AD-A164 960

AD

US ARMY MATERIEL COMMAND

TECHNICAL REPORT BRL-TR-2703

A LUMPED PARAMETER CODE FOR REGENERATIVE LIQUID PROPELLANT GUNS

Terence P. Coffee

December 1985

DTIC
ELECTE
FEB 27 1986
S D
B

DTIC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

86 2 27 088

BLANK PAGES IN THIS DOCUMENT WERE NOT FILMED

Destroy this report when it is no longer needed.
Do not return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report BRL-TR-2703

2. GOVT ACCESSION NO.: AD-A164960

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): A Lumped Parameter Code for Regenerative Liquid Propellant Guns

5. TYPE OF REPORT & PERIOD COVERED: TR

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Terence P. Coffee

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
US Army Ballistic Research Laboratory
ATTN: SLCBR-IB
Aberdeen Proving Ground, MD 21005-5066

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1L263637D155

11. CONTROLLING OFFICE NAME AND ADDRESS:
US Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

12. REPORT DATE: December 1985

13. NUMBER OF PAGES: 158

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: n/a

16. DISTRIBUTION STATEMENT (of this Report):
Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

DTIC
ELECTE
FEB 27 1986
S D
B

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
liquid monopropellants
regenerative guns
lumped parameter model
fuel injection
droplet burning
numerical solution

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The governing ordinary differential equations for a lumped parameter model of a regenerative liquid gun are derived. A number of different options and assumptions are implemented. The equations are solved numerically using an efficient and robust computer code. The solutions are compared to a standard code for liquid propellant guns, and the strengths and weaknesses of each method are discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS.....5

LIST OF TABLES.....7

I. INTRODUCTION.....9

II. THE REGENERATIVE LIQUID PROPELLANT GUN.....10

III. GOVERNING EQUATIONS.....15

IV. VENT OPTIONS

A. VENT 1.....26
B. VENT 2.....26
C. VENT 3.....28

V. DROPLET BURNING OPTIONS

A. DROP 1.....31
B. DROP 2.....32
C. DROP 3.....40

VI. MASS AND ENERGY BALANCE.....43

VII. NUMERICAL METHOD.....46

VIII. NUMERICAL COMPARISONS.....48

REFERENCES.....57

APPENDIX A.....59

APPENDIX B.....135

GLOSSARY.....149

DISTRIBUTION LIST.....155

LIST OF ILLUSTRATIONS

Figure		Page
1	A Regenerative Liquid Propellant Gun	11
2	A Regenerative Liquid Propellant Gun with an Annular Piston	12
3	Baseline 25mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Chamber Pressures. Coffee model (line); Gough model (dot)	53
4	Baseline 25mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Projectile Velocities. Coffee model (line); Gough model (dot)	54
5	Long 25mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Chamber Pressures. Coffee model (line); Gough model (dot)	55
6	Long 25mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Projectile Velocities. Coffee model (line); Gough model (dot)	56

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1 | 23

LIST OF TABLES

Table		Page
1	Baseline Input Data for Test Problem	49
2	Comparison - Baseline Input Parameters	50
3	Comparison - Back Flow into Liquid Chamber	50
4	Comparison - Initial Gas Pressure = 10 MPa	51
5	Comparison - Long Gun Tube	52

I. INTRODUCTION

In this paper we derive the governing equations for a lumped parameter model of a regenerative liquid propellant gun and discuss the numerical implementation. Morrison et. al. have covered the background and possible applications of liquid propellant guns.[1] Here the emphasis is on the derivation of a numerical model.

For solid propellant guns, there is a wide range of models.[2] Precomputer models involved approximations that allowed an analytic, closed-form solution, or else a simple numerical solution leading to tables. The advent of computers allowed the direct numerical solution of ordinary differential equations describing the interior ballistics on a case by case basis. More recently, one-dimensional and even two-dimensional models have been developed, involving the solution of partial differential equations. All of these procedures can give accurate results. The choice of a model depends on the detail desired in the solution.

For regenerative liquid propellant guns, much less work has been done. Pagan and Izod[3] have developed an ordinary differential equation model. This code assumes a particular type of regenerative gun, and has a rather complicated model for the liquid injection into the combustion chamber. Cushman[4] has written a model that has been extensively used at General Electric. The model normally used at the BRL is due to Paul

---

[1]W.F. Morrison, J.D. Knapton, and G. Klingenberg, "Liquid Propellants for Gun Applications," Proceedings of the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April, 1983.

[2]P.G. Baer, "Practical Interior Ballistic Analysis of Guns," Progress in Astronautics and Aeronautics, Interior Ballistics of Guns, (H. Krier and M. Summerfield, ed.), Vol 66, 1979.

[3]G. Pagan and D.C.A. Izod, "Regenerative Liquid Propellant Gun Modelling," Proceedings of the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April, 1983.

[4]P.G. Cushman, "Regenerative Liquid Propellant Gun Simulation User's Manual," GE Report 84-POD-004, December, 1983.

Gough.[5] This code uses a lumped parameter representation (ordinary differential equations) for the liquid reservoir and combustion chamber. The gun tube is represented by partial differential equations. This code has a number of options, including a traveling charge projectile. Since both the design of the regenerative liquid gun and the assumptions made in the model are under development, additional options are periodically added to the code.

The purpose of this report is twofold. First we develop a complete lumped parameter code (only ordinary differential equations) to describe the regenerative liquid propellant gun. This code is not as general as that developed by Gough. However, it runs much faster, which allows parametric studies to be made easily. Also, the code is written as far as possible in a modular form. Additional options can be added with a minimum amount of effort. As new gun designs are developed, or new assumptions about the behavior in the gun are tried, these can be tested fairly easily.

II. THE REGENERATIVE LIQUID PROPELLANT GUN

Figs. 1 and 2 show regenerative liquid propellant guns with an in-line piston. Another possible design is the reverse annular gun, where the piston is wrapped around the barrel and moves in the direction of the muzzle.[5] The reverse annular gun includes an intermediate combustion chamber. While we will only consider in-line pistons, in order to be consistent with the notation used by Gough, the regions of the gun will be denoted as regions 1, 3, and 4.

Region 1 is the liquid propellant reservoir. The monopropellant is pumped into the reservoir at the beginning of the firing cycle. A primer is ignited in the combustion chamber (region 3). As the chamber is pressurized, the piston is forced back. Because of the piston area

---

[5]P.S. Gough, "A Model of the Interior Ballistics of Hybrid Liquid-Propellant Guns," Final Report, Contract DAAK11-82-C-0154, PGA-TR-83-4, September 1983.

