2End
        IF ms2# < 100 THEN beta2 = -.0555#*(ms2# - 80) + 1.3# : GOTO Beta2End
        IF ms2# < 120 THEN beta2 = -.045#*(ms2# - 100) + 1.19# : GOTO Beta2End
        beta2 = -.03#*(ms2# - 120) + 1.1#
    Beta2End:
        RETURN

BlobEscape:
        x1 = x3 : t1 = t3
        x3 = 1# : x2 = x3
        t3 = (1# - x1)/W3 + t1 : t2 = t3
        u3 = uu5
        a3 = W3
        p3 = rhoe
        TT3 = T5T2*T2T1/T4T1
        rho3 = p5p2*p2p1*R4/(P4P1*TT3*R1)
        i = max : j = max
        GOSUB SavePoint
        PointLocx(max) = x3 : PointLoct(max) = t3
        PointType(max) = 6 : iPoint(max) = i : jPoint(max) = j
        GOSUB SavePoint
        GOSUB DrawPoint
        GOSUB DrawBlob
        RETURN

'------------------------------------------------
' GRAPHICS ROUTINES

Scales:
        PENSIZE 1, 1 : TEXTFONT 4 : TEXTSIZE 9 : TEXTFACE 32
        MOVETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 10 : LINETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 0
        xs = 490#*L1L2#/(1# + L1L2#) + 14 + iScrollX
        MOVETO xs + 1, 8 : PRINT " 0"
        WHILE xs < 490#*(magfactor# + L1L2#)/(1# + L1L2#) - 40 + iScrollX
                xs = xs + 50
                        'IF xs > 508 THEN GOTO ScalesEnd1
                MOVETO xs, 10 : LINETO xs, 5
                MOVETO xs + 1, 8
                x% = CINT(((xs - 14 - iScrollX)*(1# + L1L2#)/490# - L1L2#)*1000#/magfactor#)
                PRINT CSNG(x%)/1000;
        WEND
    ScalesEnd1:
        xs = 490#*L1L2#/(1# + L1L2#) + 14 + iScrollX
        WHILE xs >= 490#*L1L2#*(1# - magfactor#)/(1# + L1L2#) + 70 + iScrollX
                xs = xs - 50
                        'IF xs < 0 THEN GOTO ScalesEnd2
                MOVETO xs, 10 : LINETO xs, 5
                MOVETO xs + 3, 8
                x% = CINT(((xs - 14 - iScrollX)*(1# + L1L2#)/490# - L1L2#)*1000#/magfactor#)
                PRINT CSNG(x%)/1000;
        WEND
    ScalesEnd2:
        ts = 314 + iScrollT
        MOVETO 10, ts : LINETO 0, ts : MOVETO -2, ts - 3 : PRINT " 0";
        WHILE ts > 85
                ts = ts - 50
                                  'IF ts > 318 THEN GOTO ScalesEnd
                MOVETO 10, ts : LINETO 5, ts
                tt = CSNG((314 + iScrollT - ts)*(1# + L1L2#)/(490#*magfactor#*tstretch))
                TTs$ = STR$(CINT(tt*100))
                as$ = " " + MID$(TTs$, 2, 1)
                b$ = " " + MID$(TTs$, 3, 1)
                MOVETO 3, ts - 21 : PRINT ".";
                IF LEN(TTs$) = 4 THEN c$ = " " + MID$(TTs$, 4, 1) : MOVETO -2, ts - 25 : PRINT as$;
                IF LEN(TTs$) = 2 THEN MOVETO -2, ts - 12 : PRINT " 0"; : MOVETO -2, ts - 3 : PRINT as$; : GOTO ScalesEnd
                MOVETO -2, ts - 12 : IF LEN(TTs$) = 4 THEN PRINT b$; ELSE PRINT as$;
                MOVETO -3, ts - 3 : IF LEN(TTs$) = 4 THEN PRINT c$; ELSE PRINT b$;
    ScalesEnd:
        WEND
        MOVETO 490#*L1L2#*(1# - magfactor#)/(1# + L1L2#) + 14 + iScrollX, 10 : LINETO 490#*(magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 10
           MOVETO 10, 10 : LINETO 10, 314 + iScrollT
        TEXTFACE 1 : TEXTFONT 1 : TEXTSIZE 12
        RETURN

