A DIGITAL COMPUTER PROGRAM FOR CALCULATING THE PERFORMANCE OF SINGLE- OR MULTIPLE-DIA弗RAM SHOCK TUBES FOR ARBITRARY EQUILIBRIUM REAL GAS MIXTURES

by William L. Grose, John E. Nealy, and Jane T. Kemper
Langley Research Center
Langley Station, Hampton, Va.
A DIGITAL COMPUTER PROGRAM FOR CALCULATING THE
PERFORMANCE OF SINGLE- OR MULTIPLE-DIAPHRAGM
SHOCK TUBES FOR ARBITRARY EQUILIBRIUM
REAL GAS MIXTURES

By William L. Grose, John E. Nealy, and Jane T. Kemper

Langley Research Center
Langley Station, Hampton, Va.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00
A DIGITAL COMPUTER PROGRAM FOR CALCULATING THE
PERFORMANCE OF SINGLE- OR MULTIPLE-DIAPHRAGM
SHOCK TUBES FOR ARBITRARY EQUILIBRIUM
REAL GAS MIXTURES

By William L. Grose, John E. Nealy, and Jane T. Kemper
Langley Research Center

SUMMARY

A computer program written in FORTRAN IV language is presented which deter-
mines the performance of a shock tube for arbitrary equilibrium real gas mixtures. For
specified initial gas composition and charging conditions, the program output includes
velocity, pressure, density, enthalpy, temperature, sound speed, and species mole frac-
tions at any point in the shock tube cycle.

Several representative calculations illustrate the utility of the program. The pro-
gram is applicable to both simple and buffered shock tubes as well as to the expansion
tube.

INTRODUCTION

The shock tube has been widely utilized as an experimental device for producing
high enthalpy flows for research in such areas as aerodynamic testing, chemical kinetics,
and radiation gas dynamics. Much of the utility of the shock tube is due to the capability
of generating a broad range of test conditions in various gas mixtures.

Numerous current experimental investigations are being conducted in gas mixtures
simulating planetary atmospheres that differ markedly from air. Before initiating a test
program, it is essential to ascertain the theoretical performance of the shock tube for the
particular gas mixture being used in order to aid in determining test conditions and in
analyzing data. Previous shock tube performance investigations are too numerous to
describe herein, but they generally can be divided into two classes. Both simple and
buffered shock tubes are discussed in reference 1 and this study is an example of the
analyses that assume a perfect gas throughout the operating cycle. In reference 2 an
investigation of the expansion tunnel is presented and is illustrative of those analyses that
treat the shocked gas in the driven chamber as a real gas in thermodynamic and chemical
equilibrium, but the driver gas in the unsteady expansion process is assumed to be
perfect. The rather exhaustive treatment of the perfect gas performance of the simple shock tube in reference 3 does, however, include a discussion of real gas effects.

It is desirable to construct a digital computer program to calculate shock tube performance for arbitrary driver and driven gases so that the gas at any point in the operating cycle may be treated either as a perfect gas or as an equilibrium real gas. This report describes such a program based upon the assumptions of inviscid one-dimensional flow, no mixing at the interfaces, and instantaneous diaphragm bursts. The program is applicable to simple or buffered shock tubes and expansion tubes. Performance parameters for some typical gas mixtures and operating conditions are presented to demonstrate the utility of the program.

SYMBOLS

\begin{align*}
a & \quad \text{sound speed, cm/sec} \\
h & \quad \text{enthalpy, ergs/g} \\
p & \quad \text{pressure, dynes/cm}^2 \\
p_0 & \quad \text{standard atmospheric pressure, } 1.01325 \times 10^6 \text{ dynes/cm}^2 \\
R & \quad \text{universal gas constant, } 8.31469 \times 10^7 \text{ ergs/mole-}^0\text{K} \\
s & \quad \text{entropy, ergs/g-}^0\text{K} \\
T & \quad \text{temperature, } ^0\text{K} \\
t & \quad \text{time, sec} \\
u & \quad \text{flow speed, cm/sec} \\
U_S & \quad \text{shock speed, cm/sec} \\
x & \quad \text{distance, cm} \\
\gamma & \quad \text{ratio of specific heats} \\
\mu & \quad \text{molecular weight} \\
\end{align*}
\[ \rho \quad \text{density, g/cm}^3 \]

Subscripts:

0 \quad \text{reference conditions}

1 \quad \text{state of quiescent gas in front of normal shock}

2 \quad \text{state of gas behind normal shock}

3 \quad \text{state of expanded driver gas}

4 \quad \text{initial driver gas conditions}

5 \quad \text{state of test gas in expansion tube}

10 \quad \text{state of quiescent gas in double-diaphragm shock tube}

**ANALYSIS**

A distance-time diagram for the simple shock tube is schematically depicted in figure 1(a). Figure 1(b) illustrates the quiescent driver and test gases prior to diaphragm burst at \( t = 0 \). Figure 1(c) indicates the wave system at \( t = t_a \). The test gas in state (1) is compressed and heated by the normal shock wave to state (2), while the driver gas in state (4) undergoes an isentropic unsteady expansion to state (3).

In practice the shock tube is often designed to operate at a fixed driver pressure and the driven chamber pressure (and the buffer chamber pressure for the buffered shock tube) is adjusted to produce the desired test conditions. For specified initial charging conditions (1) and (4), the unsteady expansion must be solved for state (3) and the normal shock wave must be solved for state (2). The respective solutions for states (2) and (3) are compatible only when the pressures and velocities of the two states are equal (i.e., \( p_2 = p_3 \) and \( u_2 = u_3 \)).

**Unsteady Expansion**

The governing differential equation (ref. 4) for an isentropic unsteady expansion is

\[
du = \tau \left( \frac{dh}{a} \right)_S
\]  

(1)
where the negative and positive signs refer to upstream and downstream waves, respectively. Then, the velocity increment imparted to the flow by the expansion is

\[ \Delta u = u_3 - u_4 = \frac{1}{T} \int_{h_4}^{h_3} \frac{dh}{a} \]  

(2)

In order to relate the pressure and velocity in the as yet undetermined state for an equilibrium real gas, it is necessary to resort to numerical techniques to evaluate the integral in equation (2). The analysis formulated herein utilizes the equilibrium program of reference 5. The program includes ionization and dissociation and involves the following assumptions:

1. The mixture is composed of ideal gases.
2. For diatomic species the rigid-rotor harmonic oscillator model is used with vibrational-rotational corrections for each electronic configuration.
3. Only electronic levels with principal quantum number less than or equal to 5 are included.

The free energy of each of the species is determined from the partition function of quantum statistical mechanics. The equilibrium composition is then arrived at by minimization of free energy.

This program is used to generate an array of the thermodynamic state variables \( p, h, a, \) and \( s \) for the gas mixture over selected ranges of pressures and temperatures. Interpolation within the array permits tabulation of \( p, \rho, h, T, \) and \( a \) for constant \( s = s_4. \) The integral in equation (2) can then be evaluated by a second-order Gaussian quadrature (ref. 6) between limit \( h_4 \) and an assumed limit \( h_3. \) The pressure \( p_3 \) is then determined from the tabulation at the assumed \( h_3. \) It is thus possible to determine a unique correspondence between \( p_3 \) and \( u_3 \) at constant \( s_4. \)

For a perfect gas, equation (2) can be integrated in closed form to yield

\[ p_3 = p_4 \left[ 1 - \frac{\gamma - 1}{2} \frac{u_3 - u_4}{\gamma R T_4} \right]^{\frac{2 \gamma}{\gamma - 1}} \]  

(3)

Normal Shock Wave

The conservation equations valid across a moving normal shock wave are

\[ \rho_1 U_S = \rho_2 \left( U_S - u_2 \right) \]  

(4)
The solution to equations (4) to (6) for an equilibrium real gas requires an iterative method to determine the pressure and velocity in state \( \Theta \). This analysis utilizes the normal shock program of reference 7, which is based upon a free-energy minimization technique for calculating equilibrium thermodynamic state variables coupled with the conservation equations and a modified Newton-Raphson iteration method. The equilibrium thermochemical calculations involve the same assumptions as those in reference 5. For a specified state \( \Theta \) and incident shock velocity, this method determines \( p_2 \) and \( u_2 \). For a range of values of \( U_S \), the correspondence between \( p_2 \) and \( u_2 \) can then be established.

For a perfect gas, the following relation between \( p_2 \) and \( u_2 \) can be readily derived:

\[
    u_2 = \frac{\sqrt{\gamma R T_1}}{\mu} \left( \frac{p_2}{p_1} - 1 \right) \left( \frac{2\gamma}{\gamma + 1} \right)^{1/2} \frac{\gamma - 1}{\gamma + 1} \frac{p_2}{p_1 + \gamma - 1} \right)^{1/2}
\]

### Method of Solution

The solution generated by the program described herein is based upon the previously noted requirement that \( p_3 \) equal \( p_2 \) and \( u_3 \) equal \( u_2 \). By following the method outlined for the solution of the unsteady expansion, a monotonically decreasing sequence of pressure as a function of velocity can be calculated. In similar fashion, the normal shock solution yields a monotonically increasing sequence of pressure as a function of velocity. The method of false position coupled with a Lagrangian interpolation (ref. 6) is then used to determine the crossover point of the two sequences. The crossover values of \( p \) and \( u \) are the required solution.