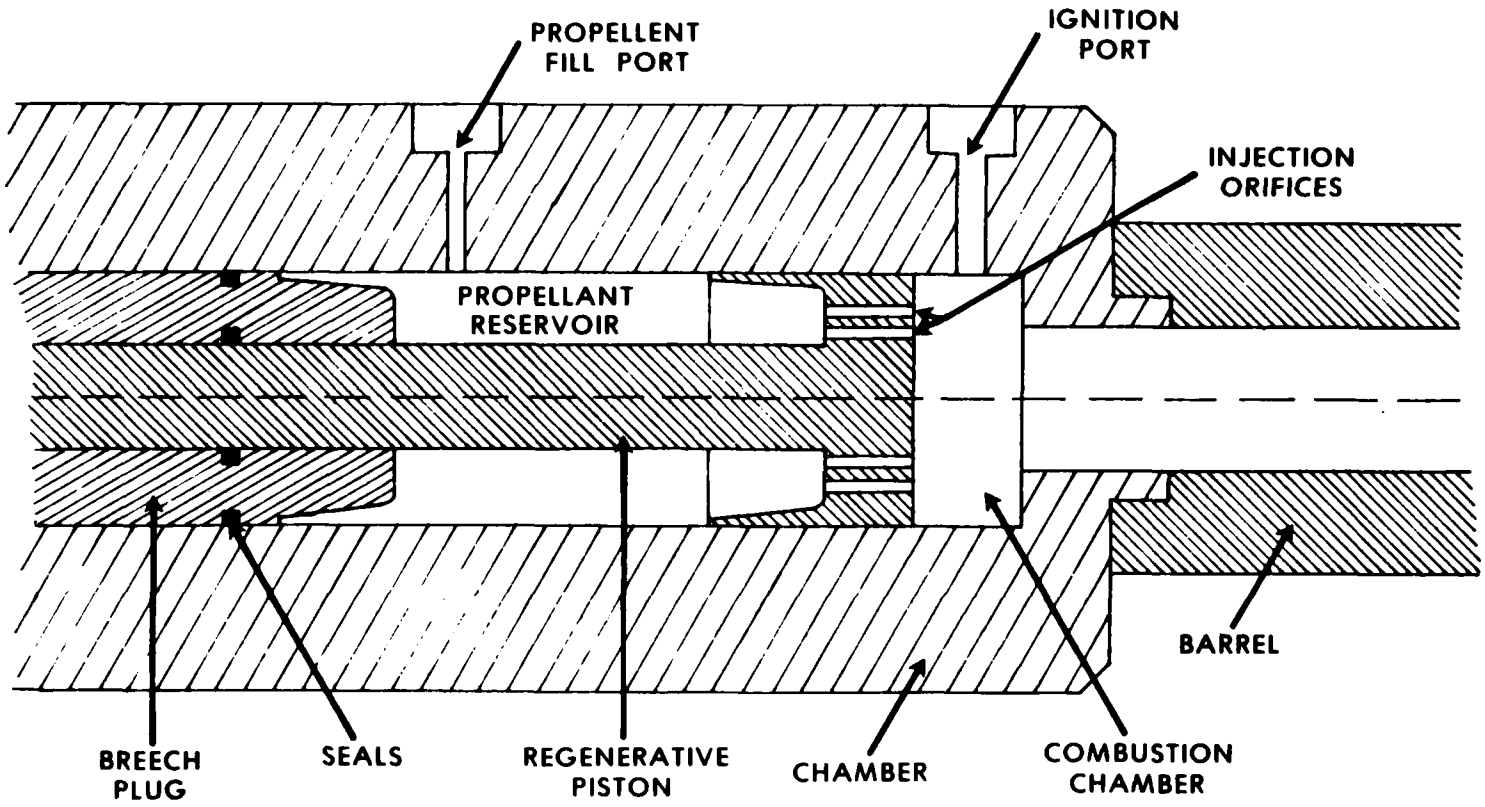

Figure 1. A Regenerative Liquid Propellant Gun

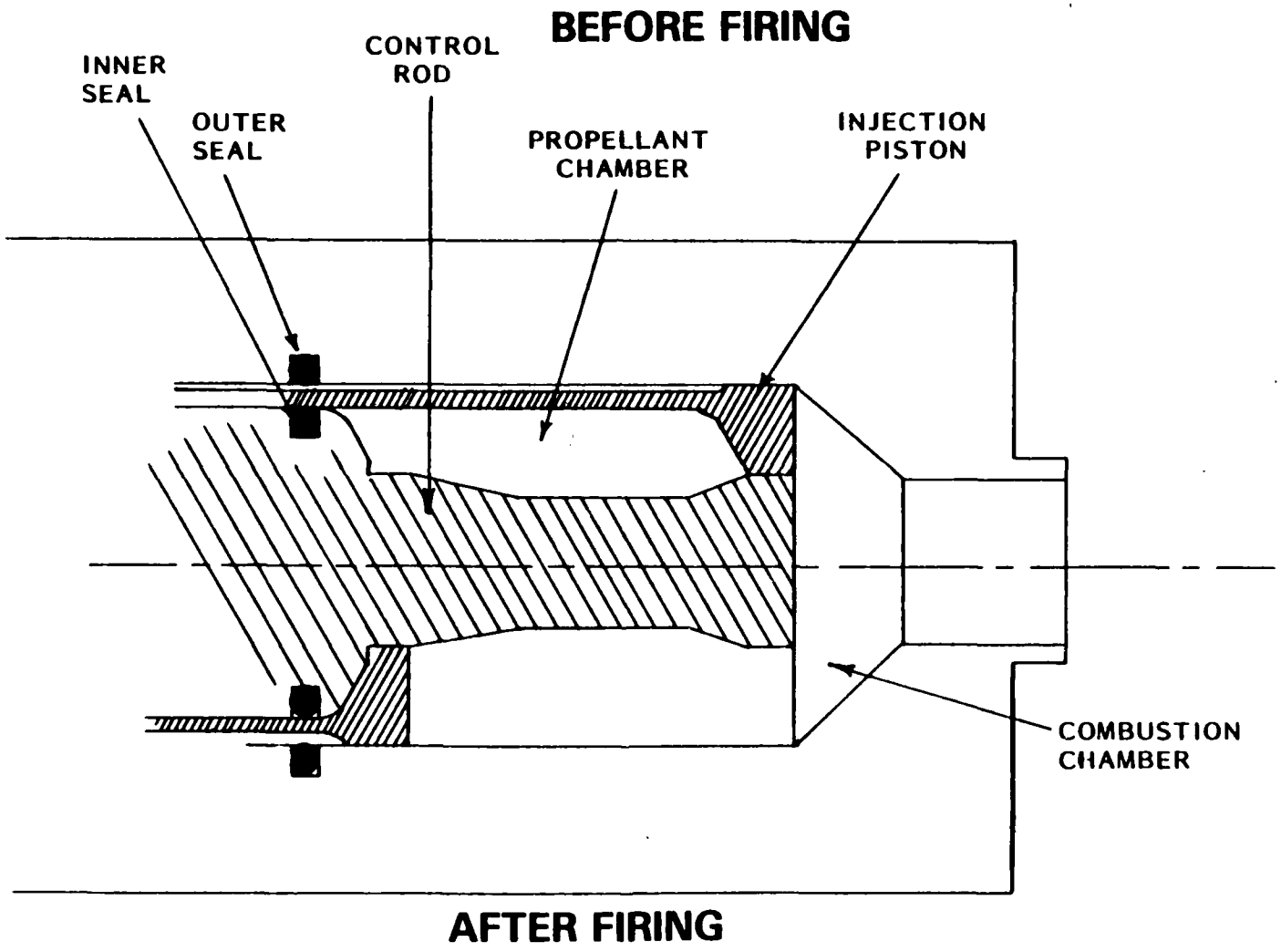

Figure 2. A Regenerative Liquid Propellant Gun with an Annular Piston

differential between the two regions, the liquid pressure is higher than the combustion chamber pressure. So liquid propellant is forced through circular holes in the piston face, and ignites and burns in the combustion chamber. Eventually, the pressure pushes the projectile down the gun tube (region 4).

In most problems involving liquids, the liquid can be considered incompressible. Because of the high pressures in a gun, this is not a good approximation, and the compressibility of the liquid propellant will be considered.

In the gun, it is possible that gas from the combustion chamber can be forced into the liquid chamber. This is most likely to occur at early times, when the primer causes the initial pressure rise in the combustion chamber. Because of the inertia of the piston, there may be a delay before the liquid pressure also rises. The modeling of bubbles in a compressible liquid is complicated, so we will assume that there is no mass flow into region 1. In the experimental fixture grease is used to block the holes so that flow into the liquid reservoir is less likely.

The orifice flow through the vents in the piston is quite complicated. Here we assume that the piston face is infinitely thin, and ignore the behavior in the orifice. The mass of the piston will be considered.

The piston design considered here was used in early test fixtures,[6] [7] but is not convenient for actual guns. Since the vent holes must be plugged before each shot, rapid fire is impossible. Other piston designs will be considered below.

---

[6] W.F. Morrison, M.J. Bulman, P.G. Baer, and C.F. Banz, "The Interior Ballistics of Regenerative Liquid Propellant Guns," 1984 JANNAF Propulsion Meeting, New Orleans, LA, CPIA Publication 390, February, 1984.

[7] P.G. Baer , C.F. Banz, I.W. May and W.F. Morrison, "A Propulsion System Comparison Study for the 120-MM Anti-Armor Cannon," 1984 JANNAF Propulsion Meeting, New Orleans, LA, CPIA Publication 390, February, 1984.

The behavior in the combustion chamber is very complicated. A primer is ignited in the chamber, leading to a gradual pressure rise. Liquid jets are forced out of the vent holes. These jets may break up into droplets because of hydrodynamic forces between the gas and the liquid or due to impact on the wall of the chamber. The droplets formed may break up further or coalesce. The propellant will eventually ignite, and may burn as individual droplets or as an envelope flame. These processes are not well understood at low pressure, and essentially no work has been done at gun pressures. Therefore, simplifying assumptions are necessary.

As a first approximation, we assume that the liquid burns instantaneously as soon as it enters the combustion chamber, releasing all its energy. The combustion chamber is treated as a homogeneous region.

The primer is assumed to be the same liquid as the propellant. Normally the combustion of the primer is not followed in detail. Instead the initial pressure in the combustion chamber is an input parameter. Then the amount of propellant needed to produce this pressure is calculated, and this is taken as the mass of the primer. The piston and projectile are not allowed to move during the burning of the primer. Later a somewhat more complicated primer option will be discussed.

There is some evidence that liquid accumulates in the combustion chamber at the beginning of the firing cycle, which changes the behavior of the pressure rise. Later we introduce finite burning rates for the liquid propellant, in an attempt to model, at least crudely, the effects of possible propellant accumulation.

Last, we have the behavior in the gun tube itself. In the Gough code, this region is modeled by one-dimensional partial differential equations. This is necessary because of the traveling charge option included in this code. We will consider this region as a lumped parameter region, using the

Lagrange pressure distribution. This has been successfully used to model the behavior of solid propellant guns.[8]

Below, we derive the governing equations for the simple lumped parameter model just described. Later, additional options will be considered.

III. GOVERNING EQUATIONS

With the above assumptions, the regenerative gun behavior can be modeled by 13 ordinary differential equations ([1] to [13]).

The equations governing the piston motion can be derived simply. Let V_1 be the volume of the fuel chamber, and A_1 be the area of the fuel side of the piston (this area includes the vent holes). Let V_{1o} be the initial volume of the liquid chamber. Let S_{ps} be the piston travel (to the left) and V_{ps} be the piston velocity. Then

$$V_1 = V_{1o} - S_{ps} A_1 \ . \tag{1}$$

Taking the derivative,

$$[1] \quad \frac{dV_1}{dt} = - V_{ps} A_1 \ . \tag{2}$$

The acceleration of the piston equals the force on the piston divided by the mass of the piston, that is

---

[8] J. Corner, Theory of the Interior Ballistics of Guns, Wiley, New York, 1950.

$$[2] \quad \frac{dV_{ps}}{dt} = \frac{g_o}{M_{ps}} [p_3(A_3 - A_v) - p_1 (A_1 - A_v)] , \tag{3}$$

where A_3 is the area of the combustion side of the piston, A_v is the area of the vent holes, p_1 is the pressure in the liquid chamber, p_3 is the pressure in the combustion chamber, and M_{ps} is the mass of the piston. The quantity $g_o = 10^7$ gm/s-cm-MPa is a conversion constant to put the acceleration in the desired units of cm^2/s. By assumption, the pressures are constant throughout the two chambers. Then the equation governing the piston travel is

$$[3] \quad \frac{dS_{ps}}{dt} = V_{ps} . \tag{4}$$

As the piston moves to the left, liquid propellant is forced through the vent holes. This is governed by Bernoulli's equation[9]

$$\frac{p_3 - p_1}{\rho_1} + \frac{v_3^2 - v_1^2}{2} = 0 \tag{5}$$

where ρ_1 is the density of the propellant in region 1, v_1 is the velocity of the fluid on the liquid side, and v_3 is the velocity of the injected fluid on the combustion chamber side of the piston. In using this equation, the flow in the orifices is assumed to be frictionless incompressible flow. The potential energy due to height differences has been left out of the equation.

---

[9] W.F. Hughes and J.A. Brighton, Fluid Dynamics, Schaum's Outline Series, McGraw-Hill, New York, 1967.

Since the liquid in region 1 is a homogeneous mixture, we take v_1 to be zero. Eq. (5) can be rewritten as

$$\rho_1 \; v_3 = \sqrt{2\rho_1 \, (p_1 - p_3)} \; . \tag{6}$$

The mass flow rate through the orifices, denoted by $\dot{m}_{13}$, equals $\rho_1 v_3 A_v$. There is usually some narrowing or constriction of the jet. This is taken into account by introducing a discharge coefficient C_D. Then

$$\begin{aligned} \dot{m}_{13} &= C_D \, A_v \sqrt{2g_o \, \rho_1 \, (p_1 - p_3)} \; , \qquad & p_1 > p_3 \; . \\ &= 0 & p_1 < p_3 \; . \end{aligned} \tag{7}$$

For short circular orifices at atmospheric pressure, the discharge coefficient should be about 0.61 to 0.63.[10] The orifices are plugged at the beginning of the firing cycle, so that gas will not be forced into region 1. So the mass flux is set to zero if $p_1 < p_3$.

Now let M_1 be the mass of the liquid in region 1. By conservation of mass,

$$\frac{dM_1}{dt} = - \dot{m}_{13} \; . \tag{8}$$

The mass is equal to the density times the volume of region 1. Rearranging eq. (8),

---

[10] W. Kaufman, Fluid Mechanics, McGraw-Hill, New York, 1963.

$$[4] \quad \frac{d\rho_1}{dt} = -\frac{\rho_1}{V_1}\frac{dV_1}{dt} - \frac{\dot{m}_{13}}{V_1} \; . \tag{9}$$

This equation becomes singular as the volume approaches zero. This problem will be discussed below in the section on the numerical solution of the equations.

The bulk modulus K of a compressible fluid is defined by[11]

$$\rho_1\left(\frac{\partial p_1}{\partial \rho_1}\right)_T = K \; , \tag{10}$$

where the subscript T indicates that the derivative is evaluated with the temperature constant. The liquid is assumed to be isothermal. The bulk modulus does vary with pressure. For lack of better information, we assume a simple linear form

$$K = K_1 + K_2 \, p_1 \; . \tag{11}$$

Combining eq. (10) and (11), and solving for p_1,

$$p_1 = [(k\rho_1)^{K_2} - K_1]/K_2 \; , \tag{12}$$

where k is a constant. Let ρ_o be the density at p=0. Usually the density is known at p = 1 atm., but compared to gun pressure ranges 1 atm. is essentially zero. Then solving for k and substituting into eq. (12)

$$p_1 = \frac{K_1}{K_2}\left[\left(\frac{\rho_1}{\rho_o}\right)^{K_2} - 1\right] \; . \tag{13}$$

---

[11] R.W. Fox and A.T. McDonald, Introduction to Fluid Mechanics, 2nd ed., John Wiley and Sons, NY, 1978.