Tube:
        PENSIZE 2, 2
        tstretch = (314 - L1L2#*490#*((uu2 - aa2)*(1# - wu2/us2)/aa2 + 1#)/(uu2*(1# + L1L2#)))/150#
        MOVETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        LINETO (L1L2# + magfactor#)*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        MOVETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        LINETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 10
        MOVETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollT
        LINETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX,10
        MOVETO (L1L2# + magfactor#)*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        LINETO (L1L2# + magfactor#)*490#/(1# + L1L2#) + 14# + iScrollX, 10
        MOVETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        LINETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollT - L1L2#*490#*magfactor#*tstretch/((1# + L1L2#)*us2)
        LINETO (L1L2# + magfactor#)*490#/(1# + L1L2#) + 14# + iScrollX, 314 + iScrollT - (1#/us2# - L1L2#/us2)*490#*magfactor#*tstretch/(1# + L1L2#)
        xi = L1L2#*490#/(1# + L1L2#) + 14 + iScrollX
        ti = 314 + iScrollT - L1L2#*490#*magfactor#*tstretch/((1# + L1L2#)*us2)
        udash = 1#/((tg2 - L1L2#/us2)*tstretch)
        xend = (L1L2# + magfactor#)*490#/(1# + L1L2#) + 14# + iScrollX
        tend = 314 + iScrollT - tg2*490#*magfactor#*tstretch/(1# + L1L2#)
        GOSUB DashLine
        xi = (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX
        ti = 314# + iScrollT
        udash = (xg + L1L2#)/(tg*tstretch)
        xend = 490#*(xg*magfactor# + L1L2#)/(1# + L1L2#) + 14# + iScrollX
        tend = 314# + iScrollT - 490#*tg*magfactor#*tstretch/(1# + L1L2#)
        GOSUB DashLine
        MOVETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        CALL PENPAT(VARPTR(idash(0)))
        IF xgblob > 0# AND tgblob > 0# THEN LINETO 490#*(xgblob*magfactor# + L1L2#)/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT -
           tgblob*magfactor#*tstretch*490#/(1# + L1L2#)
        CALL PENNORMAL
        PENSIZE 1, 1
        MOVETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollT - L1L2#*490#*magfactor#*tstretch/((1# + L1L2#)*ms2)
        LINETO xend, tend
        t = (314 + iScrollT - 10)*(1# + L1L2#)/(490#*magfactor#*tstretch)
        xend = xg
        tend = tg
        LINETO 490#*(((t - tend)*(uu3 - aa3) + xend)*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 10
        MOVETO (1# - magfactor#)*L1L2#*490#/(1# + L1L2#) + 14# + iScrollX, 314# + iScrollT
        LINETO 490#*((t*(uu3 - aa3) - L1L2#)*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 10
        RETURN

DrawPoint:
        PENSIZE 1, 1
        CIRCLE (490#*(x3*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollT - t3*magfactor#*tstretch*490/(1# + L1L2#)), 3
        RETURN

DrawContact:
        PENSIZE 2, 2
        xi = 490#*(x4*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX
        ti = 314 + iScrollY - t4*magfactor#*tstretch*490#/(1# + L1L2#)
        xend = 490#*(x3*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX
        tend = 314 + iScrollY - t3*magfactor#*tstretch*490#/(1# + L1L2#)
        udash = -(xend - xi)/(tend - ti)
        GOSUB DashLine
        PENSIZE 1, 1
        RETURN

DrawMach:
        PENSIZE 1, 1
        MOVETO 490#*(x3*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY - t3*490#*magfactor#*tstretch/(1# + L1L2#)
        IF Grid$ <> "Interiorj" THEN LINETO 490#*(x1*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY -
            t1*490#*magfactor#*tstretch/(1# + L1L2#)
        IF Grid$ = "Interiori" THEN GOTO DrawMachEnd
        MOVETO 490#*(x3*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY - t3*490#*magfactor#*tstretch/(1# + L1L2#)
        IF Grid$ <> "Expansion" THEN LINETO 490#*(x2*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY -
            t2*490#*magfactor#*tstretch/(1# + L1L2#)
        IF Grid$ = "Expansion" AND j <> -2 AND jj <> -2 THEN LINETO L1L2#*490#/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY -
            L1L2#*490#*magfactor#*tstretch/((1# + L1L2#)*ue2)
    DrawMachEnd:
        RETURN