### PROGRAM DESCRIPTION

For application to the simple shock tube (fig. 1), the program contains the four options corresponding to the combinations of either a real or perfect driver gas and either real or perfect driven gas. For the perfect gas option, it is only necessary to solve the closed-form equations (3) and (7) for the expansion and normal shock,
respectively. For the perfect gas expansion, the required input consists of the initial conditions $p_4$, $T_4$, and $u_4$ and the constants $\mu_4$, $\gamma_4$, and $R$; in addition, a range of velocities

$$u_4 < u_3 < u_4 + \frac{2}{\gamma - 1}\sqrt{\frac{\gamma R T_4}{\mu}}$$

is required to generate a velocity-pressure $(u_3-p_3)$ curve.

For the perfect gas normal shock, the required inputs are $p_1$, $T_1$, $\mu_1$, $R$, $\gamma_1$, and a range of $p_2$ values

$$p_1 < p_2 < p_4$$

from which a pressure-velocity $(p_2-u_2)$ function is computed.

The real gas expansion calculation utilizes the program of reference 5 to generate thermodynamic data. In addition to the input required for the program of reference 5, suitable ranges of temperatures and pressures must be included to define the array of thermodynamic variables. For the constant entropy value $s_4$, the program reads from the array $h$ as a function of $1/a$ and then performs the numerical integration

$$u_3 - u_4 = \int_{h_4}^{h_3} \left(\frac{dh}{a}\right)s_4$$

from which the pressure-velocity $(p_3-u_3)$ function is determined. The option exists for readout of velocity-pressure $(u_3-p_3)$ curves for various entropy values, as well as $u_3$ as a function of $h$ and $T$, or for proceeding directly to the solution of the normal shock.

The program of reference 7 is used to determine equilibrium conditions behind a normal shock for a range of $U_S$ values and for given $p_1$, $T_1$, and gas composition. Each $U_S$ value thus leads to a value of $p_2$ and corresponding value of $u_2$, which define the pressure-velocity function to be matched to the velocity-pressure $(u_3-p_3)$ curve from the expansion calculation.

For expansion tube or buffered shock tube performance calculation, the program is applied twice to the simple shock tube cycle (fig. 2). The conditions behind the normal shock (state 2) in the first calculation become the driver gas conditions (state 4) for the second calculation. It should be noted that the driver velocity for the second calculation is no longer zero, since instantaneous diaphragm rupture has been assumed (i.e., no reflection). The fact that the gas behind the first incident shock in such a system may be dissociated and ionized emphasizes the necessity for a real gas calculation of the expansion phase.
The program FORTRAN listing and a flow chart are presented for reference in the appendix.

SAMPLE CALCULATIONS

Several sample calculations illustrate the application of the program to the determination of performance for single- and double-diaphragm shock tubes. The driver pressures chosen are such that intermolecular forces may be neglected and the assumption of a mixture of ideal gas is valid.

Figure 3 shows the velocity-pressure \((u_3-p_3)\) curve for the equilibrium real gas isentropic expansion of helium and the velocity-pressure \((u_2-p_2)\) curves for normal shock waves in equilibrium air for four initial driven chamber pressures with \(T_1 = 300^\circ\text{K}\). The helium driver conditions chosen are \(T_4 = 15000^\circ\text{K}\) and \(p_4/p_0 = 315\). The crossover point of the curves determines \(p_3 = p_2\) and \(u_3 = u_2\) for each driver-driven configuration. From these results the shock speeds corresponding to each calculation may be evaluated. The performance plot of \(U_{S1}\) as a function of \(p_1/p_0\) for these driver conditions is given in figure 4.

Figures 5 to 9 illustrate a typical set of performance calculations for the expansion tube. This type of facility (ref. 8) consists of three sections separated by two destructible diaphragms as shown in figure 2. One section contains the driver gas \((4)\), the intermediate section contains the test gas \((1)\), and the remaining section contains the accelerating gas \((10)\).

For the performance calculations, the simple shock tube cycle was applied first to regions \(4\) and \(1\) in order to obtain conditions in region \(2\). Then the calculation was repeated, with conditions in region \(2\) as the driver and the gas in region \(10\) as the driven gas, which led to the determination of the test gas conditions in region \(5\).

The following initial conditions were chosen for these computations:

\[
\frac{p_4}{p_0} = 30; \quad T_4 = 2500^\circ\text{K}; \quad \text{helium (perfect gas)}
\]

\[
10^{-3} \leq \frac{p_1}{p_0} \leq 10^{-1}; \quad T_1 = 300^\circ\text{K}; \quad \text{air (real gas)}
\]

\[
10^{-3} \leq \frac{p_{10}}{p_0} \leq 10^{-2}; \quad T_{10} = 300^\circ\text{K}; \quad \text{helium (perfect gas)}
\]

In this calculation the temperatures experienced by the helium were not great enough to cause a noticeable departure from perfect gas behavior. Figure 5 shows the shock speed
in air $U_s,1$ as a function of $p_1/p_o$ resulting from the first diaphragm burst. The final shock speed in helium in the accelerating chamber is shown in figure 6 for $\frac{p_{10}}{p_o} = 5 \times 10^{-4}$, $10^{-3}$, and $5 \times 10^{-3}$. The conditions of most interest for this facility are those of the gas in region (5), which has been shocked in the first cycle and expanded and cooled in the second cycle. The test gas conditions which may be obtained from the given initial conditions are shown in figures 7, 8, and 9. As would be expected, the families of curves for shock speed $U_{s,10}$ and test gas speed $u_5$ are similar (figs. 6 and 7). The performance data indicate that the facility may be operated in a manner in which the test gas speed and pressure do not change appreciably although the temperature may vary from several thousand degrees to several hundred degrees Kelvin.

This program was also used to make a set of calculations for the 3.8-inch double-diaphragm shock tube at the Langley Research Center in order to predict the real gas performance for a test gas mixture of 90 percent N$_2$ and 10 percent CO$_2$. The calculations for a double-diaphragm, or buffered, shock tube are carried out in the same manner as those described for the expansion tube. However, in this facility the test gas is contained in the third chamber instead of in the intermediate section; hence, the results of primary interest are the shock speeds in the test gas for a range of $\frac{p_1}{p_o}$ and $\frac{p_{10}}{p_o}$, which are shown in figures 10 and 11.

Figure 10 gives the performance of this shock tube for the following initial conditions:

$$\frac{p_4}{p_o} = 100; \quad T_4 = 300^0 \text{ K}; \text{ helium (perfect gas)}$$

$$5 \times 10^{-2} \leq \frac{p_1}{p_o} \leq 5; \quad T_1 = 300^0 \text{ K}; \text{ helium (perfect gas)}$$

$$5 \times 10^{-4} \leq \frac{p_{10}}{p_o} \leq 5 \times 10^{-2}; \quad T_{10} = 300^0 \text{ K}; \text{ 90 percent N}_2 \text{ and 10 percent CO}_2 \text{ (real gas)}$$

Figure 11 shows the performance for these same initial conditions for gases in regions (1) and (10) but with perfect hydrogen used instead of helium for gas in region (4). It has been observed experimentally that for the initial conditions for which this facility is operated, the performance calculations give a reasonably accurate prediction of actual behavior.

The results of all sample calculations were checked where possible with the results of references 1, 2, 4, 7, and 8. In all calculations agreement was within 1 percent.
RÉSUMÉ

A computer program written in FORTRAN IV language is presented which determines the performance of a shock tube for arbitrary equilibrium real gas mixtures. For specified initial gas composition and charging conditions, the program output includes velocity, pressure, density, enthalpy, temperature, sound speed, and species mole fractions at any point in the shock tube cycle.

Several representative calculations illustrate the utility of the program. It was noted that these calculations agreed with results of other investigations.

It may be concluded that the program has both generality and flexibility for computing shock tube performance in arbitrary gas mixtures.

Langley Research Center,
National Aeronautics and Space Administration,
Langley Station, Hampton, Va., June 10, 1968,
129-01-03-10-23.
APPENDIX

SHOCK TUBE PERFORMANCE PROGRAM LISTING AND FLOW CHART

The FORTRAN listing and a schematic flow diagram of the program are presented. The program is composed of nine subroutines as follows:

(1) PEREX – Computes $p_3$ as a function of $u_3$ for a given $p_4$, $u_4$, $T_4$, and $R$

(2) ROGO – Computes an array of thermodynamic state variables for a given range of $p$ and $T$ (program of ref. 5)

(3) INTER – Uses a three-point Aitkens interpolation formula to find $p$, $h/RT$, and $a/a_0$ at constant $(s/R)_4$

(4) INTEG – Computes $\int \frac{dh}{a}$ by Gaussian quadrature

(5) PERNS – Computes $u_2$ for a given $p_1$, $p_2$, $T_1$, $\gamma_1$, $\mu_1$, and $R$

(6) NORMAL – With given $p$, $T_1$, and $u$, computes normal shock properties of gas
   This subroutine makes use of an equilibrium program which computes thermodynamic properties of a gas at a given $T$ and $p$ (program of ref. 7).