Taking derivatives,

$$\frac{dp_1}{dt} = \frac{K_1}{\rho_1} \left(\frac{\rho_1}{\rho_o}\right)^{K_2} \frac{d\rho_1}{dt} \tag{14}$$

and combining eq. (11) and (13) this can be rewritten as

$$\frac{dp_1}{dt} = \frac{K}{\rho_1} \frac{d\rho_1}{dt} \ . \tag{15}$$

This can be expressed in terms of the speed of sound in the liquid c_1, which is given by[12]

$$c_1 = \sqrt{g_o \left(\frac{\partial p_1}{\partial \rho_1}\right)_s} \tag{16}$$

or

$$c = \sqrt{g_o \ K/\rho_1} \ . \tag{17}$$

Then eq. (15) becomes

$$[5] \quad \frac{dp_1}{dt} = \frac{c_1^{\ 2}}{g_o} \frac{d\rho_1}{dt} \ . \tag{18}$$

Now consider the combustion chamber. By analogy with eq. [1],

---

[12] W. Band, Introduction to Mathematical Physics, D. Van Nostrand Company, Princeton, NJ., 1960.

$$[6] \quad \frac{dV_3}{dt} = V_{ps} A_3 \ . \tag{19}$$

The equation for conservation of mass is

$$\frac{dM_3}{dt} = \dot{m}_{13} - \dot{m}_{34} \ , \tag{20}$$

where $\dot{m}_{34}$ is the mass flux from region 3 to region 4. It follows that

$$[7] \quad \frac{d\rho_3}{dt} = - \frac{\rho_3}{V_3} \frac{dV_3}{dt} + \frac{\dot{m}_{13} - \dot{m}_{34}}{V_3} \ . \tag{21}$$

Finally, we need an equation for the pressure in region 3. This will be based on the Noble-Abel equation of state[8]

$$p_3 = \rho_3 \, RT_3/W(1 - \rho_3 b) \ , \tag{22}$$

where T_3 is the temperature in region 3, R is the universal gas constant, W is the molecular weight of the gas, and b is the covolume. This can also be written as

$$p_3 = \rho_3(\gamma - 1) \, c_v \, T_3/(1 - \rho_3 b) \ , \tag{23}$$

where c_v is the specific heat at constant volume and γ is the specific heat ratio c_p/c_v. It will again be useful to determine the speed of sound[8]

$$c_3 = \sqrt{g_o \left(\frac{\partial p_3}{\partial \rho_3}\right)_s} \tag{24}$$

where s is the entropy. Taking the partial derivative of eq. (23), after some algebra

$$\left(\frac{\partial p_3}{\partial \rho_3}\right)_s = \frac{\gamma p_3}{\rho_3(1 - b\rho_3)} \tag{25}$$

where we have used the fact that[13]

$$\left(\frac{\partial T_3}{\partial \rho_3}\right)_s = \frac{(\gamma - 1)T_3}{\rho_3(1 - \rho_3 b)} \ . \tag{26}$$

Combining eq. (24) and (25) results in

$$c_3 = \sqrt{\frac{g_o \ \gamma p_3}{\rho_3(1 - \rho_3 b)}} \ . \tag{27}$$

The energy equation for a homogeneous gas can be written as[14]

$$\rho_3 \ c_p \ \frac{dT_3}{dt} = \left(\frac{\partial \ln \hat{V}_3}{\partial \ln T_3}\right)_{p_3} \frac{\partial \rho_3}{\partial t} + S_3 \tag{28}$$

where the spatial terms and the viscosity have been ignored. The quantity $\hat{V} = 1/\rho_3$. Using the Noble-Abel equation,

$$\left(\frac{\partial \ln \hat{V}_3}{\partial \ln T_3}\right)_{p_3} = 1 - b\rho_3 \ , \tag{29}$$

---

[13] G.N. Lewis and M. Randall, Thermodynamics, McGraw-Hill, New York, 1961.

[14] R.B. Bird and W.E. Stewart, Transport Phenomena, John Wiley and Sons, New York, 1960.

and eq. (28) becomes

$$\rho_3 \, c_p \, \frac{\partial T_3}{\partial t} = (1 - b\rho_3) \, \frac{d\rho_3}{dt} + S_3 \, . \tag{30}$$

The source term S_3 is due to the energy release when the liquid combusts. In general,

$$S_3 = - \Sigma h_i \, R_i \, M_i \tag{31}$$

where h_i is the enthalpy of the quantity i, R_i is the rate of production of the quantity i, and M_i is the mass of the quantity i. The rate of production of gas in region 3 is m_{13}/V_3, since the liquid combusts instantaneously when it enters the combustion chamber. The outflow is irrelevant since it does not cause a change in enthalpy.

Again from the Noble-Abel equation,

$$\rho_3 \, c_p \, \frac{dT_3}{dt} = \frac{\gamma}{\gamma - 1} \left[(1 - b\rho_3) \, \frac{dp_3}{dt} - \frac{p_3}{\rho_3} \, \frac{d\rho_3}{dt} \right]. \tag{32}$$

Substituting into eq. (28) and doing the algebra

$$[8] \quad \frac{dp_3}{dt} = \frac{c_3^{\,2}}{g_o} \, \frac{d\rho_3}{dt} + \frac{\dot{m}_{13}(h_1 - h_3)(\gamma - 1)}{V_3 - bM_3} \qquad \dot{m}_{34} > 0 \, . \tag{33}$$

Normally, gas will flow from the combustion chamber into the gun tube. But if the gas flow from region 3 to region 4 ($\dot{m}_{34}$) is negative, there will be an additional source term. In this case

$$[8] \quad \frac{dp_3}{dt} = \frac{c_3^{\,2}}{g_o}\frac{d\rho_3}{dt} + \frac{\dot{m}_{13}(h_1 - h_3)(\gamma - 1)}{V_3 - bM_3} \qquad \dot{m}_{34} < 0$$

$$- \frac{\dot{m}_{34}(h_4 - h_3)(\gamma - 1)}{V_3 - bM_3} \; . \tag{34}$$

The enthalpy of the liquid is given by

$$h_1 = e_1 + p_1/\rho_1 \; , \tag{35}$$

where e_1 is the chemical energy of the liquid. This is obtained from closed bomb tests. The enthalpy of the gas is given by

$$h_3 = c_v \, T_3 + p_3/\rho_3 = c_p \, T_3 + bp_3 \; . \tag{36}$$

Finally, we consider the gun tube (region 4). The volume has the standard equation

$$[9] \quad \frac{dV_4}{dt} = V_{pj} \, A_4 \; , \tag{37}$$

where V_{pj} is the velocity of the projectile and A_4 is the area of the gun tube. To model this as a lumped parameter region, we assume a Lagrange pressure distribution.[8] That is, the density is constant with respect to space. Assuming that the gas velocity at the gun tube entrance is zero, it follows that the gas velocity is a linear function of distance. Since there is a mass flow into the gun tube, this is not strictly correct. However, the derivation with a velocity at the entrance is much more complicated, and the standard Lagrange distribution is used as a first approximation. This leads to

$$p(x) = p_R + (p_R - p_{RS}) \frac{M_4}{2M_{pj}} \left[1 - \left(\frac{x}{x_R} \right)^2 \right] \tag{38}$$

where x_R is the distance from the tube entrance to the base of the projectile. Integrating over the tube length,

$$p_4 = p_R \left(1 + \frac{M_4}{3M_{pj}} \right) - p_{RS} \frac{M_4}{3M_{pj}} \tag{39}$$

where p_4 is the average pressure in the tube. This can be rearranged as

$$p_R = \frac{p_4 + p_{RS} M_4/(3M_{pj})}{1 + M_4/(3M_{pj})} \qquad p_R > p_{RS} \tag{40}$$

where p_R is the pressure at the right end of the gun tube (base of the projectile), M_4 is the mass of the gas in the tube, M_{pj} is the mass of the projectile, and p_{RS} is the resistance pressure. The latter is an input parameter, which takes into account the shot start engraving force and frictional forces between the projectile and the bore. The pressure at the gun tube entrance is

$$p_L = p_R \left(1 + \frac{M_4}{2M_{pj}} \right) - p_{RS} \frac{M_4}{2M_{pj}} \qquad p_R > p_{RS} \; . \tag{41}$$

The above two equations assume that the projectile has started to move, creating a pressure differential. If the projectile has not moved, then region 4 is treated as just an extension of region 3, that is,

$$p_4 = p_R = p_L = p_3 \; . \tag{42}$$

Given the pressure on the projectile, the acceleration equation is

$$[10] \quad \frac{dV_{pj}}{dt} = (P_R - P_{RS})\, A_4\, g_o/M_{pj} \qquad P_R > P_{RS} \tag{43}$$

$$= 0 \qquad P_R < P_{RS} \; .$$

The projectile travel S_{pj} is then given by

$$[11] \quad \frac{dS_{pj}}{dt} = V_{pj} \; . \tag{44}$$

The final two differential equations are exactly analogous to region 3. That is,

$$[12] \quad \frac{d\rho_4}{dt} = -\frac{\rho_4}{V_4}\frac{dV_4}{dt} + \frac{\dot{m}_{34}}{V_4} \tag{45}$$

where ρ_4 is the density of the gas in region 4, and

$$[13] \quad \frac{dp_4}{dt} = \frac{c_4^{\,2}}{g_o}\frac{d\rho_4}{dt} + \frac{\dot{m}_{34}(h_3 - h_4)(\gamma - 1)}{V_4 - bM_4} \qquad \dot{m}_{34} > 0 \tag{46}$$

$$= \frac{c_4^{\,2}}{g_o}\frac{d\rho_4}{dt} \qquad \dot{m}_{34} < 0$$

where $\dot{m}_{34}$ is the mass flux into region 4, given by

$$\dot{m}_{34} = C_D^{\prime}\, A_4 \sqrt{2g_o\, \rho_3(p_3 - p_4)} \qquad p_3 > p_4 \tag{47}$$

$$= -\, C_D^{\prime}\, A_4 \sqrt{2g_o\, \rho_4(P_4 - P_3)} \qquad p_3 < p_4 \; .$$

The speed of sound in region 4 is given by

$$c_4 = \sqrt{\frac{g_o \gamma p_4}{\rho_4 (1 - b\rho_4)}} \quad . \tag{48}$$

IV. VENT OPTIONS

The model described above has a simple vent option. The vent area between the liquid chamber and the combusion chamber is fixed, and equals the total area of the holes drilled in the piston. Three vent options are included in the code, where the constant vent area is a special case of option 1. For each option, the input and the vent area calculation is in a separate subroutine, and other options can be added easily.

A. VENT 1

A table of vent areas is read in as a function of piston travel. Linear interpolation is used to determine the vent area for any given piston travel. The case of constant vent area is a special case of this option.

B. VENT 2

A standard concept in interior ballistics is that of the optimum (constant pressure) gun. That is, for a given gun design, we want to know the maximum possible performance (muzzle velocity) for a given limit on the breech pressure.[2] There can also be an acceleration limit on the projectile (base pressure). For a lumped parameter solid propellant gun, this is fairly straightforward. These models usually assume a Lagrange pressure distribution or a similar fixed pressure distribution. There is a known relation between the breech pressure and the pressure at the base of the projectile. Usually, the shot start pressure is chosen as equal to the desired base pressure, so the projectile does not move until the desired pressure is reached. The burning rate for the propellant is adjusted so as

to reach the desired maximum breech (base) pressure and remain at this level until propellant burnout.

For a liquid propellant gun, the relation between the breech pressure and the base pressure is not known analytically, because of the added complication of the piston motion. Also, the burning rate cannot be controlled directly, since this depends on the mass flux into the combustion chamber. What can be controlled is the vent area.

Our scheme assumes that there is a limit both on the breech pressure and the base pressure (projectile acceleration). The input parameters are AVMIN = initial vent area = minimum vent area, AVMAX = maximum vent area, and ACS = maximum acceleration (kilo-g). The desired base pressure is

$$P_D = M_{pj}\ (.098146)\ ACS/A_4 - P_{RS} \tag{49}$$

where the number .098146 is a conversion constant. Then we define

$$RATIO = (P_D - P_R)/P_D \tag{50}$$

and let

$$A_v\ (new) = A_v\ (old)\ (1 + AK*RATIO) \tag{51}$$

where AK is a proportionality constant (usually about one). However, the new vent area is overridden by AVMIN and AXMAX.

In practice, the vent area stays at AVMIN until the desired base pressure has been achieved. The projectile then starts moving, and the base pressure falls. The vent area then increases over several time steps to maintain the base pressure. There is a delay in increasing the vent area and a further delay until the pressure increase in the combustion chamber affects the pressure distribution in the gun tube. So the base pressure

falls and then increases again. AVMAX is normally chosen by trial and error so that the breech pressure does not exceed a desired limit. Once AVMAX is reached, the breech and base pressures again fall off.

There appears to be no way to implement exactly a constant pressure algorithm for the regenerative gun without major modifications in the piston motion algorithm. The above algorithm does give an approximation to the maximum performance possible for given restrictions on the breech pressure and the base pressure.

C. VENT 3

Recent regenerative guns have been designed with an annular piston (see fig. 2). The piston has a circular hole in the center around a central rod or bolt. The fuel is stored between the bolt and the piston. The bolt remains fixed as the piston moves. The motion of the piston opens a wider annular gap, and fuel is forced between the piston and the bolt.