Grid:
        CALL PENPAT(VARPTR(igrey(0))) : CALL PENSIZE(1, 1) : CALL PENMODE(9)
        xs = 490#*L1L2#/(1# + L1L2#) + 14 + iScrollX
        WHILE xs < 454
                xs = xs + 50
                MOVETO xs, 10 : LINETO xs, 318
        WEND
        xs = 490#*L1L2#/(1# + L1L2#) + 14 + iScrollX
        WHILE xs >= 64
                xs = xs - 50
                MOVETO xs, 10 : LINETO xs, 318
        WEND
        ts = 314 + iScrollY
        WHILE ts > 85
                ts = ts - 50
                MOVETO 10, ts : LINETO 508, ts
        WEND
        CALL PENNORMAL
        RETURN

DashLine:
        'IF xi < 14 AND ti <= 314 THEN ti = -(14 - xi)/udash + ti : xi = 14 : GOTO DashStart
        'IF ti > 314 AND xi >= 14 THEN xi = -udash*(314 - ti) + xi : ti = 314 : GOTO DashStart
        'IF xi < 14 AND ti > 314 THEN ti2 = -(14 - xi)/udash + ti : IF ti2 <= 314 THEN xi = 14 : ti = ti2 ELSE xi = -udash*(314 - ti) + xi : ti = 3 4
ELSE REM
    DashStart:
        WHILE xi < xend AND ti > tend 'AND xi < 504 AND ti > 10
                MOVETO xi, ti
                xi1 = 10#*udash/SQR(udash^2 + 1#) + xi
                ti1 = -(xi1 - xi)/udash + ti
                IF xi1 >= xend OR ti1 <= tend THEN xi1 = xend : ti1 = tend
                LINETO xi1, ti1
                xi = 5#*udash/SQR(udash^2 + 1#) + xi1
                ti = -(xi - xi1)/udash + ti1
        WEND
        RETURN

ReDraw:
        PICTURE OFF : CALL HIDEPEN : PICTURE ON : CALL SHOWPEN
    Redraw2:
        IF igridflag = 1 AND inumberflag = 0 THEN GOSUB Grid
        IF inumberflag = 1 AND igridflag = 0 THEN GOSUB ShowNumber
        IF inumberflag = 1 AND igridflag = 1 THEN GOSUB Grid : GOSUB ShowNumber
        GOSUB Tube
        ii = -1
        WHILE TRUE% = 1
                ii = ii + 1
                GET #1, ii + 6
                IF EOF(1) OR ii > max - 6 THEN GOTO Redraw5
                i = CVI(i$) : j = CVI(j$) : jj = 0
                x3 = CVD(x$)
                t3 = CVD(t$)
                Grid$ = gg$
                iblank = INSTR(Grid$, " ") : Grid$ = LEFT$(Grid$, iblank - 1)
                GOSUB GetLocations
                IF Grid$<>"Interior" AND Grid$<>"Expansion" AND Grid$<>"Contactt" AND Grid$<>"Contactd" AND Grid$<>"Interiori" AND Grid$<>"Interiorj"
                        AND Grid$<>"Blob" AND Grid$<>"FirstPoint" THEN GOTO ReDraw4
                IF Grid$ = "Interior" AND i <> 0 THEN GOSUB FindPoint1 : GOSUB FindPoint2 : GOTO ReDraw3
                IF Grid$ = "Interiori" THEN GOSUB FindPoint1 : GOTO ReDraw3
                IF Grid$ = "Interiorj" THEN GOSUB FindPoint2 : GOTO ReDraw3
                IF Grid$ = "Expansion" THEN GOSUB FindPoint1 : GOTO ReDraw3
                IF Grid$ = "Contactt" THEN GOSUB FindPoint2 : GOSUB FindPoint4t : GOSUB FindPoint1 : i = ii : GOSUB FindPoint1 : GOTO ReDraw3
                IF Grid$ = "Blob"  AND i <> 0 THEN GOSUB FindPoint1 : GOTO ReDraw3
                IF Grid$ = "FirstPoint" AND icircleflag = 1 THEN GOSUB DrawPoint : GOTO ReDraw4
            ReDraw3:
                IF icircleflag = 1 THEN GOSUB DrawPoint
                IF Grid$ = "Contactt" THEN GOSUB DrawContact
                IF Grid$ = "Blob" AND i <> 0 THEN x2 = x3 : t2 = t3 : GOSUB DrawBlob
                IF i = 0 AND j = 0 THEN GOTO ReDraw4
                IF Grid$ <> "Blob" THEN GOSUB DrawMach
            ReDraw4:
        WEND
    Redraw5:
        LINE (0, 0) - (508, 10) , 30, bf
        LINE (0, 10) - (10, 318), 30, bf
        GOSUB Scales
        RETURN