(7) SOLUT – Given $(p_2, u_2)$ and $(p_3, u_3)$, finds solution to curves

(8) FTLUP – Uses Aitkens interpolation formula for three points; finds $U_S$ to correspond to $u_2$

(9) SLITE – Sets a flag to be checked later on in subroutine

Subroutines (1) to (4) apply to the expansion phase only and the program may be stopped at this point. Also, $u_3$, $p_3$, $T_3$, $(h/RT)_3$, and $(a/a_0)_3$ may be punched out to save for later computer runs so that the curves do not have to be recomputed. Subroutines (5) to (9) apply to the normal shock portion of the program.

The program listing and flow chart are reproduced in the following pages.
PROGRAM D1282 (INPUT, OUTPUT, PUNCH, TAPE5=INPUT, TAPE6=OUTPUT, TAPE9)
REAL MU
INTEGER PUN, COMPUT, REAL, EXPO
DIMENSION AM(30), HM(30)

INPUT FOR EXPANSION - REAL GAS

COMMON PN(30), TM(30), NUMT, NCAPX, SOR4, U4

INPUT FOR EXPANSION - PERFECT GAS

COMMON P4, T4, U3(30)

INPUT CODES

COMMON PUN, EXPO, REAL, COMPUT

INPUT FOR NORMAL SHOCK

COMMON P1, T1, P2(30), NUMP, USTO(30), U2(30)
COMMON ISPEC(30), JMOL(10), M, N, BETA(5), NS,
1 AMC, NB, NBTA(5)

INPUT FOR PERFECT EXPANSION AND SHOCK

COMMON MU, GAMMA, R

GAS NAME

COMMON NAME(10)

COMMON/BLOCK/HM, AM, MM, P3(30)

NAMELIST/EXP/PN, TM, NUMT, NCAPX, SOR4, U4, P4, T4, U3, MU, GAMMA, R, PUN,
1 EXPO, REAL, COMPUT
NAMELIST/NORS/P1, T1, P2, NUMP, USTO, U2, ISPEC, JMOL, M, N, BETA, NS, NB,
1 NBTA, MU, GAMMA, R, REAL, COMPUT
EXPO=1
CALL SLITE(1)
NS=0
REAL =0
COMPUT=0
1 READ(5,2010)NAME
2010 FORMAT(10A6)
READ(5, EXP)
WRITE(6,2008)
2008 FORMAT(31HSHOCK TUBE PERFORMANCE PROGRAM//14HEN. J. NEALY*10X*
115HPROG. J. KEMPER)

C IF (COMPUT =1, OR 0 READ IN U3*P3 * COMPUTE U3, P3
C IF (COMPUT*EQ*0) GO TO 10
C READ IN U3, P3
C IF (REAL*EQ*1) GO TO 5
READ(5,2005) SOR4*U4
2005 FORMAT(~~X*E~S.B~~XIE~~~~)
GO, TO 6
5 READ(5,2006) P4*T4
2006 FORMAT(16X*E15*8.5X*E15*8)
6 READ(5,2007) (P3(I)I=1,NUMT)
2007 FORMAT(4E17.8)
READ(5,2007) (U3(I),I=1,NUMT)
GO TO 101
10 CONTINUE
C IF REAL =0+1 REAL* OR PERFECT GAS
IF (REAL*EQ*0) GO TO 15
CALL PEREX(P4*,U4*,GAMMA*R*T4*,MU*,U3*,NUMT*,P3)
GO TO 101
15 CALL ROGO(PN*,TM*,NUMT*,NCAPX*,GAMMA*,SOR4)
MM=NUMT
CALL INTEG(U3*,U4*)
WRITE(6,201) SOR4*,(1,TM(I)*P3(I)*HM(I)*AM(I)*U3(I),I=1,NUMT)
201 FORMAT(7HOUTPUT/15X,4HS/R=E14*7/3H M *2X*8X*1HT,8X,10X,1HP,
118X,1HH,18X,1HA,
118X,2HU3/(13,5(2X,E17*8)))
101 IF (PUN*EQ*0, 1) GO TO 110
100 WRITE(6,2009)P4*T4*,(U3(I),P3(I),I=1,NUMT)
2009 FORMAT(1H1*2X,3HP4=E15*7,2X,3HT4=E15*7,8X,2HU3,17X,2HP3/
1(E17*8,2X,E17*8))
102 IF (EXP*EQ*1) GO TO 150
GO TO 1
C PUNCH U3*,P3
C 110 IF (REAL*EQ*0) GO TO 111
PUNCH 2000*,P4*,T4
2000 FORMAT(11HPERFECT GAS*2X*3HP4=E15.8*2X*3HT4=E15.8)
   GO TO 112
111 PUNCH 2001, SOR4, U4
2001 FORMAT(8HREALGAS*2X*4HS/R=E15.8*2X*3HU4=E15.8)
   GO TO 112
112 PUNCH 2003*(P3(I),I=1,NUMT)
2003 FORMAT(4E17.8*4X*2HP3)
   PUNCH 2004, (U3(I),I=1,NUMT)
2004 FORMAT(4E17.8*4X*2HU3)

C
   GO TO 102
C
NORMAL SHOCK
C
150 READ(5*NORMS)
   CALL SLITE(2)
1515 WRITE(6,206)NAME, P4, T4, T1
206 FORMAT(7H1OUTPUT/1H0, 9X, 2HP4, 17X, 2HT4, 17X, 2HT1, 20X, 10A6/
   X3(E17.8*2X, 7X, 2HP1
1*17X, 2HUS, 17X, 2HU2, 17X, 2HP2/)
IC=1
159 IF(COMPUT.EQ.0) GO TO 160
   GO TO 165
160 IF(REAL.EQ.0) GO TO 164
161 CALL PERN5(P1, T1, GAMMA, MU, R, P2, U2, NUMP)
   GO TO 165
164 CALL NORMAL(P1, T1)
165 CALL SOLUT(U3, P3, U2, P2, NUMT, NUMP, U*P)
   IF(REAL.EQ.0) GO TO 170
   UP=SQRT(GAMMA*R*T1/MU)*(SQRT(1+(P/P1-1)*((GAMMA+1)/(2*GAMMA))))
   UP=UP/30*48
   GO TO 175
170 CALL FTLUP(U, UP, -1, NUMP, U2, UST0)
175 CONTINUE
   IF(NS.EQ.1) WRITE(6,4456)
   WRITE(6,207)P1, UP, *P
207 FORMAT(4(E17.8*2X))
   IF(NS.EQ.1) WRITE(6,4456)
4456 FORMAT(/)
   PLAST=P1