We assume that the bolt is made of frustrums of cones. Let x_i be the distance from the combustion chamber end of the bolt to the junction between piece i-1 and i. Then the volume of the bolt is

$$V_b = \sum_{i=2}^{N} \frac{\pi}{3} (x_i - x_{i-1})(r_i^2 + r_i r_{i-1} + r_{i-1}^2) \ . \tag{52}$$

Again we assume that the piston has zero thickness. Then the volume of the liquid is

$$V_l = \pi r_L^2 (x_N - x_1) - V_b \tag{53}$$

$$= \sum_{i=2}^{N} \pi(x_i - x_{i-1}) \left[r_L^2 - \frac{1}{3} (r_i^2 + r_i r_{i-1} + r_{i-1}^2)\right]$$

where r_L is the radius of the liquid chamber and N-1 is the number of pieces in the bolt.

The required information to determine the geometry is the x_i and the r_i values. In practice, we often want to specify the total liquid volume and the vent areas for parametric studies. So the actual input used is a table of distances and vent areas, as well as the volume and area of the liquid chamber, and the area of the hole in the piston. The radii and then the volume of the chamber can be computed. By equation (53), the volume V_l is directly proportional to the distances x_i. So the x_i are scaled to give the desired V_l. If the dimensions of an actual gun are used, where the input liquid volume matches the input dimensions, no adjustment is made.

There are several changes necessary in the previous governing equations. Let the piston travel satisfy the relation $x_{i-1} < S_{ps} < x_i$. The radius of the rod at this point can be found by linear interpolation

$$r = r_{i-1} + \frac{(S_{ps} - x_{i-1})}{(x_i - x_{i-1})} (r_i - r_{i-1}) \ . \tag{54}$$

Taking the derivative,

$$\frac{dr}{dt} = V_{ps} \frac{(r_i - r_{i-1})}{(x_i - x_{i-1})} \ . \tag{55}$$

Now the volume of the liquid reservoir is given by

$$V_l = V_{lo} - \pi r_L^2 (S_{ps} - x_l) + \frac{\pi}{3} (S_{ps} - x_{i-1})(r^2 + r\, r_{i-1} + r_{i-1}^2) \tag{56}$$

$$+ \sum_{j=2}^{i-1} \frac{\pi}{3} (x_j - x_{j-1}) (r_j^2 + r_j\, r_{j-1} + r_{j-1}^2)$$

or taking derivatives

$$\frac{dV_1}{dt} = - \pi r_L^{\,2} \frac{dS_{ps}}{dt} + \frac{\pi}{3} \frac{dS_{ps}}{dt} \left(r^2 + r\, r_{i-1} + r_{i-1}^{\,2}\right) \tag{57}$$

$$+ \frac{\pi}{3} \left(S_{ps} - x_{i-1}\right) \left(2r \frac{dr}{dt} + r_{i-1} \frac{dr}{dt}\right) .$$

Rearranging this,

$$[1] \quad \frac{dV_1}{dt} = - V_{ps} A_1 + \frac{\pi}{3} \left[V_{ps} \left(r^2 + r\, r_{i-1} + r_{i-1}^{\,2}\right) \right. \tag{58}$$

$$\left. + \left(S_{ps} - x_{i-1}\right) \left(2r + r_{i-1}\right) \frac{dr}{dt}\right] .$$

Similarly,

$$[6] \quad \frac{dV_3}{dt} = V_{ps} A_3 - \frac{\pi}{3} \left[V_{ps} \left(r^2 + r\, r_{i-1} + r_{i-1}^{\,2}\right) \right. \tag{59}$$

$$\left. + \left(S_{ps} - x_{i-1}\right) \left(2r + r_{i-1}\right) \frac{dr}{dt}\right] .$$

The piston acceleration is given by

$$[2] \quad \frac{dV_{ps}}{dt} = \frac{g_o}{M_{ps}} \left[p_3 \left(A_3 - A_h\right) - p_1 \left(A_1 - A_h\right)\right] \tag{60}$$

where A_h is the area of the hole in the piston. The acceleration on the piston no longer depends on the vent area. The vent area itself is given by

$$A_v(S_{ps}) = A_h - A_b(S_{ps}). \tag{61}$$

This is used to compute the mass flux $\dot{m}_{13}$. The other governing equations are unchanged.

V. DROPLET BURNING OPTIONS

So far we have assumed that the liquid combusts instantaneously upon entering the combustion chamber. There is some evidence, however, that liquid accumulates in the combustion chamber. To see what effect this may have on the gun performance, simple approximations for the formation and combustion of droplets in the combustion chamber are considered.

The behavior of the liquid as it enters the combustion chamber is very complicated. We will assume that the liquid instantaneously forms droplets of a fixed size. The droplets will then combust at some given rate. The initial size of the droplets is an input parameter.

In the above equations, we included the possibility that gas could flow from the gun tube back into the combustion chamber. In practice, this occurs only rarely, and the amounts involved are very small. Since the equations below are already very complicated, an additional assumption is made that the mass flux from the combustion chamber to the gun tube is positive. This eliminates some of the algebra, and has no noticeable effect on the results.

A. DROP 1

The liquid combusts instantaneously as it enters the combustion chamber, releasing all of its chemical energy. This is the option derived above.

B. DROP 2

Gough[5] has implemented a simple droplet combustion scheme. The droplets have a fixed size at all times, and combustion simply reduces the number of drops. The effect of droplet accumulation may be considered, without implementing a whole distribution of droplet sizes. We will derive the appropriate equations for this scheme in our lumped parameter model.

Let M_{L3} be the liquid mass in region 3, V_{L3} be the liquid volume, M_{G3} be the gas mass, and V_{G3} be the gas volume. Then

$$\rho_{L3} = M_{L3}/V_{L3} \tag{62}$$

and

$$\rho_{G3} = M_{G3}/V_{G3} \tag{63}$$

where ρ_{L3} is the density of the liquid and ρ_{G3} is the density of the gas. Define the porosity

$$\varepsilon_3 = V_{G3}/V_3 \ . \tag{64}$$

Then

$$\rho_3 = M_3/V_3 \tag{65}$$

$$= (1 - \varepsilon_3)\ P_{L3} + \varepsilon_3\ \rho_{G3} \ .$$

Following eq. (17), the speed of sound in the liquid is

$$c_{L3} = \sqrt{g_o\ K/\rho_{L3}} \tag{66}$$

and following eq. (27) the speed of sound in the gas is

$$c_{G3} = \sqrt{\frac{g_o \ \gamma P_3}{\rho_{G3} \ (1 - b \ \rho_{G3})}} \ . \tag{67}$$

The liquid is still considered to be isothermal, so the internal energy does not change. The heat transfer to the liquid is ignored because of the complexity of including this effect. The enthalpy of the liquid is

$$h_{L3} = e_1 + P_3/\rho_{L3} \tag{68}$$

where e_1 is the chemical energy of the liquid. Let $\dot{m}_3$ be the rate at which the liquid in region 3 is combusting and forming the final gas products.

The thirteen equations derived in Section III are the same except for the energy equations [8] and [13]. To derive the new equations, we need some preliminary results. Consider the mass conservation equations

$$\frac{dM_{L3}}{dt} = \frac{d}{dt} \ [\rho_{L3}(1 - \varepsilon_3) \ V_3]$$

$$= \dot{m}_{13} - \frac{M_{L3}}{M_3} \ \dot{m}_{34} - \dot{m}_3 \tag{69}$$

and

$$\frac{dM_{G3}}{dt} = \frac{d}{dt} \ [\rho_{G3} \ \varepsilon_3 \ V_3] \tag{70}$$

$$= \dot{m}_3 - \frac{M_{G3}}{M_3} \ \dot{m}_{34} \ .$$

Expanding the equations,

$$\frac{d\rho_{L3}}{dt}(1-\varepsilon_3)V_3 - \rho_{L3}V_3\frac{d\varepsilon_3}{dt} + \rho_{L3}(1-\varepsilon_3)\frac{dV_3}{dt} \tag{71}$$

$$= \dot{m}_{13} - \frac{M_{L3}}{M_3}\dot{m}_{34} - \dot{m}_3$$

$$\frac{d\rho_{G3}}{dt}\varepsilon_3 V_3 + \rho_{G3}V_3\frac{d\varepsilon_3}{dt} + \rho_{G3}\varepsilon_3\frac{dV_3}{dt} \tag{72}$$

$$= \dot{m}_3 - \frac{M_{G3}}{M_3}\dot{m}_{34} .$$

Combining the two equations,

$$\frac{d\rho_{L3}}{dt}\frac{(1-\varepsilon_3)}{\rho_{L3}} + \frac{d\rho_{G3}}{dt}\frac{\varepsilon_3}{\rho_{G3}} + \frac{1}{V_3}\frac{dV_3}{dt} \tag{73}$$

$$= \frac{1}{\rho_{L3}V_3}\left[\dot{m}_{13} - \frac{M_{L3}}{M_3}\dot{m}_{34} - \dot{m}_3\right] + \frac{1}{\rho_{G3}V_3}\left[\dot{m}_3 - \frac{M_{G3}}{M_3}\dot{m}_{34}\right] .$$

Now consider the pressure equations. Assume that the gas and liquid pressures in the region are identical. The liquid pressure equation, following eq. (18), is

$$\frac{dp_3}{dt} = \frac{c_{L3}^{\ 2}}{g_o}\frac{d\rho_{L3}}{dt} \tag{74}$$

and the gas phase equation, following eq. (33), is

$$\frac{dp_3}{dt} = \frac{c_{G3}^{\ 2}}{g_o}\frac{d\rho_{G3}}{dt} + \frac{\dot{m}_3(h_{L3} - h_{G3})(\gamma - 1)}{V_{G3} - bM_{G3}} \ . \tag{75}$$

Note the gas phase equation is no longer affected directly by the liquid injection into the chamber, but by the rate at which the liquid is turned into gas. Now we define the speed of sound in the mixture by the equation

$$\frac{1}{\rho_3 c_3^{\ 2}} = \frac{\varepsilon_3}{\rho_{G3} c_{G3}^{\ 2}} + \frac{(1 - \varepsilon_3)}{\rho_{L3} c_{L3}^{\ 2}} \tag{76}$$

or

$$c_3 = \sqrt{\frac{1}{\rho_3}\left[\frac{1}{\varepsilon_3/(\rho_{G3}\ c_{G3}^{\ 2}) + (1 - \varepsilon_3)/(\rho_{L3}\ c_{L3}^{\ 2})}\right]} \ . \tag{77}$$

Combining eq. (74) and (75), and using eq. (76), the result is

$$[8] \quad \frac{dp_3}{dt} = \frac{\rho_3}{g_o}\frac{c_3^{\ 2}}{V_3}\left\{-\frac{dV_3}{dt} + \frac{\dot{m}_{13}}{\rho_{L3}}\right. \tag{78}$$

$$\left. + \frac{\dot{m}_3}{\rho_{G3}}\left[1 - \frac{\rho_{G3}}{\rho_{L3}} + \frac{g_o\ (h_{L3} - h_{G3})(\gamma - 1)}{c_{G3}^{\ 2}\ (1 - b\rho_{G3})}\right] - \frac{\dot{m}_{34}}{\rho_3}\right\} .$$

Now consider region 4. The mass conservation equations are

$$\frac{dM_{L4}}{dt} = \frac{M_{L3}}{M_3}\dot{m}_{34} - \dot{m}_4 \tag{79}$$

$$\frac{dM_{G4}}{dt} = \frac{M_{G3}}{M_3} \dot{m}_{34} + \dot{m}_4 \tag{80}$$

or expanding as before

$$\frac{d\rho_{L4}}{dt} (1 - \varepsilon_4) V_4 - \rho_{L4} V_4 \frac{d\varepsilon_4}{dt} \tag{81}$$

$$+ \rho_{L4} (1 - \varepsilon_4) \frac{dV_4}{dt} = \frac{M_{L3}}{M_3} \dot{m}_{34} - \dot{m}_4$$

$$\frac{d\rho_{G4}}{dt} \varepsilon_4 V_4 + \rho_{G4} V_4 \frac{d\varepsilon_4}{dt} + \rho_{G4} \varepsilon_4 \frac{dV_4}{dt} = \frac{M_{G3}}{M_3} \dot{m}_{34} + \dot{m}_4 \ . \tag{82}$$