GetLocations:
        IF UCASE$(Grid$) = "INTERIOR" THEN PointType(ii + 6) = 0
        IF UCASE$(Grid$) = "EXPANSION" THEN PointType(ii + 6) = 1
        IF UCASE$(Grid$) = "CONTACTT" THEN PointType(ii + 6) = 2
        IF UCASE$(Grid$) = "CONTACTD" THEN PointType(ii + 6) = 11
        IF UCASE$(Grid$) = "INTERIORI" THEN PointType(ii + 6) = 3
        IF UCASE$(Grid$) = "INTERIORJ" THEN PointType(ii + 6) = 4
        IF UCASE$(Grid$) = "BLANK" THEN PointType(ii + 6) = 5
        IF UCASE$(Grid$) = "BLOB" THEN PointType(ii + 6) = 6
        IF UCASE$(Grid$) = "FIRSTPOINT" THEN PointType(ii + 6) = 10
        IF UCASE$(Grid$) = "PITOT" THEN PointType(ii + 6) = 7
        IF UCASE$(Grid$) = "CONTACTTIME" THEN PointType(ii + 6) = 8
        IF UCASE$(Grid$) = "BLOBTIME" THEN PointType(ii + 6) = 9
        PointLocx(ii + 6) = x3
        PointLoct(ii + 6) = t3
        iPoint(ii + 6) = i : jPoint(ii + 6) = j
        RETURN

RePlot:
        PICTURE OFF : CALL HIDEPEN : PICTURE ON : CALL SHOWPEN
    RePlot2:
        IF igridflag = 1 AND inumberflag = 0 THEN GOSUB Grid
        IF inumberflag = 1 AND igridflag = 0 THEN GOSUB ShowNumber
        IF inumberflag = 1 AND igridflag = 1 THEN GOSUB Grid : GOSUB ShowNumber
        GOSUB Tube
        FOR i = 6 TO max
                x3 = PointLocx(i)
                t3 = PointLoct(i)
                ii = iPoint(i) : jj = jPoint(i) : j = 0
                IF PointType(i) = 0 THEN Grid$ = "Interior" : IF ii <> 0 THEN x1 = PointLocx(iPoint(i)) : t1 = PointLoct(iPoint(i)) : x2 =
                        PointLocx(jPoint(i)) : t2 = PointLoct(jPoint(i)) : GOTO RePlot3 ELSE GOTO RePlot3 ELSE REM
                IF PointType(i) = 1 THEN Grid$ = "Expansion" : x1 = PointLocx(iPoint(i)) : t1 = PointLoct(iPoint(i)) : GOTO RePlot3
                IF PointType(i)=2 THEN Grid$ ="Contact1":x2=PointLocx(jPoint(i)):t2=PointLoct(jPoint(i)):x4=PointLocx(iPoint(jPoint(i)))
                        :t4=PointLoct(iPoint(jPoint(i))):x1=PointLocx(iPoint(iPoint(i))):t1=PointLoct(iPoint(iPoint(i))):GOTO RePlot3
                IF PointType(i) = 11 THEN Grid$ = "Contact2" : GOTO RePlot4
                IF PointType(i) = 3 THEN Grid$ = "Interiori" : x1 = PointLocx(iPoint(i)) : t1 = PointLoct(iPoint(i)) : GOTO RePlot3
                IF PointType(i) = 4 THEN Grid$ = "Interiorj" : x2 = PointLocx(jPoint(i)) : t2 = PointLoct(jPoint(i)) : GOTO RePlot3
                IF PointType(i) = 6 THEN Grid$ = "Blob" : IF ii <> 0 THEN x1 = PointLocx(iPoint(i)) : t1 = PointLoct(iPoint(i)) : GOTO RePlot3 ELSE
                        GOTO RePlot3 ELSE REM
                IF PointType(i) = 10 AND icircleflag = 1 THEN GOSUB DrawPoint
                GOTO RePlot4
            RePlot3:
                IF icircleflag = 1 THEN GOSUB DrawPoint
                IF Grid$ = "Contact1" THEN GOSUB DrawContact
                IF Grid$ = "Blob" AND ii <> 0 THEN x2 = x3 : t2 = t3 : GOSUB DrawBlob
                IF ii = 0 AND jj = 0 THEN GOTO RePlot4
                IF Grid$ <> "Blob" THEN GOSUB DrawMech
            RePlot4:
        NEXT i
        LINE (0, 0) - (508, 10), 30, bf
        LINE (0, 10) - (10, 318), 30, bf
        GOSUB Scales
        RETURN