C
C READ NEXT P
READ(5*NORMS)
   IF(P1.EQ.PLAST) GO TO 1
   GO TO 159

END
$IBFTC PEREX DECK

SUBROUTINE PEREX(P4, U4, GAMMA, R, T4, MU, U3, M, P3)
DIMENSION U3(30), P3(30)
REAL MU
GAMY = 2. * GAMMA / (GAMMA - 1.)
P4 = P4 * 1.01325E6
DO 10 I = 1, M
   P3(I) = P3(I) / 1.01325E6
10 P3(I) = P3(I) / 1.01325E6
P4 = P4 / 1.01325E6
RETURN
END
SUBROUTINE ROGO(PN, TM, NUMT, NCAPX, GAM, SOR4)
C  PERRY NEWMAN EQUILIBRIUM THERMODYNAMIC PROPERTIES WITH DERIVATIVES
80  FORMAT(215*2E14*l8)
81  FORMAT(A6,315*3E14*8)
82  FORMAT(5E14*8)
83  FORMAT(E14*8*l5)
90  FORMAT(//4H  MU=E15*8,2X,3HP0=EX15*8,2X,5HRH00=EX15*8,2X,3HA0=E15*8)
93  FORMAT(//(8E16*8))
97  FORMAT(//,1X6HA(I,J),2X2HI=I4,2X2HJ=I4/(8E16*8))
98  FORMAT(1X2HP=E15*8,2X2BHI00 ITERATIONS-NONCONVERGENT)
100  FORMAT(16HlEXPANSION PHASE)
101  FORMAT(75H0EQUILIBRIUM THERMODYNAMIC PROPERTIES WITH DERIVATIVES 1 IN REAL GAS SYSTEM//)
999  FORMAT(/)
1000 FORMAT(1Hl,57X2HT=I5,lXIHK//)
1001 FORMAT(6X3HLOG,5X3HLOG,7X1HZ,9X4HP/RT,6X3HS/R,6X3HLOG,6X6H1RHD0T,5
1X6HDRH0DP,3X4HCP/R,6X4HCV/R,5XSGAMMA,3X6H0AMAE,3X4HA/A0/5X4HP/P0
2X3BHRHO/RH00,3X4BHNHE,5X7H(T/RHO),4X7H(P/RHO)//)
      REAL NO, M, LAMB, LAMBDA, MU, NE, NEFR, LOGNE, MASSFR, LMIN
      INTEGER F(30), V(30), 10)
      DIMENSION SPECIE(30), LB(30), M(30), DELHF(30), BETA(30), NDEBUG(30)
1 1  IPIVOT(10), R(I,10), SUMAY(10,1), G(30,30), E(30,30), BE(30,10), ALPHA
2E(30,10), OMEGA(30,10), OMEGAX(30,10), XOMEG(30,4), XOMEGX(30
3,4), Z(30), SIGMA(10,10), U(10), DELTA(10), GAMMA(10), XX(10,10), Y(30,10), X
4(30), A(30,9), HORT(30,9), FORT(30), NEFR(30), PI(9*2), XPRIME(30), MSSF
SR(30), CAPX(50), YINT(30,15), CSUBP(30), PSI(30,2), CON(10,2), DXDT(30,R
6R(10,10), 0(10,10), ABL(30), SBL(30), HBL(30), PN(30), TM(30)
    WRITE(6,100)
7    WRITE(6,101)
7    CALL SLITET(1, JJ)
7    IF(JJ.EQ.2) GO TO 7777
H=6.62517E-27
XK=1.38044E-16
PREF=1.013250E+6
NO=6.03322E+23
C=2.99793E+10
1 READ(5,80) NUMSP, JINDX, EA, ER
C  IF NDEBUG EQUALS 0, DEBUG
DO 3 I=1, NUMSP
READ(5,81) SPECIE(I), LB(I), F(I), NDEBUG(I), M(I), DELHF(I), BETA(I)
IL=LB(I)
READ(5,82) (G(I*L),E(I*L),L=1,IL)
IF(F(I)*EQ.0) GO TO 3
IF(F(I)*EQ.2) GO TO 123
READ(5,84) (BE(I*L),ALPHA(I*L),OMEGA(I*L),OMEGAX(I*L),V(I*L),L=1,IL)
GO TO 3
123 READ(5,82) BE(I*L),ALPHA(I*L)
READ(5,82) (XOMEG(I*LW),XOMEGX(I*LW),LW=1,4)
CONTINUE
READ(5,82) MU*(YINT(I*1),I=1,NUMSP)
READ(5,82) ((A(I*J),J=1,JINDX),I=1,NUMSP)
7777 CONTINUE
RHOO=PREF*MU/(NO*XX*273.15)
AO=SQRT(GAM+(PREF/RHOO))
WRITE(6,90) MU,PREF,RHOO,AO
DO 278 KP=1,NCAPX
278 CAPX(KP)=ALOG10(PN(KP))
279 DO 19 KI=1,NUMT
T=TM(KI)
KT=T
WRITE(6,1000) KT
WRITE(6,1001)
NY=1
PART=H*(C/(XX*T))
DO 9 I=1,NUMSP
IL=LB(I)
IF(F(I)*EQ.1) GO TO 111
IF(F(I)*EQ.2) GO TO 112
QSUM=0*
FPQSUM=0*
SPQSUM=0*
DO 2 L=1,IL
Z(L)=PART*E(I*L)
GEZ=G(I*L)*EXP(-Z(L))
QSUM=QSUM+GEZ
FPQSUM=FPQSUM+GEZ*Z(L)
2 SPQSUM=SPQSUM+(Z(L)-2.)*GEZ*Z(L)
FPQSUM=FPQSUM/T
SPQSUM=SPQSUM/T**2
QI=(M(I)**T**32807618)**1.5*QSUM*13623883*T
GO TO 71
111 QSUM=0*
FPQSUM=0*
SPQSUM=0*
DO 11 L=1,IL
Z(L) = PART*E(I*L)
SIGMA(L) = PART*(BE(I*L) - 5*ALPHAE(I*L))
U(L) = PART*(OMEGA(I*L) - 2*OMEGAX(I*L))
DELTA(L) = ALPHAE(I*L) + 5*(BE(I*L) - 5*ALPHAE(I*L))
GAMMA = (BE(I*L) + OMEGA(I*L) - 1*OMEGAX(I*L))
XX(L) = OMEGAX(I*L) + OMEGA(I*L) - 2*OMEGAX(I*L)
THREE = 0
FOUR = 0
FIVE = 0
NV = V(I*L) + 1
DO 4 IV = 1, NV
   W = IV - 1
   CC = (1 - W*DELTA(L))
   AA = SIGMA(L)*CC
   BB = U(L)*W - XX(L)*W*(W - 1)
   ONE = 1 + AA + 2*GAMMA(L)/(AA**2 + CC) + 33333333 + AA/12*
   TWO = 1 + AA + 2*GAMMA(L)/(AA**2 + CC) - AA/12 - 394*GAMMA(L)**2/(AA**3*
   IC**2) = THREE + ONE*EXP(-BB) = FOUR + FIVE - TWO*EXP(-BB)
   FIVE = FIVE + (BB**2*ONE + BB*TWO + GAMMA(L)/(AA**2 + CC) + 48 - 3476*GAMMA(L)/
   AA + 6080*GAMMA(L)**2/(AA**2 + CC**2) + 2 + AA*EXP(-BB))
   GEZ = G(I*L)*EXP(-Z(L))
   Q(L) = THREE/BETA(I)*GEZ
   QSUM = QSUM + Q(L)
   FPQSUM = FPQSUM + (FOUR + THREE*Z(L)) + GEZ
   QSUM = QSUM + Z(L)**2*EXP(-Z(L))
11 SPQSUM = (SPQSUM + Z(L)**2)*THREE + TWO*Z(L) + FOUR*FIVE*GEZ
   FPQSUM = FPQSUM/(T*BETA(I))
   SPQSUM = SPQSUM/(T**2*BETA(I))
   QI = (M(I)*T**3.2807618)**1.5*QSUM**1.13623883*T
   SUM1 = SUM1 + GEZ
   SUM2 = SUM2 + ONE*GEZ
   GO TO 71
112 SIGMA = PART*(BE(I*L) - 5*ALPHAE(I*L))
   SIGMA = SIGMA + SUM1
   SUM2 = SUM2 + W*DELTA(L)
   W = W + 1
   GO TO 112