Again we assume that the gas and liquid pressures in the region are identical. Then the pressure equations are

$$\frac{dp_4}{dt} = \frac{c_{L4}^{\ 2}}{g_o} \frac{d\rho_{L4}}{dt} \tag{83}$$

and

$$\frac{dp_4}{dt} = \frac{c_{G4}^{\ 2}}{g_o} \frac{d\rho_{G4}}{dt} + \frac{\dot{m}_4 (h_{L4} - h_{G4})(\gamma - 1)}{V_{G4} - bM_{G4}} \tag{84}$$

$$+ \frac{M_{G3}}{M_3} \frac{\dot{m}_{34} (h_{G3} - h_{G4})(\gamma - 1)}{V_{G4} - bM_{G4}} \ .$$

Note that the mixture in region 3 is assumed to be homogeneous, and the fluid that flows into region 4 has the same porosity as the fluid in region 3. In general, region 4 has a different porosity. So the gas phase

pressure equation for region 4 has to include the effect of that portion of the mixture in region 3 that is gas. Combining the equations as before

$$[13] \quad \frac{dp_4}{dt} = \frac{\rho_4 \, c_4^{\,2}}{g_o \, V_4} \left\{ - \frac{dV_4}{dt} + \frac{\dot{m}_4}{\rho_{G4}} \left[1 - \frac{\rho_{G4}}{\rho_{L4}} + \frac{g_o (h_{L4} - h_{G4})(\gamma - 1)}{c_{G4}^{\,2} \, (1 - b\rho_{G4})} \right] \right. \tag{85}$$

$$\left. + \frac{\dot{m}_{34}}{\rho_{G4} M_3} \left[M_{G3} + M_{L3} \frac{\rho_{G4}}{\rho_{L4}} + \frac{g_o \, M_{G3} \, (h_{L3} - h_{L4})(\gamma - 1)}{c_{G4}^{\,2} \, (1 - b\rho_{G4})} \right] \right\} .$$

To complete the system, equations are required to determine the liquid density and mass in the regions. The corresponding gas quantities can be easily derived. The liquid pressure equations (74) and (83) are rewritten as

$$[14] \quad \frac{d\rho_{L3}}{dt} = \frac{g_o}{c_{L3}^{\,2}} \frac{dp_3}{dt} \tag{86}$$

and

$$[16] \quad \frac{d\rho_{L3}}{dt} = \frac{g_o}{c_{L4}^{\,2}} \frac{dp_4}{dt} \, . \tag{87}$$

The mass conservation equations for the liquid are written as

$$[15] \quad \frac{dM_{L3}}{dt} = \dot{m}_{13} - \frac{M_{L3}}{M_3} \dot{m}_{34} - \dot{m}_3 \tag{88}$$

and

$$[17] \quad \frac{dM_{L4}}{dt} = \frac{M_{L3}}{M_3} \dot{m}_{34} - \dot{m}_4 \ . \tag{89}$$

To close the system, we need information on the rate at which the liquid droplets are combusting. The rate of surface regression is assumed to be of the form

$$\text{rate of surface regression} = A\ p^B. \tag{90}$$

Liquid propellant burning rates have been measured by McBratney.[15 16] The rate of burning

$$\dot{m}_3 = \rho_{L3}\ S\ A\ p_3^{\ B}\ , \tag{91}$$

where S is the total surface area of the droplets in a region. All the droplets are assumed to have a constant diameter d.

The surface area of a single drop is given by

$$S_D = \pi d^2\ . \tag{92}$$

and the volume of a single drop by

$$V_D = \frac{\pi}{6}\ d^3\ . \tag{93}$$

---

[15] W.F. McBratney, "Windowed Chamber Investigation of the Burning Rate of Liquid Monopropellants for Guns," ARBRL-MR-03018, April 1980.

[16] W.F. McBratney, "Burning Rate Data, LPG 1845," ARBRL-MR-03128, August 1981.

If N_D is the total number of drops in region 3, then

$$N_D = V_{L3}/V_D \ . \tag{94}$$

So the total surface area in the region

$$S = 6V_{L3}/d \tag{95}$$

and eq. (91) can be written as

$$\dot{m}_3 = \rho_{L3} \ V_{L3} \ \frac{6}{d} \ Ap_3{}^B \tag{96}$$

or

$$\dot{m}_3 = M_{L3} \ \frac{6}{d} \ Ap_3{}^B \ . \tag{97}$$

For region 4, the pressure is no longer constant over the region, but follows the Lagrange pressure distribution. But this would be quite complicated to keep track of, and very little liquid goes into region 4 for most problems. So the combustion rate is assumed to depend on the average pressure P_4, and

$$\dot{m}_4 = M_{L4} \ \frac{6}{d} \ Ap_4{}^B \ . \tag{98}$$

C. DROP 3

The above representation assumes that all the droplets remain the same size at all locations and times. In practice, some distribution of droplets will exist as a function of space and time. In this section, we present a slight generalization of the above model.

The drops still form instantaneously as the fluid enters the combustion chamber, and all the droplets are initially the same size. However, the droplets will shrink as they burn. They will be represented by an approximation to a probability distribution. One advantage is that different probability distributions can be easily implemented using the formalism developed below. So alternate theories of droplet formation and combustion can easily be added.

A standard procedure is to define a probability distribution function $f(r)$ = the probable number of droplets with radius between r and $r + dr$.[17] For our purposes it is more convenient to use the mass of the droplets rather than the number of droplets. So define $M(r)$ = the mass of the droplets in region 3 with diameter between r and $r + dr$. Since this is a lumped parameter region, there is no space dependency. Then let $\dot{m}(r)$ be the rate of consumption of drops with radius r, or

$$\dot{m}(r) = M(r) \frac{3}{r} A p_3^{\,B} \; . \tag{99}$$

Then we have the partial differential equation

$$\frac{\partial M(r)}{\partial t} = - \frac{M(r)}{M_3} \dot{m}_{34} - \dot{m}(r) + \frac{\partial M(r)}{\partial r} \frac{\partial r}{\partial t} \; . \tag{100}$$

---

[17] F.A. Williams, "Progress in Spray-Combustion Analysis," in The Eighth International Combustion Symposium, Williams and Wilkins Co., Baltimore, MD, 1962.

The first term is the loss term to region 4, the second term is the loss term to the gas phase, and the last term is the loss to smaller drops. There is also an input boundary condition for r = initial radius of the droplets and an output boundary condition at r = 0.

Rather then solve the above partial differential equation, the distribution is discretized to obtain a set of ordinary differential equations. To be consistent with the previous notation, the diameter rather than the radius of the droplets will be used. More or less arbitrarily, the distribution is split into ten classes. Let d_{init} = the initial diameter of the droplets and $d_{inc} = d_{init}/10$ be the difference in diameter of the ten droplet classes. Then define $M_{L3}(i)$ = the mass of the droplets with diameter between $d_{init} - (i)d_{inc}$ and $d_{init} - (i+1)d_{inc}$. Then let $d_i = (1.05 - 0.1i)d_{init}$ be the working diameter for each class of droplets. That is, for purposes of obtaining burning rates, assume that this is the diameter of all droplets in the class. Then

$$\dot{m}_3(i) = M_{L3}(i)\,\frac{6}{d_i}\,Ap_3{}^B \tag{101}$$

is the mass burning rate for each class, and

$$\dot{m}_3 = \sum_{i=1}^{10} \dot{m}_3(i) \tag{102}$$

is the total mass burning rate.

To derive the governing ordinary differential equations, assume that in any class, the mass is evenly distributed among the possible diameters. This will contradict the assumption above that all the droplets have an average diameter. But in the limit of infinitely many classes, it will be equivalent. Looking at the first class,

$$\Delta M_{L3}(1) = \dot{m}_{13}\Delta t - \frac{M_{L3}(1)}{M_3}\dot{m}_{34}\Delta t - \dot{m}_3(1)\Delta t - M_{L3}(1)\Delta d/d_{inc} \qquad (103)$$

or the change in mass of class 1 = influx into region 3 - efflux to region 4 - loss to the gas phase - loss to the next smaller class. Since we assume that the mass is equally distributed over the possible diameters, $\Delta d/d_{inc}$ of the mass will shrink into the next smaller class. Taking the limit

$$[18] \quad \frac{dM_{L3}(1)}{dt} = \dot{m}_{13} - \frac{M_{L3}(1)}{M_3}\dot{m}_{34} - \dot{m}_3(1) - \frac{M_{L3}(1)}{d_{inc}} 2Ap_3{}^B. \qquad (104)$$

Note that the rate of change of the diameter = twice the surface regression rate = $2Ap_3{}^B$. Since we assume that all the droplets initially start in class one, all the influx from region 1 goes into class 1. For the other classes, the equations are

$$[19\text{-}27] \quad \frac{dM_{L3}(i)}{dt} = -\frac{M_{L3}(i)}{M_3}\dot{m}_{34} - \dot{m}_3(i) \qquad (105)$$

$$+ \frac{M_{L3}(i-1) - M_{L3}(i)}{d_{inc}} 2Ap_3{}^B, \; i = 2 \ldots 10.$$

All the earlier equations [1] to [17] from the above section are the same, except for the new definition of $\dot{m}_3$ and the corresponding definition of $\dot{m}_4$. One of the equations in (105) is in fact redundant, since $M_{L3} = M_{L3}(1) + \ldots M_{L3}(10)$. The extra equation is included for the sake of simplicity.

The equations for region 4 are similar, except that there is no source term from region 1. The loss terms for region 3 are the source terms for

region 4. That is,

$$[28\text{-}37] \quad \frac{dM_{L4}(i)}{dt} = \frac{M_{L3}(i)}{M_3}\,\dot{m}_{34} - \dot{m}_4(i) \tag{106}$$

$$- \frac{M_{L4}(i)}{d_{inc}}\,2Ap_4{}^B \quad i = 1 \ldots 10.$$

VI. MASS AND ENERGY BALANCE

As a check on the equations derived, both mass and energy should be conserved. The mass balance is straightforward.

$$M_T = M_1 + M_3 + M_4 \tag{107}$$

where M_T is the total mass of the propellant and primer. This should be constant throughout the integration.

The energy balance is more complicated. The energy loss through friction is quite small, and is ignored. The code presently does not include heat loss or air resistance. The liquid is considered to be an isothermal fluid, and its internal energy e is just the chemical energy of the propellant. The total energy of the liquid is

$$E_1 = e_1(M_1 + M_{L3} + M_{L4}) \ . \tag{108}$$

The internal energy of the gas in region 3 is given by

$$e_3 = c_v\,T_3 \tag{109}$$

and the total gas energy is

$$E_3 = c_v T_3 M_{G3} \cdot \tag{110}$$

The average internal energy of the gas in region 4 is given by

$$e_4 = \frac{1}{x_R}\int_o^{x_R} c_v T_4 \, dx \tag{111}$$

or

$$e_4 = \frac{1}{x_R}\int_o^{x_R} \frac{p(x)(1 - b\rho_4)}{\rho_4(\gamma - 1)} \, dx \tag{112}$$

by the Noble-Abel equation. Using eq. (38) for the pressure function, this can be integrated to yield

$$e_4 = \frac{(1 - b\rho_4)}{\rho_4(\gamma - 1)} p_4 \tag{113}$$

and

$$E_4 = e_4 M_{G4} \ . \tag{114}$$

The kinetic energy of the piston is given by

$$EK_{ps} = 0.5 M_{ps} V_{ps}^2/g_o \tag{115}$$

and the kinetic energy of the projectile by

$$EK_{pj} = 0.5 M_{pj} V_{pj}^2/g_o. \tag{116}$$

By our assumptions, only the fluid in region 4 is moving. The kinetic energy of the gas is

$$EK_{G4} = \int_o^{x_R} 0.5\ \rho_{G4}\ A_4\ v^2/g_o\ dx\ . \tag{117}$$

We ignore drag and assume that the liquid has the same velocity as the gas. Then the kinetic energy of the liquid is

$$EK_{L4} = \int_o^{x_R} 0.5\ \rho_{L4}\ A_4\ v^2/g_o\ dx\ . \tag{118}$$

The Lagrange pressure distribution implies a linear velocity profile with the gas velocity at the tube entrance equal to zero. This implies that

$$v = V_{pj}\ x/x_R\ . \tag{119}$$

Substituting and integrating

$$EK_{G4} = 0.5\ M_{G4}\ V_{pj}^{\ 2}\ x_R/(3g_o) \tag{120}$$

and

$$EK_{L4} = 0.5\ M_{L4}\ V_{pj}^{\ 2}\ x_R/(3g_o)\ . \tag{121}$$

Then the total energy of the system is

$$E_T = E_1 + E_3 + E_4 + EK_{ps} + EK_{pj} + EK_{G4} + EK_{L4}. \tag{122}$$

This should be constant throughout the integration.