FlashLine:
        PENSIZE 1, 1
        IF itimer = 1 THEN itimer = 0 : LINE (sx1, st1) - (sx2, st2), 33 : RETURN
        IF itimer = 0 THEN itimer = 1 : LINE (sx1, st1) - (sx2, st2), 30
        CIRCLE (sx1, st1), 3
        CIRCLE (sx2, st2), 3
        RETURN

RePicture:
        PICTURE OFF : Image$ = PICTURE$ : PICTURE ON : PICTURE, Image$
        RETURN

DrawBlob:
        CALL PENPAT(VARPTR(idash(0))) : PENSIZE 2, 2
        MOVETO 490#*(x1*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY - 490#*t1*magfactor#*tstretch/(1# + L1L2#)
        LINETO 490#*(x2*magfactor# + L1L2#)/(1# + L1L2#) + 14 + iScrollX, 314 + iScrollY - 490#*t2*magfactor#*tstretch/(1# + L1L2#)
        PENNORMAL : PENSIZE 1, 1
        RETURN

'______________________________________________
' MOUSE INPUT ROUTINES

SelectContact:
        BREAK ON : ibreak = 0
    SelectContact1:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SContactEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectContact1
        ii1 = ii
        BEEP
    SelectContact2:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SContactEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectContact2
        ii2 = ii
    SContactEnd:
        BREAK OFF
        RETURN

SelectExpansion:
        BREAK ON : ibreak = 0
        iexpflag = 0
    SelectExpansion1:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SExpEnd
                IF iexpflag = 0 THEN Test$ = INPUT$(2)
                IF iexpflag = 0 THEN expangle = VAL(Test$)
                IF iexpflag = 0 THEN IF Test$ = "" THEN iexpflag = 0 ELSE iexpflag = 1 : BEEP ELSE REM
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0 : ifudgeflag = 1
        GOSUB CheckPoint : ifudgeflag = 0
        IF icheckflag <> 1 OR iexpflag = 0 THEN BEEP : BEEP : GOTO SelectExpansion1
    SExpEnd:
        BREAK OFF
        RETURN

SelectInterior:
        BREAK ON : ibreak = 0
    SelectInterior1:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SInteriorEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0 : idriverflag = 1
        GOSUB CheckPoint : idriverflag = 0
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectInterior1
        x1 = sx : t1 = st : ii1 = ii
        BEEP
    SelectInterior2:
        s = MOUSE(0)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SInteriorEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectInterior2
        x2 = sx : t2 = st : ii2 = ii
    SInteriorEnd:
        BREAK OFF
        RETURN

CheckPoint:
        IF inewflagc = 0 AND inewflagi = 0 THEN GOTO CheckPoint1
        sxg = 490!*(xg*magfactor# + L1L2!)/(1! + L1L2!) + 14! + iScrollX
        stg = 314! + iScrollY - 490!*tg*magfactor#*tstretch/(1! + L1L2!)
        sxi = 490!*((((314! + iScrollY - 10!)*(1! + L1L2!)/(490!*magfactor#*tstretch) - tg)*(uu3 - aa3) + xg)*magfactor# + L1L2!)/(1! + L1L2!) + 14! +
iScrollX
        sti = 10
        m# = (stg - sti)/(sxg - sxi)
        sx = (ttt - stg + m#*sxg + xxx/m#)/(m# + 1!/m#)
        st = m#*(sx - sxg) + stg
        IF SQR((xxx - sx)^2 + (ttt - st)^2) < 2! THEN icheckflag = 1 ELSE GOTO CheckPoint1
        Grid$ = "Interior"
        i = 0 : j = 0
        x3 = ((sx - 14 - iScrollX)*(1! + L1L2!)/490 - L1L2!)/magfactor#
        t3 = (314 + iScrollY - st)*(1! + L1L2!)/(490*magfactor#*tstretch)
        u3 = uu3
        a3 = aa3
        p3 = p3p4
        TT3 = a3^2
        rho3 = p3/TT3
        GOSUB SavePoint
        IF inewflagi = 1 THEN GOSUB DrawPoint
        inewflagc = 0
        PointLocx(max) = x3 : PointLoct(max) = t3 : iPoint(max) = 0 : jPoint(max) = 0 : PointType(max) = 0
        ii = max
        GOTO CheckEnd
    CheckPoint1:
        FOR ii = max TO 6 STEP -1
                IF idriverflag = 1 AND PointType(ii) = 11 THEN GOTO CheckPoint3
                IF ifudgeflag = 1 AND PointType(ii) = 10 THEN ifudgeflag = 0 : GOTO CheckPoint3
                IF PointType(ii) = 0 OR PointType(ii) = 1 OR PointType(ii) = 2 OR PointType(ii) = 3 OR PointType(ii) = 4 OR PointType(ii) = 6 OR
                        PointType(ii) = 10 OR PointType(ii) = 11 THEN GOTO CheckPoint2
                GOTO CheckPoint3
            CheckPoint2:
                x = PointLocx(ii) : t = PointLoct(ii)
                sx = 490!*(x*magfactor# + L1L2!)/(1! + L1L2!) + 14 + iScrollX
                st = 314 + iScrollY - 490!*t*magfactor#*tstretch/(1! + L1L2!)
                IF ABS(sx - xxx) <= 2! AND ABS(st - ttt) <= 2! THEN icheckflag = 1 : GOTO CheckEnd
            CheckPoint3:
        NEXT ii
    CheckEnd:
        RETURN