APPENDIX
71 HORT(I) = 2.5 + T/QSUM*FPQSUM + DELHF(I) / (NO*XK*T)
   FORT(I) = DELHF(I) / (NO*XK*T) - ALOG(QI)
   NEGFRT(I) = -FORT(I)
9 CSUBP(I) = 2.5 + 2.0*T/QSUM*FPQSUM - (T*FPQSUM/QSUM)**2 + T**2*SPQSUM/QSUM
   NNN = 1
   DO 39 KP = 1 + NCAPX
9500 P = (10.0**CAPX(KP))**PREF
   NYY = NY
   DO 500 I = 1 + NUMSP
500 Y(I) = YINT(I + NY)
9501 NUMIT = 0
   *^* MM = JINDX + 1
   DO 301 J = 1 + JINDX
      O(J*MM) = 0 *
   DO 60 I = 1 + NUMSP
60 O(J*MM) = O(J*MM) + A(I*J)*Y(I)
      O(MM*J) = O(MM*J)
   301 CONTINUE
   O(MM*MM) = O * 50 CONTINUE
   YBAR = Y(I)
   NUMIT = NUMIT + 1
   IF (NUMIT.EQ.100) GO TO 390
   DO 5 I = 2 + NUMSP
5 YBAR = YBAR + Y(I)
   DO 7 K = 1 + JINDX
      R(I*K) = 0 *
   DO 6 I = 1 + NUMSP
6 R(I*K) = R(I*K) + A(I*J)*A(I*K)*Y(I)
   7 CONTINUE
1 COUNT = 1
   JJ = 2
   DO 28 J = JJ + JINDX
   DO 18 K = J + JINDX
      R(J*K) = 0 *
   DO 8 I = 1 + NUMSP
8 R(J*K) = R(J*K) + A(I*J)*A(I*K)*Y(I)
   18 CONTINUE
   DO 10 K = 1 + I COUNT
10 R(J*K) = R(K*J)
   I COUNT = I COUNT + 1
28 JJ = 1 + I COUNT
   DO 3011 J = 1 + MM
   R(J*MM) = O(J*MM)
3011 R(MM*J) = O(J*MM)
DO 302 J=1,MM
DO 302 K=1,MM
302 RR(J,K)=R(J,K)
PYBAR=PREF*YBAR
DO 304 J=1,JINDEX
SUMAY(J+1)=0.
DO 303 I=1,NUMSP
THIS=P*Y(I)/PYBAR
IF(THIS.LE.0.) GO TO 303
SUMAY(J+1)=SUMAY(J+1)+A(I,J)*Y(I)*(FORT(I)+ALOG(THIS))
303 CONTINUE
304 CONTINUE
SUMAY(MM+1)=0.
DO 305 I=1,NUMSP
THIS=P*Y(I)/PYBAR
IF(THIS.LE.0.) GO TO 305
SUMAY(MM+1)=SUMAY(MM+1)+Y(I)*(FORT(I)+ALOG(THIS))
305 CONTINUE
MN=1
NMAX=10
CALL SIMEQ(R,MM,SUMAY,MN,DETERM,PIVOT,NMAX,ISCALE)
DO 306 J=1,JINDEX
306 PI(J,1)=SUMAY(J+1)
U=SUMAY(MM+1)
L_MIN=1
LCOUNT=0
DO 40 I=1,NUMSP
API=0.
DO 401 J=1,JINDEX
API=API+A(I,J)*PI(J,1)
THIS=P*Y(I)/PYBAR
IF(THIS.LE.1.E-38) GO TO 402
X(I)=Y(I)*(NEGFR(I)-ALOG(THIS)+U+1.+API)
GO TO 403
402 X(I)=0.
403 IF(X(I)) 20,30,40
20 LAMB=-Y(I)/(X(I)-Y(I))
IF(LAMB.GT.0.) GO TO 21
Y(I)=0.
GO TO 50
21 LCOUNT=1
L_MIN=AMIN(1,L_MIN,LAMB)
GO TO 40
30 IF(Y(I).EQ.0.) GO TO 40
LCOUNT=1
LAMB=1
LMIN=AMIN1(LMIN>LAMB)
GO TO 40
40 CONTINUE
IF(LCOUNT.EQ.0) GO TO 51
LAMBDAT=999999*LMIN
DO 41 I=1,NUMSP
41 Y(I)=(I-LAMBDAT)*Y(I)+LAMBDAT*X(I)
GO TO 50
51 DO 52 I=1,NUMSP
IF(Y(I).EQ.0.0) GO TO 52
IF(ABS(X(I)-Y(I))/.GE.ERR OR ABS(Y(I)-Y(I))/.GE.EA) GO TO 53
52 CONTINUE
GO TO 29
53 XBAR=0.
LAMBDAT=1.
LASTCT=0
531 DO 54 I=1,NUMSP
XPRIME(I)=(I-LAMBDAT)*Y(I)+LAMBDAT*X(I)
54 XBAR=XBAR+XPRIME(I)
DFLAMB=0.
DO 541 I=1,NUMSP
THIS=P*XPRIME(I)/(PREF*XBAR)
IF(THIS.LE.0.) GO TO 541
DFLAMB=DFLAMB+(X(I)-Y(I))*(FORT(I)+ALOG(THIS))
541 CONTINUE
IF(DFLAMB.GT.0.) GO TO 56
542 DO 55 I=1,NUMSP
55 Y(I)=XPRIME(I)
GO TO 50
56 LASTCT=LASTCT+1
IF(LASTCT.EQ.4) GO TO 542
LAMBDAT=9*LAMBDAT
XBAR=0.
GO TO 531
29 XBAR=0.
ONE=0.
TWO=0.
DO 57 I=1,NUMSP
XBAR=XBAR+X(I)
ONE=ONE+X(I)*HORT(I)
MASSFR(I)=X(I)*M(I)
IF(Y(I).EQ.0.) GO TO 57
TWO=TWO+X(I)*(FORT(I)+ALOG(X(I)))
57 CONTINUE
RECIPZ=1/(MU*XBAR)
CAPU=CAPX(KP)+ALOG10(RECIPZ*273.15/T)
CHORT=MU*ONE
SOR=CHORT -MU*(XBAR*ALOG(P/(PREF*XBAR)) +TWO)
RHO=P/(XBAR*NO*XK*T)
NE=X(1)*RHO*NO
IF(X(1) GT (6*10**(-11))) GO TO 58
LOGNE=-0
GO TO 391
58 LOGNE=ALOG10(NE)
59 GO TO 391
390 WRITE(6,t98)P
GO TO 39
391 CONTINUE
DO 600 I=1,NUMSP
TEST=X(I)*P/(XBAR*PREF)
IF(TEST LT 10**(-20)) GO TO 601
PSI(I+1)=X(I)/T*(HORT(I)-FORT(I)-ALOG(TEST))
PSI(I+2)=-X(I)
GO TO 600
601 PSI(I+1)=0
PSI(I+2)=0
600 CONTINUE
DO 602 J=1,JINDEX
CON(J+1)=A(I,J)*PSI(I+1)
CON(J+2)=A(I,J)*PSI(I+2)
DO 603 I=2,NUMSP
CON(J+1)=CON(J+1)+A(I+J)*PSI(I+1)
CON(J+2)=CON(J+2)+A(I+J)*PSI(I+2)
603 CONTINUE
CON(MM+1)=PSI(I+1)
CON(MM+2)=PSI(I+2)
DO 607 I=2,NUMSP
CON(MM+1)=CON(MM+1)+PSI(I+1)
CON(MM+2)=CON(MM+2)+PSI(I+2)
NC=2
CALL SIMEQ(RR,MM,CON,NC,DETERM,IPIVOT,NMAX,ISCALE)
DO 604 J=1,MM
PI(J+1)=CON(J+1)
DO 604 PI(J+2)=CON(J+2)
DMHT=0
DO 605 I=1,NUMSP
SUMAP=A(I+1)*PI(I+1)
DO 606 J=2,JINDEX
SUMAP=SUMAP+A(I+J)*PI(J+1)
DXDT(I) = PSI(I) - X(I) * (PI(MM+1) + SUMAP) + T*HORT(I) * DXDT(I))

DHDT = DHDT + (X(I)*CSUBP(I) + T*HORT(I)*DXDT(I))

DRHODT = T*PI(MM+1) - 1

DRHODP = 1 + PI(MM+2)

CPOR = MU * DHDT

XZ = XBAR * MU

CVOR = CPOR - DRHODT ** 2 / DRHODP * XZ

XGAMMA = CPOR / CVOR

GAMMAE = XGAMMA / DRHODP

RHOR = 1 / (10 ** CAPU)

TEST = (GAMMAE / GAM * P / PREF * RHOR)

IF (TEST LE 0.) WRITE (6, 1002) CAPX(KP), CAPU, XZ, CHORT, SOR, LOGNE, DRHODT, DRHODP, CPOR, AOAO

AOAO = SQRT (TEST)

WRITE (6, 1002) CAPX(KP), CAPU, XZ, CHORT, SOR, LOGNE, DRHODT, DRHODP, CPOR

IF (NNN EQ 10) GO TO 608

GO TO 609

608 WRITE (6, 999)

NNN = 0

609 NNN = NNN + 1

ABL(KP) = AOAO * AO

SBL(KP) = SOR

HBL(KP) = CHORT * T * 31469E7 / MU

CONTINUE

CALL INTER(PN, T, NCAPX, SBL, HBL, ABL, SOR, K1)

CONTINUE

RETURN

END
SUBROUTINE INTER(PN, TM, N, SBL, HBL, ABL, SOR4, M)
DIMENSION PN(30), SBL(30), HBL(30), ABL(30),
      IAM(30), HM(30)
COMMON/BLOCK/HM, IAM, M, P3(30)
DO 5 I=1, N
5 PN(I)=ALOG10(PN(I))
   NP=1
   CALL FTLUP(SOR4, P3(M), NP, N, SBL, PN)
   NP=-1
   CALL FTLUP(P3(M), HM(M), NP, N, PN, HBL)
   CALL FTLUP(P3(M), IAM(M), NP, N, PN, ABL)
DO 45 I=1, N
45 PN(I)=10**PN(I)
   P3(M)=10**P3(M)
RETURN
END
SUBROUTINE INTEG(DU,U4)
EXTERNAL FUNC
DIMENSION AM(30), HM(30), DU(30), SU(1:30)
COMMON/BLOCK/HM, AM, M, P3(30)
DO 1 I=1:M
1 AM(I)=1./AM(I)
DU(I)=U4
DO 10 I=2:M
NN=I
CALL MGAUSS(HM(I), HM(I), NN, SU(1:1), FUNC, FOFX, 1)
10 DU(I)=U4-SU(1:1)/30.48
RETURN
END
SUBROUTINE PERNS(P1,T1,GAMMA,MU,P2,U2,NUMP)
REAL MU
DIMENSION P2(30),U2(30)
GAMM=GAMMA-1.
GAMP=GAMMA+1.
SQGRT=SQRT(GAMMA*R*T1/MU)
DO 10 I=1,NUMP
U2(I)=SQGRT*(P2(I)/P1-1.)*SQRT((2./GAMMA)/(GAMP*P2(I)/P1+GAMM))
10 U2(I)=U2(I)/30.48
RETURN
END
SUBROUTINE NORMAL(P10, T10)

P-886.5
NORMAL SHOCK PROGRAM
PROGRAMMED FOR THE IBM 7094
YIELDING SOLUTIONS FOR FLOW PARAMETERS IN ARBITRARY GAS
MIXTURES IN THE FOLLOWING SITUATIONS-
1. BEHIND NORMAL SHOCK

DIMENSION YSTO(30), SHBL(8)