VII. NUMERICAL METHOD

The ordinary differential equations derived above are solved using the code EPISODE.[18] This is a robust and efficient code for the solution of ordinary differential equations. EPISODE proper consists of the subroutines DRIVE to SING (see Appendix A).

Only two changes have been made in EPISODE proper. The first concerns the error control. Originally this could be either a relative or an absolute error control. A relative error control can be wasteful if some of the quantities being integrated become negligibly small. However, an absolute error control is inaccurate if the quantities integrated vary widely in magnitude.

A solution to this problem was developed during work on integrating chemical kinetics networks, where the concentrations of species differ by many orders of magnitude.[19] This is a semi-relative error control. If the quantities are above some cutoff value (SREC) a relative error control is used. If they are below SREC, an absolute error control is used. While an absolute error control would be adequate for the simpler equations now being considered, the semi-relative error control has been left in the code. SREC has normally been chosen as 10^{-6}.

The other change is at the end of the routine TSTEP, which actually takes the integration steps. Several options have been included here. First, diagnostic printouts of the quantities of interest may be made after

---

[18] A.C. Hindmarsh and G.D. Byrne, "EPISODE: An Effective Package for the Integration of Systems of Ordinary Differential Equations," UCID-30112-Rev.1, Lawrence Livermore Laboratory, 1977.

[19] T.P. Coffee, "A Computer Code for the Solution of the Equations Governing a Laminar, Premixed, One-Dimensional Flame," ARBRL-MR-03165, April 1982.

each time step (KWRITE=1). Second, if the optimum gun option is being used (VENT 2), the vent area is updated at the end of each time step. Finally, some information about regions 1 and 4 must be included. Eq. [4] for the liquid density becomes singular as V_1 approaches zero. So when V_1 becomes some fraction VPER of the initial volume, the region is closed. That is, region 1 is ignored in the integration from then on. This will lead to a small error, due to the liquid propellant left in region 1 when the region is closed. However, VPER is an input parameter, and by trial and error can be chosen small enough that the error is negligible. If V_1 ever becomes negative, the step is rejected, and the integration starts again with a smaller time step. There is a similar problem with region 4, since the projectile may start at the throat, and V_4 will be zero. So region 4 starts out as closed. When the projectile is at least .01 cm. down the gun tube, the region is opened. Before this, region 4 is considered as an extension of region 3.

There are actually two integration routines included in EPISODE. The first is the Adams method, suitable for non-stiff problems (METH=1). The second is the backward differentiation method, suitable for stiff problems (METH=2). We have normally used METH=2. However, the governing equations are not very stiff, and METH=1 can also be used. The run times are about equivalent for the two methods.

There are also several methods for the solution of the nonlinear system of equations required at each step of the integration. We have used the Newton method with an internally computed finite difference approximation to the Jacobian. So the only subroutine that needs to be supplied by the user is DIFFUN. Given a set of values for the unknowns, DIFFUN computes the time derivatives of the unknowns.

Subroutine DIFFUN in this code only calls one of three subroutines (FDROP1, FDROP2, or FDROP3). These routines correspond to the instantaneous burning option, the fixed drop size option, and the shrinking drop size option. Since the governing equations are different, it is easier to set these up as separate subroutines.

The rest of the routines have to do primarily with input and output. The main routine RLGD read in the input data, sets the initial conditions, calls the integrator, and writes the main output file.

To make it easier to add or change options, some of the input is in separate subroutines. For instance, subroutines VENT1, VENT2, and VENT3 read in the required data for the three vent options. These routines also compute the vent area for a given piston travel. So if a new vent option is desired, it is only necessary to add a new subroutine and change the few lines that call the VENT subroutines.

Similarly, DROP2 and DROP3 read in the input parameters for the droplet options. PISRES reads in the piston resistance as a function of piston travel. DIS1 and DIS2 read in the discharge coefficients. These can be either a function of piston travel or a function of liquid pressure. PROJRES reads in the projectile resistance pressure as a function of projectile travel. Subroutine PRIM2 is a simple primer option. That is, the primer may be included as droplets in region 3 to follow the initial behavior of the system in more detail.

Subroutines CAPTION and OUT control the output files. Subroutine OUTGRA creates a graphics file, so that any of the quantities computed may later be graphed.

VIII. NUMERICAL COMPARISONS

As a check, we will compare our code with the numerical code developed by Gough.[5] The sample problem used is discussed in ref. [6]. This is a 25 mm. gun with a constant vent area. The input data required is reproduced in Table 1.

Table 1. Baseline Input Data for Test Problems.

fuel side area of piston	13.00	cm^2
combustion side area of piston	15.48	cm^2
gun tube area	4.91	cm^2
initial fuel reservoir volume	96.00	cm^3
initial combustion chamber volume	35.40	cm^3
initial projectile offset in tube	1.27	cm
projectile travel	213.36	cm
projectile weight	194.40	gm
piston weight	480.82	gm
bore discharge coefficient	.95	
piston discharge coefficient	.58	
shot start pressure	36.00	MPa
liquid density (1 atm.)	1.23	gm/cm
bulk modulus	2413.10	MPa
liquid chemical energy	3330.90	joules/gm
ratio of specific heats	1.26	
molecular weight of combustion gases	19.01	gm/mole
covolume	1.26	cm^3/gm
initial liquid pressure	.10	MPa
initial gas pressure	81.00	MPa

The answer given here for the Gough code are slightly different than those reported in ref. 6. The Gough code controls the integration time step size based on the grid in the gun tube. The time step must be small enough not to violate the Courant-Friedrich-Levy condition.[20] The maximum number of grid points and the minimum grid spacing in the gun tube are input conditions. A smaller grid size than was used in ref. 6 was required to obtain convergence.

---

[20] P.J. Roache, Computational Fluid Dynamics, Hermosa Publishers, 1972.

The results for these computations are given in Table 2. The agreement between the two codes is not very good.

Table 2. Comparison - Baseline Input Parameters.

	Gough	Coffee
Muzzle velocity (m/s)	1097	1161
Max. liquid pressure (MPa)	289	351
Max. combustion pressure (MPa)	231	277
Time until projectile exit (ms)	3.7	3.2
Computer time (s)	20.6	1.2

This can be partially explained by considering the way the Gough code computes the mass flux between the liquid chamber and the combustion chamber. If the combustion chamber pressure is higher, gas can flow into the liquid chamber. Since the liquid chamber is one phase, the code converts this into liquid, which defies conventional physics. The new code normally does not allow flow into the liquid chamber. However, it can be run with this option. The results are in Table 3.

Table 3. Comparison - Back Flow into Liquid Chamber.

	Gough	Coffee
Muzzle velocity (m/s)	1097	1140
Max. liquid pressure (MPa)	289	316
Max. combustion pressure (MPa)	231	251
Time until projectile exit (ms)	3.7	3.4
Computer time (s)	20.6	0.7

The agreement is better, but there are still noticeable differences. Another possible cause for error is the time step control in the Gough code, which is based solely on the conditions in the gun tube. However, in the present problem we are starting with a large pressure difference between the combustion chamber and the liquid reservoir. The time step must be reduced to resolve the interaction between these two chambers and the piston movement.

To check this hypothesis, the initial pressure conditions were changed to initial liquid pressure = 9 MPa and initial gas pressure = 10 MPa. This is less of a strain on the error control. As Table 3 shows, the two codes now give almost identical answers.

Table 4. Comparison - Initial Gas Pressure = 10 MPa.

	Gough	Coffee
Muzzle velocity (m/s)	1114	1133
Max. liquid pressure (MPa)	311	310
Max. combustion pressure (MPa)	247	247
Time until projectile exit (ms)	4.2	4.2
Computer time (s)	23.5	2.5

The only noticeable difference is in the muzzle velocity. The difference is in the expected direction. That is, the new code assumes a Lagrange pressure distribution in the gun tube. So any changes in pressure in the combustion chamber instantaneously effect the projectile, no matter how far down the gun tube it is located. The Gough code uses partial differential equations to describe the gun tube, and actually follows the pressure waves as they propagate. So there is a delay between the pressure rise in the combustion chamber and the corresponding pressure rise at the base of the projectile. This results in a slightly lower muzzle velocity.

Fig. 3 compares the combustion chamber pressures as a function of time, and fig. 4 compares the projectile velocities. These curves confirm the above analysis. The agreement is excellent at early times. But as the projectile moves down the gun tube, differences appear and become greater with time.

It is useful to know if these differences can become important. So we took the previous problem and increased the liquid reservoir volume to 300 cm and the projectile travel to 1000 cm. Results are given in Table 5 and in figs. 5 and 6. There is in fact a fairly large difference in muzzle velocities.

Table 5. Comparison - Long Gun Tube.

	Gough	Coffee
Muzzle velocity (m/s)	1556	1685
Max. liquid pressure (MPa)	264	261
Max. combustion pressure (MPa)	205	203
Time until projectile exit (ms)	10.0	10.2
Computer time (s)	32.8	2.4

In conclusion, the Gough code may be inaccurate if the mechanisms controlling the time step occur in the lumped parameter regions. The new code, however, will overpredict the muzzle velocity for a long gun tube. It may also be inaccurate for a high performance (large muzzle velocity) gun, if the projectile moves fast enough that the pressure increase in the throat of the gun tube cannot catch the projectile. The new code runs about 10 to 15 times faster than the Gough code. Several other comparisons have been made, and the above conclusions still hold.