SelectSplit:
        BREAK ON : ibreak = 0
    SelectSplit1:
        z = MOUSE(0)
        WHILE MOUSE(0) <> 1
                GOSUB FlashLine
                IF ibreak = 1 THEN GOTO SSplitEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        m# = (st2 - st1)/(sx2 - sx1)
        IF ABS(m#) > 1! THEN xxxd = (xxx/m#^2 + (ttt - st1)/m# + sx1)/(1! + 1!/m#^2) ELSE xxxd = (m#^2*sx1 + (ttt - st1)*m# + xxx)/(1! + m#^2)
        IF ABS(m#) > 1! THEN tttd = (xxx - xxxd)/m# + ttt ELSE tttd = (xxxd - sx1)*m# + st1
        IF SQR((tttd - ttt)^2 + (xxxd - xxx)^2) > 5! THEN icheckflag = 0 ELSE icheckflag = 1
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectSplit1
        xxxd = (xxxd - 14 - iScrollX - 490!*L1L2!/(1! + L1L2!))*(1! + L1L2!)/(490!*magfactor#)
        tttd = (314 + iScrollY - tttd)*(1! + L1L2!)/(490!*magfactor#*tstretch)
    SSplitEnd:
        BREAK OFF
        RETURN

LocateInfo:
        WINDOW 3 : mouseflag = 1
        SETCURSOR VARPTR(icrosshair(0))
    SelectInfo:
        z = MOUSE(0)
        WHILE MOUSE(0) <> 1
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectInfo
        SETCURSOR VARPTR (iwatch(0))
        iDirty = 0
        IF firstinfo <> 1 THEN GOSUB FindInfo
        mouseflag = 0
        RETURN

LocateErase:
        IF infoflag = 1 THEN WINDOW 3
        mouseflag = 1
        SETCURSOR VARPTR(icrosshair(0))
        BREAK ON : ibreak = 0
    SelectErase:
        z = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO EraseEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF PointType(ii) = 2 OR PointType(ii) = 3 OR PointType(ii) = 4 OR PointType(ii) = 7 OR PointType(ii) = 9 OR PointType(ii) = 11 THEN icheckflag
            = 0
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectErase
    EraseEnd:
        BREAK OFF : IF ibreak = 1 THEN RETURN
        SETCURSOR VARPTR (iwatch(0))
        RETURN

LocateScroll:
        IF infoflag = 1 THEN WINDOW 3
        mouseflag = 1
        BREAK ON : ibreak = 0
        SETCURSOR VARPTR(ihand(0))
    SelectScroll:
        z = MOUSE(0)
        WHILE z = 0
```

```
                IF ibreak = 1 THEN GOTO ScrollEnd
                s = MOUSE(0)
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        IF xxx < 0 OR xxx > 507 OR ttt < 0 OR ttt > 317 THEN BEEP : BEEP : GOTO SelectScroll
    ScrollEnd:
        BREAK OFF : IF ibreak = 1 THEN RETURN
        iDirty = 0
        RETURN

SelectBlob:
        BREAK ON : ibreak = 0
    SelectBlob1:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO BlobEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0 : ifudgeflag = 1 : idriverflag = 1
        GOSUB CheckPoint : ifudgeflag = 0 : idriverflag = 0
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectBlob1
        x1 = sx : t1 = st : ii1 = ii
        BEEP
    SelectBlob3:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO BlobEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0 : idriverflag = 1
        GOSUB CheckPoint : idriverflag = 0
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectBlob3
        x3 = sx : t3 = st : ii3 = ii
        BEEP
    SelectBlob6:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO BlobEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0 : idriverflag = 1
        GOSUB CheckPoint : idriverflag = 0
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectBlob6
        x6 = sx : t6 = st : ii6 = ii
    BlobEnd:
        BREAK OFF
        RETURN