EQUILIBRIUM INPUT
REAL MU
INTEGER PUN, COMPUT, REAL, EXPO
COMMON /BLOCK1/ICODE(30), F(30), CAPM(30), DHF0(30), L(30), G(30, 30),
ISMLE(30, 30), CAPLAM(30, 30), OMEG(5, 30, 30), A(10, 30), CONR, CONPRF,
2CONNO, CONH, CONK, P1, EPS1, NIT, EPS2, IC1
COMMON PN(30), TM(30), NUMT, NCAPX, SOR4, U4, P4, T4, U3(30),
1PUN, EXPO, REAL, COMPUT, P1, T1, P2(30), NUMP, USTO(30), U2(30)
COMMON ISPEC(30), JMOL(10), M, N, BETA(5), NS,
1AMC, NB, NBTA(5), MU, GAMMA, R, NAME(10)

SHOCK PROGRAM INPUT

IF (NS.EQ.1) WRITE(6, 4001) P10, T10
4001 FORMAT (1H1, 13H NORMAL SHOCK/4H P1=E15.7, 2X, 3HT1=E15.7)
P10=P10*1.01325E6
DO 100 IJZ=1, NUMP
CALL SLITET(2, KK)
GO TO (44, 45), KK
44 CALL TAPE(N, ISPEC, M, JMOL)
45 M=M
US=USTO(IJ)
AMC=0*
DO 52 I=1, NB
J=NBTA(I)
52 AMC=AMC+BETA(I)*CAPM(J)
A10=368.*SQRT(T10/AMC)
AM10=US/A10
C SET UP GUESS Y(I)

C IF(ISPEC(1) EQ 1) GO TO 48
DO 46 I = 1, N
46 YSTO(I) = 1.0E-12
GO TO 51
48 YSTO(I) = 0.
DO 49 I = 2, N
49 YSTO(I) = YSTO(I) + A(M+I)*1.0E-12
YSTO(I) = -YSTO(I)
DO 50 I = 2, N
50 CONTINUE
DO 3035 I = 1, NB
J = NBTA(I)
3035 YSTO(J) = BETA(I)/AMC
RH010 = P10/(T10/300.0)*0.00619*AMC
TP = T10*(1.0*16*(1.5*AM10**2+1.0)/AM10**2*(AM10**2-1.0))
1*(1.0-5.0*(US-5000.0)/15000.0)
RH02 = RH010*(4.0*AM10**2)/(5.0*AM10**2+2.0)

C CONVERSION
RH010 = RH010*1.0E-3
RH02 = RH02*1.0E-3
US = US*30.48
IF (T10 LE 800.0) CALL SLITE(4)
CALL ECOM(T10,P10,00Z,H0ZRT,H10,RH010,YSTO,SOR)

C STORE INITIAL P,T,RHO,AND U IN SHBL(1-4)

SHBL(1) = P10
SHBL(2) = T10
SHBL(3) = RH010
SHBL(4) = US
CALL SHOCK(TP,SHBL,H10,YSTO,RH02,US)

C UPON RETURN SHBL(5-8) CONTAINS P2,T2,RH02,AND UF

PPRINT = SHBL(5)/CONPRF
UPRINT = SHBL(8)/30.48

P2(IJ) = PPRINT
U2(IJ) = UPRINT
IF (NS NE 1) GO TO 100
WRITE (6,202) US, PPRINT, SHBL(7), 00Z, HOZRT, SOR, SHBL(6), AMC
202 FORMAT(//4H US=E15.7//9X*1HP*13X*3HRHO*14X*3H1/Z*14X*5HH/ZRT*,
        X12X*3HS/R*,
        114X*1HT*16X*2HM1//(7E17.8))
        WRITE(6,221)
221 FORMAT(//23H FINAL Y FROM ITERATION*4X*7HSPECIES//)
        WRITE(6,220) (YSTO(I),ICODE(I),I=1,N)
220 FORMAT(E17.8*10X*1A6)
100 CONTINUE
   P10=P10/1.01325E6
   RETURN
END
SUBROUTINE TAPE(N, ISPEC, J, JMOL)

COMMON /BLOCK1/ICODE(30), F(30), CAPM(30), DHFO(30), L(30), G(30), I(30), 15MLE(30, 30), CAPLAM(30, 30), OMEG(5, 30), A(10, 30), CONR, CONPRF, 2CONNO, CONH, CONK, P1, EPS1, EPS2, EPS3, EPS4, EPS5

DIMENSION BLOCK(150), LBLOCK(30), ISPEC(30), JMOL(10), 1OBL(5, 30)

READ (9) (LBLOCK(I), I = 1, 30)
DO 1 IC = 1, N
ISP = ISPEC(IC)
1 ICODE(IC) = LBLOC(K(ISP))

READ (9) (BLOCK(I), I = 1, 30)
DO 2 IC = 1, N
ISP = ISPEC(IC)
2 F(IC) = BLOCK(ISP)

READ (9) (BLOCK(I), I = 1, 30)
DO 3 IC = 1, N
ISP = ISPEC(IC)
3 CAPM(IC) = BLOCK(ISP)

READ(9)(BLOCK(I), I = 1, 30)
DO 4 IC = 1, N
ISP = ISPEC(IC)
4 DHFO(IC) = BLOCK(ISP)

READ(9)(LBLOC(K(I), I = 1, 30)
DO 5 IC = 1, N
ISP = ISPEC(IC)
5 L(IC) = LBLOC(K(ISP))

IC = 1
DO 6 I = 1, 30
READ (9) (BLOCK(IL), IL = 1, 30)
IF(ISPEC(IC) = 1) 6, 55, 6
55 DO 56 LI = 1, 30
56 G(LI, IC) = BLOCK(LI)
IC = IC + 1
6 CONTINUE

IC = 1
DO 7 I = 1, 30
READ (9) (BLOCK(IL), IL = 1, 30)
IF(ISPEC(IC)-1)7,65,7
65 DO 66 LI=1,30
66 SMLE(LI*IC)=BLOCK(LI)
   IC=IC+1
7 CONTINUE
   IC=1
   DO 12 I=1,30
      READ (9) (BLOCK(IL), IL=1,30)
      IF(ISPEC(IC)-I)12,13,12
13 DO 125 LI=1,30
125 CAPLAM(LI,IC)=BLOCK(LI)
   IC=IC+1
   12 CONTINUE
C
   IIIC=1
   DO 8 I=1,30
      READ(9) ((OBL(IC,IL)),IL=1,30)
      IF(ISPEC(IIIC)-I)8,75,8
75 DO 76 LI=1,30
   DO 76 IC=1,5
76 OMEG(IC,LI,IIIC)=OBL(IC,LI)
   IIIC=IIIC+1
   8 CONTINUE
C
   IC=1
   DO 10 I=1,30
      READ(9) (BLOCK(IJ), IJ=1,10)
      IF(ISPEC(IC)-I)10,85,10
85 DO 9 IJ=1,J
      IJM=JMOL(IJ)
9 A(IJ,IC)=BLOCK(IJM)
   IC=IC+1
10 CONTINUE
C
   CONR=8.3146938E7
   CONPRF=1.01325E6
   CONNO=6.02322E23
   CONH=6.62517E-27
   CONK=1.38044E-16
   PI=3.14159
   NIT=300
   EPS1=1.0E-7
   EPS2=0.1
   IF(ISPEC(N)-28)14,15,14
14 IC1=1
SUBROUTINE SHOCK(TGUESS,BLOCK,H10,YSTO,RH02,RHO)

THIS SUBROUTINE USES A ONE-DIMENSIONAL NEWTON-RAPHSON ITERATION
SCHEME TO FIND TEMPERATURE AND PRESSURE AT EQUILIBRIUM BEHIND
INCIDENT SHOCK. IT WILL CALL SUBROUTINE ECOM TO COMPUTE THE
EQUILIBRIUM PROPERTIES.