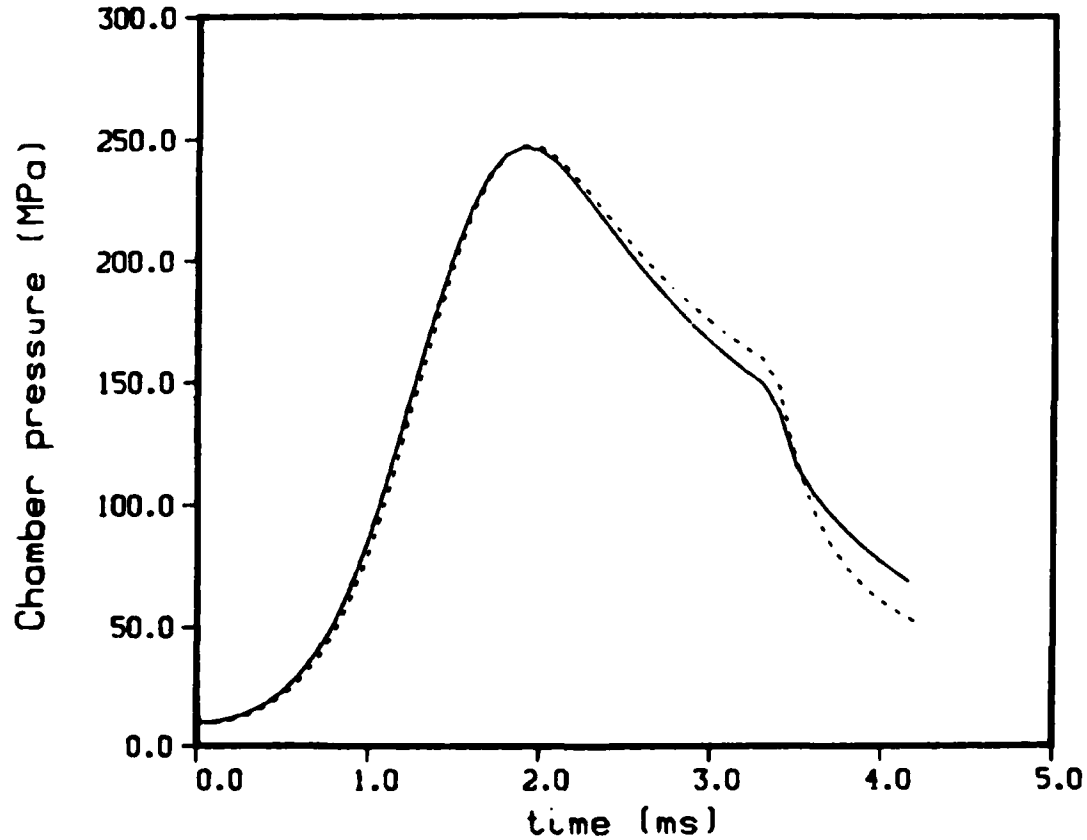

Figure 3. Baseline 25 mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Chamber Pressures. Coffee model (line); Gough model (dot)

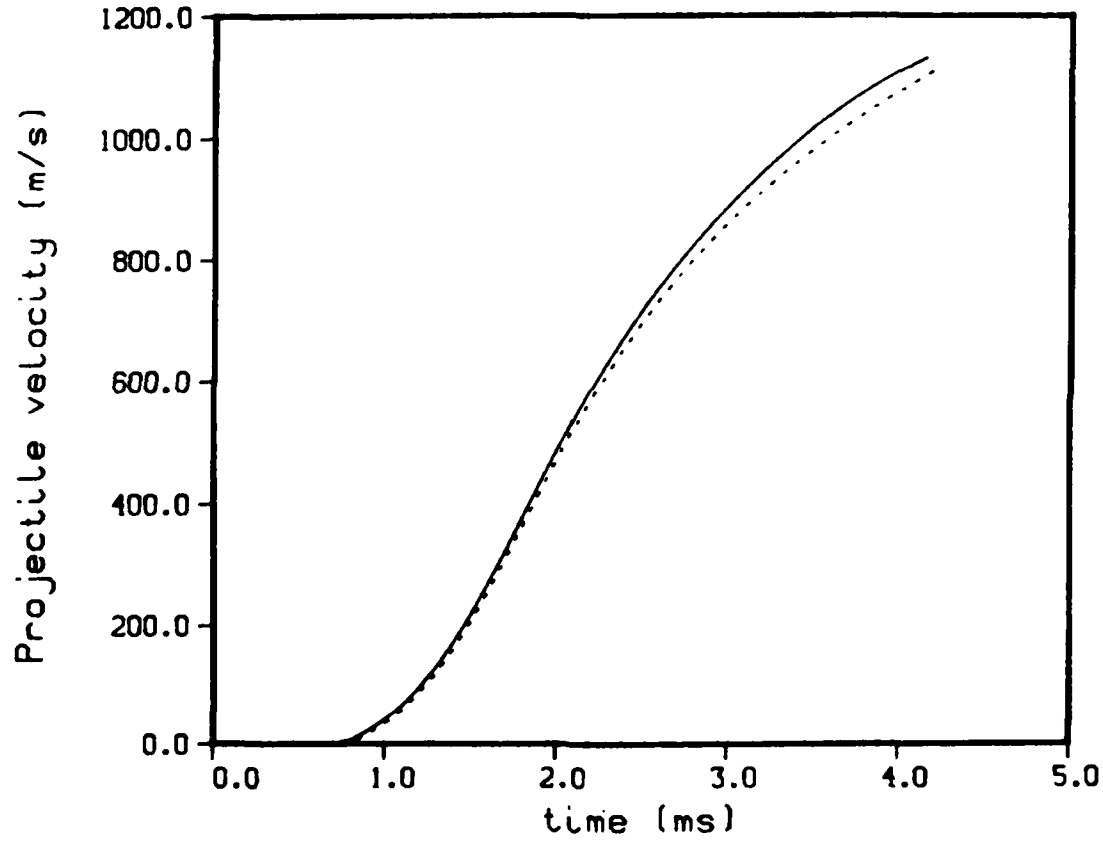

Figure 4. Baseline 25 mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Projectile Velocities. Coffee model (line); Gough model (dot)

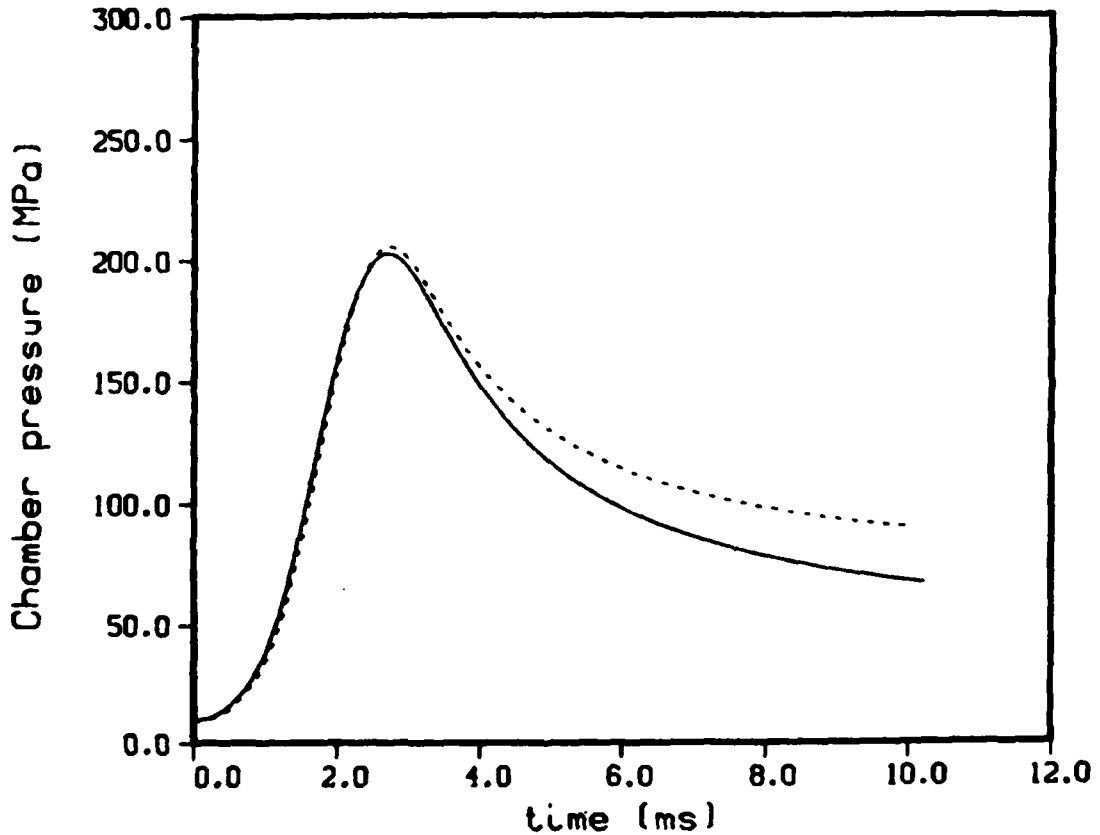

Figure 5. Long 25 mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Chamber Pressures. Coffee model (line); Gough model (dot)

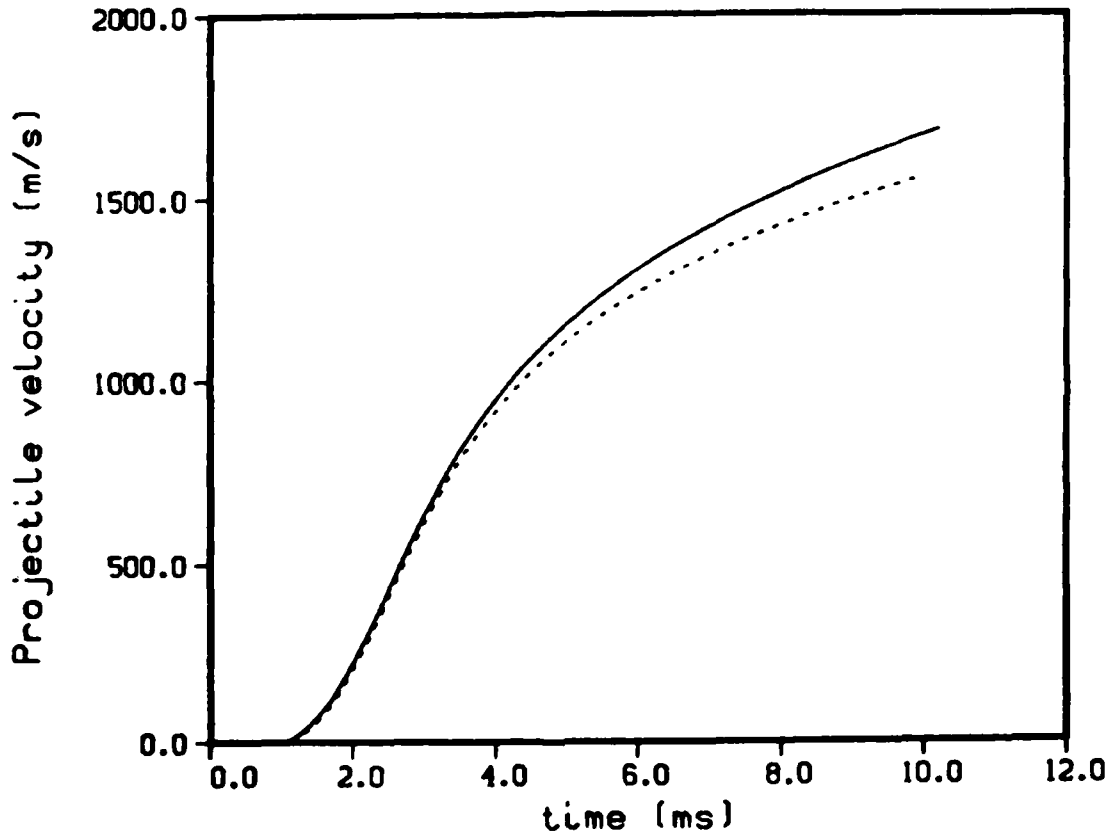

Figure 6. Long 25 mm Gun - Instantaneous Combustion - Initial Chamber Pressure = 10 MPa. - Projectile Velocities. Coffee model (line); Gough model (dot)

REFERENCES

1. W.F. Morrison, J.D. Knapton, and G. Klingenberg, "Liquid Propellants for Gun Applications," Proceedings of the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April 1983.

2. P.G. Baer, "Practical Interior Ballistic Analysis of Guns," Progress in Astronautics and Aeronautics, Interior Ballistics of Guns, (H. Krier and M. Summerfield, ed.), Vol. 66, 1979.

3. G. Pagan and D.C.A. Izod, "Regenerative Liquid Propellant Gun Modelling," Proceedings of the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April 1983.

4. P.G. Cushman, "Regenerative Liquid Propellant Gun Simulation User's Manual," GE Report 84-POD-004, December 1983.

5. P.S. Gough, "A Model of the Interior Ballistics of Hybrid Liquid-Propellant Guns," Final Report, Contract DAAK11-82-C-0154, PGA-TR-83-4, September 1983.

6. W.F. Morrison, M.J. Bulman, P.G.Baer, and C.F. Banz, "The Interior Ballistics of Regenerative Liquid Propellant Guns," 1984 JANNAF Propulsion Meeting, New Orleans, LA, CPIA Publication 390, February 1984.