SelectPitot:
        BREAK ON : ibreak = 0
    SelectPitot1:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SPitotEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectPitot1
        x1 = sx : t1 = st : ii1 = ii
        BEEP
    SelectPitot2:
        s = MOUSE(0)
        WHILE MOUSE(0) <> 1
                IF ibreak = 1 THEN GOTO SPitotEnd
        WEND
        xxx = MOUSE(5) : ttt = MOUSE(6)
        icheckflag = 0
        GOSUB CheckPoint
        IF icheckflag <> 1 THEN BEEP : BEEP : GOTO SelectPitot2
        x2 = sx : t2 = st : ii2 = ii
        GOSUB PitotCalcs
    SPitotEnd:
        BREAK OFF : WINDOW 4 : INITCURSOR
        RETURN

'______________________________________________
' SUBROUTINES

SUB zeroin(ax,bx,tol,iflag,z) STATIC
        SHARED gamma4, gamma1, R1, R4, T4T1, P4P1, T2T1, p2p1, p1p10, M2#, t6, t3, x6, x3, u6, u3, rho6, rho3, u1, rhom, rho1, W1, t1, x1, ua2, uu2,
            aa2, lm#, rho2rho1, L1L2#, vv, us2C, lm2#, rho20rho10
     Compute eps, the relative machine precision
        eps = 1#
10      eps = eps/2#
        toll = 1# + eps
        IF (toll > 1#) GOTO 10
'    Initialisation
        a = ax
        b = bx
        IF iflag = 0 THEN xx = a : GOSUB Shock1 : fa = yy : xx = b : GOSUB Shock1 : fb = yy
        IF iflag = 1 THEN xx = a : GOSUB Shock2 : fa = yy : xx = b : GOSUB Shock2 : fb = yy
        IF iflag = 2 THEN xx = a : GOSUB BlobCalc : fa = yy : xx = b : GOSUB BlobCalc : fb = yy
        IF iflag = 3 THEN xx = a : GOSUB GCalc : fa = yy : xx = b : GOSUB GCalc : fb = yy
        IF iflag = 4 THEN xx = a : GOSUB GCalcB : fa = yy : xx = b : GOSUB GCalcB : fb = yy
        IF iflag = 5 THEN xx = a : GOSUB LamMir : fa = yy : xx = b : GOSUB LamMir : fb = yy
'    Begin step
20      c = a
        fc = fa
        D = b - a
        e = D
30      IF (ABS(fc) >= ABS(fb)) GOTO 40
        a = b
        b = c
        c = a
        fa = fb
        fb = fc
        fc = fa
'    Convergence test
40      toll = 2#*eps*ABS(b) + .5*tol
        xm = .5*(c - b)
        IF (ABS(xm) <= toll) GOTO 90
        IF (fb = 0#) GOTO 90
'    Is bisection necessary
        IF (ABS(e) < toll) GOTO 70
        IF (ABS(fa) <= ABS(fb)) GOTO 70
'    Is quadratic interpolation possible
        IF (a <> c) GOTO 50
'     Linear interpolation
        s = fb/fa
        p = 2#*xm*s
```

OF POOR QUALITY