DIMENSION YSTO(30)
DIMENSION T(2),H(2),BLOCK(8)

LET RH02 = FIRST RHO

VEL1(AA)=C*D/AA
PRES1(AA,BB)=B+C*D**2-AA*BB**2
ENTH1(AA)=H10+(D**2)/2-(AA**2)/2.
DELT=10*
IT=3
EPS5=1*E-5
NCOUNT=1
B=BLOCK(1)
C=BLOCK(3)
D=BLOCK(4)
U2=VEL1(RH02)
P2=PRES1(RH02,U2)
H2=ENTH1(U2)

COMPUTE FIRST POINT

ITT=IT
7 T(1)=TGUESS
CALL ECOM(T(1),P2,00Z,H0ZRT,H(1),RHO,YSTO,SOR)

COMPUTE SECOND POINT

T(2)=T(1)+DELT
CALL ECOM(T(2),P2,00Z,H0ZRT,H(2),RHO,YSTO,SOR)
S=(H(2)-H(1))/(T(2)-T(1))
T(1)=T(2)

TEMPERATURE FROM FIRST ITERATION

T(2)=T(2)+(H2-H(2))/S
H(1)=H(2)
IF(T(2)<25,25,8

$IBFTC SHOCK DECK
SUBROUTINE SHOCK(TGUESS,BLOCK,H10,YSTO,RH02,RHO)
C
C    THIS SUBROUTINE USES A ONE-DIMENSIONAL NEWTON-RAPHSON ITERATION
C    SCHEME TO FIND TEMPERATURE AND PRESSURE AT EQUILIBRIUM BEHIND
C    INCIDENT SHOCK. IT WILL CALL SUBROUTINE ECOM TO COMPUTE THE
C    EQUILIBRIUM PROPERTIES.
C    DIMENSION YSTO(30)
C    DIMENSION T(2),H(2),BLOCK(8)
C
C    LET RH02 = FIRST RHO
C
C    VEL1(AA)=C*D/AA
C    PRES1(AA,BB)=B+C*D**2-AA*BB**2
C    ENTH1(AA)=H10+(D**2)/2-(AA**2)/2.
C    DELT=10*
C    IT=3
C    EPS5=1*E-5
C    NCOUNT=1
C    B=BLOCK(1)
C    C=BLOCK(3)
C    D=BLOCK(4)
C    U2=VEL1(RH02)
C    P2=PRES1(RH02,U2)
C    H2=ENTH1(U2)
C
C    COMPUTE FIRST POINT
C
C    ITT=IT
C    7 T(1)=TGUESS
C    CALL ECOM(T(1),P2,00Z,H0ZRT,H(1),RHO,YSTO,SOR)
C
C    COMPUTE SECOND POINT
C
C    T(2)=T(1)+DELT
C    CALL ECOM(T(2),P2,00Z,H0ZRT,H(2),RHO,YSTO,SOR)
C    S=(H(2)-H(1))/(T(2)-T(1))
C    T(1)=T(2)
C
C    TEMPERATURE FROM FIRST ITERATION
C
C    T(2)=T(2)+(H2-H(2))/S
C    H(1)=H(2)
C    IF(T(2)<25,25,8

APPENDIX

31
8 CALL ECOM(T(2),P2,00Z,H0ZRT,H(2),RHO,YSTO,SOR)

S IS SLOPE \( \frac{(H_2-H_1)}{(T_2-T_1)} \)

85 \( S=(H(2)-H(1))/(T(2)-T(1)) \)

T3

\( T(1)=T(2) \)

TEMPERATURE FROM SECOND ITERATION

\( T(2)=T(2)+(H_2-H(2))/S \)

\( H(1)=H(2) \)

IF \( T(2) \geq 25 \), iterate again on temperature with first pressure

\( T(2)=T(2)+(H_2-H(2))/S \)

11 ITT=ITT-1

GO TO 85

TEST RHO FOR CONVERGENCE

12 IF(ABS((RHO-RHO2)/RHO2)-EPS5)\leq 20

NON-CONVERGENCE-

COMPUTE NEW PRESSURE AND CONTINUE ITERATION ON TEMPERATURE AND PRESSURE UNTIL RHO CONVERGES

13 RHO2=RHO

U2=VEL1(RHO2)

P2=PRES1(RHO2,U2)

H2=ENTH1(U2)

NCOUNT=NCOUNT+1

\( T(1)=TLAST \)

14 CALL ECOM(T(1),P2,00Z,H0ZRT,H(1),RHO,YSTO,SOR)

145 S=SLAST

15 \( T(2)=T(1)+(H_2-H(1))/S \)
CALL ECOM(T(2)*P2,00Z,H0ZRT,H(2),RHO,YSTO,SOR)
SLAST=(H(2)-H(1))/(T(2)-T(1))
TLAST=T(2)
GO TO 12

CONVERGENCE - STORE OUTPUT

20 U2=VEL1(RHO)
    UF=US-U2
    RHO=RHO*1.E3
    BLOCK(5)=P2
    BLOCK(6)=TLAST
    BLOCK(7)=RHO
    BLOCK(8)=UF
    RETURN

TEMPERATURE ESTIMATE TOO HIGH - ADJUST

25 TGUESS=(TGUESS-T10)/2.
    GO TO 7
END
SUBROUTINE ECOM(T, PSTO, O0Z, HOZRT, H, RHO, YST0, SOR)

SUBROUTINE WHICH, GIVEN A TEMPERATURE AND PRESSURE, COMPUTES
THE THERMODYNAMIC EQUILIBRIUM PROPERTIES OF A GAS DESCRIBED BY
THE INPUT.

REAL MU
INTEGER PUN, COMPUT, REAL, EXPO
DIMENSION SMALE(30, 30), X(30), YST0(30)
DIMENSION E(30), Y(30), Q(30), CAPFI(30), R(10, 10), B(10),
3TRIPS(10), BSUM(11, 11), ABLOCK(11, 11), PTEMP(30), ZETA(30),
4ZETAPR(30), ALAM(30),
5PIVOT(11), DQINT(30), QINT(30, 30)

COMMON /BLOCK1/ICODE(30), F(30), CAPM(30), DHFO(30), L(30), G(30, 30),
1SMLE(30, 30), CAPLAM(30, 30), OMEG(5, 30, 30), A(10, 30), CONR, CONPRF,
2CONNO, CONK, CONH, PI, EPS1, NIT, EPS2, IC1
COMMOM PN(30), TM(30), NUMT, NCAPX, SOR4, U4, P4, T4, U3(30),
1PUN, EXPO, REAL, COMPUT, P1, T1, P2(30), NUMP, USTO(30), U2(30)
COMMON ISPEC(30), JMOL(10), N, N, BETA(5), NS,
1AMC, NB, NBETA(5), MU, GAMMA*R, NAME(10)

EQUIVALENCE(SMLE(1, 1), SMALE(1, 1)), (ICODE(1), CODE(1))

PI=3.14159
C=2*99793E10
NCOUNT=0
LTEST=LTEST
N2=N
DO 5 I=1, N
5 Y(I)=YST0(I)
P=PST0
34 TK=CONK*T
RT=CONR*T
346 YBAR=0.0
DO 347 I=1, N
347 YBAR=YBAR+Y(I)
DO 40 I=1, N
TEMP1=0
LEND=L(I)
DO 37 L1=1, LEND
IF(F(I))31, 35, 31
31 PROD=1.
DO 33 IC=1, IC1
IF(OMEG(I, L, I)) = 32, 33, 32
32 PROD = PROD * (1.0 - EXP(-CONH*C*OMEG(I, L, I) / TK))
33 CONTINUE
   PART = (T / (CAPLAM(L, I) * PROD)) * F(I)
   GO TO 36
35 PART = 1.
36 QINT(L, I) = PART * G(L, I) * EXP(-CONH*C*SMALE(L, I) / TK)
37 TEMP1 = TEMP1 + QINT(L, I)
   Q(I) = (SQRT(2.0 * PI / CONH / TK) / (CONH * CONNO) * CAPM(I)) ** 3 * TK / CONPRF * TEMP1
   IF(Y(I)) = 38, 38, 39
38 CAPFI(I) = 0
   GO TO 40
39 CAPFI(I) = Y(I) * (ALOG(P / CONPRF) + ALOG(Y(I)/YBAR) - ALOG(Q(I)) + DHFO(I)
   1 / RT)
40 CONTINUE
   CALL SLITET(4, JJ)
   GO TO (95, 396), JJ
396 DO 50 J = 1, M
   DO 50 K = 1, M
   R(K, J) = 0.0
   B(J) = 0.0
   DO 50 I = 1, N
   B(J) = B(J) + A(J, I) * Y(I)
50 R(K, J) = R(K, J) + A(J, I) * A(K, I) * Y(I)
   SET UP MATRIX FOR SOLUTION OF EQUATIONS
   DO 60 J = 1, M
   TEMPS(J) = 0.0
   DO 55 I = 1, N
55 TEMPS(J) = TEMPS(J) + A(J, I) * CAPFI(I)
   BSUM(J, I) = B(J) + TEMPS(J)
   CONSTANT TERMS IN BSUM BLOCK
   DO 56 K = 1, M
   K1 = K + 1
56 ABLOCK(J, K1) = R(K, J)
   PI TERMS IN ABLOCK IN COLUMNS 2 THROUGH N + 1
   DO 60 J = 1, M
   ABLOCK(J, 1) = B(J)
60 ABLOCK(J, 1) = B(J)
   (X/Y) TERMS IN FIRST COLUMN
M1=M+1
ABLOCK(M1+1)=0*0
DO 61 K=1,M1
K1=K+1
61 ABLOCK(M1,K1)=B(K)
BSUM(M1+1)=0*0
DO 62 I=1,N
62 BSUM(M1+1)=BSUM(M1+1)+CAPFI(I)
C MATINV EXPECTS AN M+1 BY M+1 MATRIX
C CALL SIMEQ(ABLOCK(1+1),M1,BSUM(1+1),1,DETERM,1PIVOT,1+1,0)
C RETURN WITH ANSWERS IN BSUM
C ZETAP=BSUM(1+1)*YBAR
ZERO=0*
NEG=0*0
DO 70 I=1,N
PTEMP(I)=0*0
DO 65 J=1,M
J1=J+1
65 PTEMP(I)=PTEMP(I)+BSUM(J1)*A(J,I)*Y(I)
ZETA(I)=-CAPFI(I)+Y(I)*BSUM(I)+PTEMP(I)
C TEST FOR NEGATIVE OR ZERO ZETA
C IF(ZETA(I))69*70
69 PIECE=Y(I)/(ZETA(I)-Y(I))
IF(PIECE)69*691
691 NEG=NEG+1
ALAM(NEG)=PIECE
GO TO 70
692 Y(I)=0
ZERO=1*
GO TO 70
695 IF(Y(I))69*70
70 CONTINUE
C FIND GREATEST NEGATIVE ZETA-Y
C IF(ZERO)700*698
698 IF(NCOUNT-NIT)699*100
699 NCOUNT=NCOUNT+1
GO TO 346
700 IF(NEG-1)78,71,73
71 ALAMPR=.999999*ALAM(I)
    GO TO 745
73 ARG1=ALAM(I)
    DO 74 I=2,NEG
72 ARG2=ALAM(I)
        ARG1=AMIN1(ARG1*ARG2)
    CONTINUE
    ALAMPR=.999999*ARG1
745 IIC=0
75 ZETAP=0
    DO 76 I=1,N
        ZETAPR(I)=Y(I)+ALAMPR*(ZETA(I)-Y(I))
    ZETAP=ZETAP+ZETAPR(I)
    DLAM=0
    DO 77 I=1,N
        IF(ZETAPR(I))77,77,765
765 DLAM=DLAM+(ZETA(I)-Y(I))*(ALOG(P/CONPRF)-ALOG(G(I))+DHF(1)/RT+ALOG(ZETAPR(I))/ZETAP))
    CONTINUE
    IF(DLAM)81,81,80
80 IF(IIC-3)805,81,81
805 IIC=IIC+1
        ALAMPR=ALAMPR*.9
    GO TO 75
78 ALAMPR=1.
    GO TO 745

CONVERGENCE TEST FOR Y(I)s

81 IF(ALAMPR-50)83,815,815
815 DO 82 I=1,N
        IF(ZETAPR(I))813,816,813
813 REL=Y(I)-ZETAPR(I)
        IF(ABS(REL)-EPS1)818,818,813
818 REL=ZETAPR(I)/Y(I)-1.
        IF(ABS(REL)-EPS2)82,82,83
816 IF(Y(I))817,82,817
817 GO TO 83
82 CONTINUE

Y(I)s CONVERGE

DO 800 I=1,N
800 Y(I)=ZETAPR(I)
GO TO 95
C
NON-CONVERGENCE OF Y(I)S
C
83 NCOUNT=NCOUNT+1
     IF(NCOUNT-NIT)84,100,100
84 DO 85 I=1,N
85 Y(I)=ZETAPR(I)
C
REPEAT WITH NEW Y(I)S AND NO. OF ITERATIONS LESS THAN NIT
C
GO TO 346
95 DO 201 I=1,N
201 X(I)=Y(I)*CAPM(I)
    YBAR=0.0
    CAPMI=0
    DO 2026 I=1,N
    YBAR=YBAR+Y(I)
2026 CAPMI=CAPMI+X(I)/CAPM(I)
    CAPMI=1.0/CAPMI
    Z=AMC/CAPMI
    ESUM=0
    DO 2029 I=1,N
    QSUM=0
    DQINT(I)=0
    LEND=L(I)
    DO 2028 L1=1,LEND
    SUM=0
    DO 2027 IC=1,IC1
    HOOTK=CONH*C*OMEG(IC,L1,I)/TK
    IF(OMEG(IC,L1,I))2000,2027,2027
2000 SUM=SUM+HOOTK/(EXP(HOOTK)-1.0)
2027 CONTINUE
    DQINT(I)=DQINT(I)+QINT(L1+1)*F(T)+QSUM+SUM+SMALE(L1+1)*CONH*C
        1/(TK**T)
2028 QSUM=QSUM+QINT(L1+1)
    E(I)=1.0/CAPM(I)*(1.5*RT+RT*T/QSUM*DQINT(I)+DHFO(I))
2029 ESUM=ESUM+X(I)*E(I)
    MOZRT=CAPMI*ESUM/RT+1.0
    H=MOZRT*CONR*Z/AMC
    TK=T*CONK
    FSUM=0
    DO 2040 I=1,N
2033 IF(Y(I))2034,2034,2035
2034 CAPFI(I)=0
GO TO 2040
2035 CAPFI(I)=Y(I)*(ALOG(P/CONPRF)+ALOG(Y(I)/YBAR)-ALOG(G(I))+DF0(I))
1/RT)
2040 FSUM=FSUM+CAPFI(I)
     SOZR=HOZRT-CAPMI*FSUM
     SOR=SOZR*Z
     RHO=P*CAPMI/RT
     U=CAPX+43429*ALOG(273*16/(Z*T))
     O0Z=1.0/Z
     DO 300  I=1+N
     YSTO(I)=Y(I)
   300 X(I)=X(I)*CAPMI/CAPM(I)
      RETURN
100 WRITE(6,5000)
5000 FORMAT(1HO,25H THIS CASE NON-CONVERGENT)
      CALL EXIT
      END
SUBROUTINE FUNC(DUM, FOFX)
COMMON/BLOCK/HM, AM, M *P3(30)
DIMENSION HM(30), AM(30)
CALL FTLUP(DUM, AA, -2, M, HM, AM)
FOFX = AA
RETURN
END
SUBROUTINE SOLUT(U3*,P3*,U2*,P2*,M*,N*,UR*,P)
DIMENSION U3(30),P3(30),U2(30),P2(30),U(2)
FUNCTION(P,P*,U*,R)=P-P*U/R
FUNCTION(P,P*,U,U*)=(P-P*)/(U-U*)
USE END POINTS FOR FIRST INTERSECTION
MR=1
NR=1
IF(P2(1) > P2(2)) NR=MR
IF(P3(1) > P3(2)) MR=MR
P31=P3(1)
P32=P3(M)
P21=P2(1)
P22=P2(N)
U21=U2(1)
U22=U2(N)
U31=U3(1)
U32=U3(M)
AA=FUNCTION(P22,P21,U22,U21)
BB=FUNCTION(P32,P31,U32,U31)
CC=FUNCTION(P21,U21-AA)
DD=FUNCTION(P31,U31+BB)
UR=(CC-DD)/(BB-AA)
PR=CC+UR*AA
CALL FTLUP(PR,U(1),NR,N,P2,U2)
CALL FTLUP(PR,U(2),MR,M,P3,U3)
IF(ABS((U(1)-U(2))/U(1))<0001)12,12,10
P31=P32
P32=PR
P21=P22
P22=PR
U31=U32
U32=U3(M)
U21=U22
U22=U2(N)
GO TO 5
P=PR
RETURN
END
REFERENCES


Figure 1.- Operating sequence of a simple shock tube.

(a) Distance-time diagram.

(b) Prior to diaphragm burst, \( t = 0 \).

(c) After diaphragm burst, \( t = t_a \).
Figure 2.— Schematic diagram of expansion tube flow cycle.
Figure 3.- Velocity $u_3$ as a function of pressure $p_3$ for real gas isentropic expansion of helium for $p_4/p_0 = 315$ and $T_4 = 15000^0$ K; velocity $u_2$ as a function of pressure $p_2$ for incident normal shock in air.
Figure 4.- Incident shock speed in air as a function of $p_1/p_0$ for initial helium driver conditions of $p_4/p_0 = 315$ and $T_4 = 15,000^\circ K$. 
Figure 5.- Shock speed as a function of initial air pressure for helium driver conditions of $p_4/p_0 = 300$ and $T_4 = 2500^\circ$ K.
Figure 6.- Shock speed as a function of $p_1/p_0$ for helium driver conditions of $p_4/p_0 = 300$ and $T_4 = 2500^\circ$ K.
Figure 7.- Expansion tube test gas speed as a function of $p_1/p_0$ for helium driver conditions of $p_4/p_0 = 300$ and $T_4 = 2500^\circ$ K.
Figure 8.— Expansion tube test gas temperature as a function of initial air pressure for helium driver conditions of $p_4/p_0 = 300$ and $T_4 = 2900^\circ$K.
Figure 9.- Expansion tube test gas pressure as a function of initial air pressure for helium driver conditions of $p_4/p_0 = 300$ and $T_4 = 2500^\circ$ K.
Figure 10.- Test gas shock speed as a function of buffer gas pressure for test gas of 90 percent N₂ and 10 percent CO₂ and helium driver conditions of \( p_4/p_0 = 100 \) and \( T_4 = 300^0 \text{K} \).
Figure 11.- Test gas shock speed as a function of buffer gas pressure for test gas of 90 percent N₂ and 10 percent CO₂ and hydrogen driver conditions of $p_d/p_0 = 100$ and $T_d = 300°$ K.