```
        Q = 1# - s
        GOTO 60
'     Inverse quadratic interpolation
50      Q = fa/fc
        R = fb/fc
        s = fb/fa
        p = s*(2#*xm*Q*(Q - R) - (b - a)*(R - 1#))
        Q = (Q - 1#)*(R - 1#)*(s - 1#)
'     Adjust signs
60      IF (p > 0#) THEN Q = -Q
        p = ABS(p)
'     Is interpolation acceptable
        IF ((2#*p) >= (3#*xm*Q - ABS(tol1*Q))) GOTO 70
        IF (p >= ABS(.5*e*Q)) GOTO 70
        e = D
        D = p/Q
        GOTO 80
'      Bisection
70      D = xm
        e = D
'     Complete step
80      a = b
        fa = fb
        IF (ABS(D) > tol1) THEN b = b + D
        IF (ABS(D) <= tol1) THEN b = b + ABS(tol1)*SGN(xm)
        IF iflag = 0 THEN xx = b : GOSUB Shock1 : fb = yy
        IF iflag = 1 THEN xx = b : GOSUB Shock2 : fb = yy
        IF iflag = 2 THEN xx = b : GOSUB BlobCalc : fb = yy
        IF iflag = 3 THEN xx = b : GOSUB GCalc : fb = yy
        IF iflag = 4 THEN xx = b : GOSUB GCalcB : fb = yy
        IF iflag = 5 THEN xx = b : GOSUB LamMix : fb = yy
        IF ((fb*(fc/ABS(fc))) > 0#) GOTO 20
        GOTO 30
'     Done
90      z = b
        GOTO ZeroEnd
     Shock1:
        IF ilangleyflag = 0 THEN yy = xx*(SQR((gamma4 + 1#)/2#) - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(xx -
            1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(xx - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        IF ilangleyflag = 1 THEN yy = xx*(1# - ((gamma4 - 1#)*SQR((gamma1*R1)/(gamma4*R4*T4T1))*(xx - 1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma
            1#)*(xx - 1#)))))^(-2#*gamma4/(gamma4 - 1#)) - P4P1
        RETURN
     Shock2:
        yy = xx*(1# + (gamma1 - 1#)*M2#/2# - ((gamma1 - 1#)*SQR(1#/T2T1)*(xx - 1#))/(SQR((2#*gamma1)*(2#*gamma1 + (gamma1 + 1#)*(xx - 1#))))^(-
            2#*gamma1/(gamma1 - 1#)) - p2p1*p1p10
        RETURN
     BlobCalc:
        t2 = (t6 - t3)*(xx - x6)/(x6 - x3) + t6
        u2 = (SQR((t3 - t2)^2 + (x3 - xx)^2)*u6 + SQR((t6 - t2)^2 + (x6 - xx)^2)*u3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        rho2 = (SQR((t3 - t2)^2 + (x3 - xx)^2)*rho6 + SQR((t6 - t2)^2 + (x6 - xx)^2)*rho3)/SQR((t6 - t3)^2 + (x6 - x3)^2)
        IF rhom > 1# THEN W2 = (u2 + u1)/2# ELSE W2 = (u2 - u1)*(3#/(2#*rhom + 1#)) + W1
        x2 = (t2 - t1)*2#*W1*W2/(W1 + W2) + x1
        yy = xx - x2
        RETURN
     GCalc:
        yy=(5#/8#)*(LOG((1#-(xx*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(xx*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((xx*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(xx*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2/(rho2rho1*2#*lm#))*(xx/(uu2-aa2)+L1L2#/us2)
        RETURN
     GCalcB:
        yy=(5#/8#)*(LOG((1#-(xx*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)/(1#+(xx*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#))-2#*ATN((xx*vv*(us2/(uu2-aa2)-
            1#)/lm#)^.2#)+4#*(xx*vv*(us2/(uu2-aa2)-1#)/lm#)^.2#)+(us2*vv/(rho2rho1*2#*lm#))*(xx/(uu2-aa2)+L1L2#/us2)
        RETURN
     LamMix:
        yy = LOG(1# - SQR((us20*(xx - L1L2#/us2) - 1#)/lm2#)) + SQR((us20*(xx - L1L2#/us2) - 1#)/lm2#) + us20*(xx - L1L2#/us2)/(2#*lm2#*rho20rho10)
        RETURN
     ZeroEnd:
        END SUB
```

National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CR-181722

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Expansion Tube Test Time Predictions

5. Report Date
September 1988

6. Performing Organization Code

7. Author(s)
C. M. Gourlay

8. Performing Organization Report No.
8/88

10. Work Unit No.
505-62-81-61

9. Performing Organization Name and Address
University of Queensland
Department of M.E.
St. Lucia, Queensland 4067
AUSTRALIA

11. Contract or Grant No.
NAGW-674

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Griffin Y. Anderson
Final Report

16. Abstract

The interaction of an interface between two gases and a strong expansion is investigated and the effect on flow in an expansion tube is examined. Two mechanisms for the unsteady pitot-pressure fluctuations found in the test section of an expansion tube are proposed. The first mechanism depends on the Rayleigh-Taylor instability of the driver-test gas interface in the presence on a strong expansion. The second mechanism depends on the reflection of the strong expansion from the interface. Predictions compare favourably with experimental results. The theory is expected to be independent of the absolute values of the initial expansion tube filling pressures.

**ORIGINAL PAGE IS
OF POOR QUALITY**

17. Key Words (Suggested by Author(s))
shock tunnel, stability, interface
hypervelocity

18. Distribution Statement
Unclassified - Unlimited

Subject Category 09

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	97	A05

C-2