7. P. G. Baer, C.F. Banz, I.W. May and W.F. Morrison, "A Propulsion System Comparison Study for the 120-MM Anti-Armor Cannon," 1984 JANNAF Propulsion Meeting, New Orleans, LA, CPIA Publication 390, February 1984.

8. J. Corner, Theory of the Interior Ballistics of Guns, Wiley, New York, 1950.

9. W.F. Hughes and J.A. Brighton, Fluid Dynamics, Schaum's Outline Series, McGraw-Hill, New York, 1967.

10. W. Kaufman, Fluid Mechanics, McGraw-Hill, New York, 1963.

11. R.W. Fox and A.T. McDonald, Introduction to Fluid Mechanics, 2nd ed., John Wiley and Sons, NY, 1978.

12. W. Band, Introduction to Mathematical Physics, D. Van Nostrand Company, Princeton, NJ, 1960.

13. G.N. Lewis and M. Randall, Thermodynamics, McGraw-Hill, New York, 1961.

14. R.B. Bird, W.E. Stewart, and R.B. Bird, Transport Phenomena, John Wiley and Sons, New York, 1960.

15. W.F. McBratney, "Windowed Chamber Investigation of the Burning Rate of Liquid Monopropellants for Guns," ARBRL-MR-03018, April 1980.

16. W.F. McBratney, "Burning Rate Data, LPG 1845," ARBRL-MR- 03128, August 1981.

17. F.A. Williams, "Progress in Spray-Combustion Analysis," in The Eighth International Combustion Symposium, Williams and Wilkins Co., Baltimore, MD, 1962.

18. A.C. Hindmarsh and G.D. Byrne, "EPISODE: An Effective Package for the Integration of Systems of Ordinary Differential Equations," UCID-30112-Rev.1, Lawrence Livermore Laboratory, 1977.

19. T.P. Coffee, "A Computer Code for the Solution of the Equations Governing a Laminar, Premixed, One-Dimensional Flame," ARBRL-MR-03165, April 1982.

20. P.J. Roache, Computational Fluid Dynamics, Hermosa Publishers, 1972.

APPENDIX A

A listing of the computer code follows. Routines RLGD to PEDERV have been written for the regenerative liquid gun application. Routines DRIVE to SING comprise the code EDISODE.

```
      PROGRAM RLGD(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     + TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,TAPE14,
     + TAPE15,TAPE16,TAPE17,TAPE2)
C  THIS PROGRAM IS AN IMPLEMENTATION OF EPISODE FOR
C  INTERIOR BALLISTICS PROBLEMS.
C  REGENERATIVE LIQUID PROPELLANT GUN.
C  FEBRUARY 13, 1985.
      DIMENSION Y(50),YDOT(50)
      DIMENSION II(72)
      COMMON/EPCOM9/HUSED,NOUSED,NSTEP,NFE,NJE,NMI
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET
      COMMON/TAB2/RK1,RK2,RHII,VII,VPER
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4
      COMMON/TAB7/PJWT,TRAVEL
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,FKG4,ESUM,EINT
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HLI,HL3,HL4,H3,H4
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)
      COMMON/TAB12/NVENT,NVENT,AVMIN,AVMAX,AK,ACS
      COMMON/TAB13/PI
      COMMON/TAB14/DDR,ADR,BDR
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4
      COMMON/TAB18/DINC,PODR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)
      COMMON/TAB19/SIGMA,U13,WE
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)
      DIMENSION TPRIM(8)
      DATA PI/3.14159265 4/
C  ************************
C  THIS ROUTINE CONTROLS THE PROGRAM.
C  ALL INPUT DATA IS READ IN HERE (TAPE 5).
C  INFORMATION IS PRINTED ON THE OUTPUT FILES (TAPE6-TAPE12).
C  GRAPHICS INFORMATION IS PRINTED ON TAPE17.
C  ************************
C  ************************
      FT=SECOND(CP)
C  READ IN THE INPUT.
      READ(5,2)TITLE
    2 FORMAT(8A10)
      WRITE(6,3)TITLE
    3 FORMAT(1H1,1X,8A10)
      LBSTAR=1H*
    5 READ(5,8)(II(I),I=1,72)
    8 FORMAT(72A1)
      IF(II(1).EQ.LBSTAR)GO TO 9
      WRITE(6,28)II
   28 FORMAT(2X,72A1)
      GO TO 5
    9 CONTINUE
      READ(5,10)OFFSET,TRAVEL
      WRITE(6,100)OFFSET,TRAVEL
      READ(5,10)D4,DC34
      A4=3.14159265*D4*D4/4.0
```

```
      WRITE(6,115)D4,A4                                                 000670
      WRITE(6,120)DC34                                                  000680
      READ(5,10)PJWT                                                    000690
      READ(5,10)PSWT                                                    000700
      WRITE(6,140)PJWT,PSWT                                             000710
      READ(5,15)MBACK                                                   000720
      IF(MBACK.EQ.1)WRITE(6,142)MBACK                                   000730
      IF(MBACK.EQ.0)WRITE(6,143)MBACK                                   000740
      READ(5,10)V1I,V3I                                                 000750
      VOL=V1I+V3I                                                       000760
      WRITE(6,150)V1I,V3I,VOL                                           000770
      READ(5,10)VPER                                                    000780
      WRITE(6,152)VPER                                                  000790
      READ(5,10)A1,A3                                                   000800
      WRITE(6,155)A1,A3                                                 000810
      BORE=SQRT(4.0*A3/PI)                                              000820
      READ(5,2)TVENT                                                    000830
      WRITE(6,158)TVENT                                                 000840
C  VENT AREA TABLE                                                      000850
      IF(TVENT(1).EQ.10HVENT1      )CALL VENT1(1)                       000860
C  CONSTANT ACCELERATION OPTION - CHANGE VENT AREA BY 'MAGIC'           000870
      IF(TVENT(1).EQ.10HVENT2      )CALL VENT2(1)                       000880
C  CENTRAL ROD OR BOLT - INFINITELY THIN PISTON                         000890
      IF(TVENT(1).EQ.10HVENT3      )CALL VENT3(1)                       000900
      BSS=BORE/PSTRAV                                                   000910
      WRITE(6,159)BORE,BSS                                              000920
C  ESTIMATE LENGTH OF COMBUSTOR BY PASKO FORMULA.                       000930
C  2.25*LENGTH LIQUID CHAMBER + LENGTH COMBUSTION CHAMBER               000940
C  + PISTON THICKNESS (5.4 CM) + END PLATE (10.2 CM).                   000950
      CLEN=2.25*PSTRAV+V3I/A3+5.4+10.2                                  000960
      WRITE(6,161)CLEN                                                  000970
C  PISTON RESISTANCE TABLE.                                             000980
      CALL  PISRES(1,0.0,PSRS)                                          000990
C  DISCHARGE COEFFICIENT TABLE                                          001000
      READ(5,2)TDIS                                                     001010
      WRITE(6,158)TDIS                                                  001020
C  DISCHARGE COEFFICIENT AS A FUNCTION OF PISTON TRAVEL.                001030
      IF(TDIS(1).EQ.10HDIS1        )CALL DIS1(1)                        001040
C  DISCHARGE COEFFICIENT AS A FUNCTION OF LIQUID PRESSURE.              001050
      IF(TDIS(1).EQ.10HDIS2        )CALL DIS2(1)                        001060
C  PROJECTILE RESISTANCE TABLE.                                         001070
      CALL  PROJRES(1,0.0,PJRS)                                         001080
      READ(5,10)RH1I,RK1,RK2                                            001090
      WRITE(6,165)RH1I,RK1,RK2                                          001100
      READ(5,10)ENER,GAM                                                001110
      WRITE(6,200)ENER,GAM                                              001120
      READ(5,10)SIGMA                                                   001130
      WRITE(6,202)SIGMA                                                 001140
      READ(5,10)WG,COV                                                  001150
      WRITE(6,205)WG,COV                                                001160
      READ(5,10)PLI,PGI                                                 001170
      WRITE(6,170)PLI,PGI                                               001180
C  UNIVERSAL GAS CONSTANT.                                              001190
      RU=8.318                                                          001200
C  SPECIFIC GAS CONSTANT.                                               001210
      RS=RU/WG                                                          001220
C  SPECIFIC HEAT OF THE GAS (CONSTANT VOLUME).                          001230
```

```
      CV=RS/(GAM-1.0)                                                   001240
C  SPECIFIC HEAT OF THE GAS (CONSTANT PRESSURE).                        001250
      CP=GAM*CV                                                         001260
      WRITE(6,172)RS                                                    001270
      WRITE(6,173)CV,CP                                                 001280
      READ(5,2)TDROP                                                    001290
      WRITE(6,158)TDROP                                                 001300
C  DROPLETS - CONSTANT SIZE                                             001310
      IF(TDROP(1).EQ.10HDROP2     )CALL DROP2                           001320
C  DROPLETS - SHRINK WHILE BURNING                                      001330
      IF(TDROP(1).EQ.10HDROP3     )CALL DROP3                           001340
C  PRIMER OPTION.                                                       001350
C  IF TPRIM = PRIM1, THEN PRIMER INSTANTANEOUSLY GOES TO GAS.           001360
C  IF TPRIM = PRIM2, THEN PRIMER STARTS AS FLUID.                       001370
C  THE COMBUSTION CHAMBER IS AT ATMOSPHERIC PRESSURE.                   001380
C  MUST HAVE A DROPLET OPTION.                                          001390
C  IF SHRINKING DROPLET OPTION, NPRIM IS THE CLASS INTO                 001400
C  WHICH THE LIQUID IS PUT.                                             001410
      READ(5,2)TPRIM                                                    001420
      WRITE(6,158)TPRIM                                                 001430
      READ(5,15)NPRIM                                                   001440
      WRITE(6,156)NPRIM                                                 001450
C  INTEGRATION PARAMETERS.                                              001460
      READ(5,10)TINC                                                    001470
      WRITE(6,175)TINC                                                  001480
      READ(5,10)EPS,SREC                                                001490
      READ(5,15)MF,KWRITE                                               001500
      WRITE(6,185)MF,EPS,SREC                                           001510
      IF(KWRITE.EQ.1)WRITE(6,186)KWRITE                                 001520
10    FORMAT(1P5E12.4)                                                  001530
12    FORMAT(I4,1P5E12.4)                                               001540
15    FORMAT(10I4)                                                      001550
100   FORMAT(/5X,'OFFSET =',1PE12.4,5X,'TRAVEL =',1PE12.4/)             001560
115   FORMAT(/5X,'TUBE DIAM =',1PE12.4,5X,'TUBE AREA =',1PE12.4)        001570
120   FORMAT(/5X,'TUBE ENTRANCE COEFF =',1PE12.4)                       001580
140   FORMAT(/5X,'PJWT = ',1PE12.4,5X,'PSWT =',1PE12.4)                 001590
142   FORMAT(/5X,'MBACK =',I4,5X,'BACK FLOW')                           001600
143   FORMAT(/5X,'MBACK =',I4,5X,'NO BACK FLOW')                        001610
150   FORMAT(/5X,'V1 IN =',1PE12.4,5X,'V3 IN =',1PE12.4,5X,             001620
     * 'VOL =',1PE12.4)                                                 001630
152   FORMAT(/5X,'VPER =',1PE12.4)                                      001640
155   FORMAT(/5X,'AREA FUEL =',1PE12.4,5X,'AREA  COMB =',1PE12.4)       001650
  156 FORMAT(/5X,'NPRIM =',I4)                                          001660
      G0=1.0E7                                                          001670
  158 FORMAT(//5X,8A10)                                                 001680
  159 FORMAT(/5X,'BORE =',1PE12.4,5X,'BORE/STROKE =',1PE12.4)           001690
  161 FORMAT(/5X,'COMBUSTOR LENGTH =',1PE12.4)                          001700
  162 FORMAT(/5X,'DISCHARGE COEFF =',1PE12.4)                           001710
165   FORMAT(/5X,'DENS LIQUID =',1PE12.4,5X,'K1 =',1PE12.4,5X,          001720
     * 'K2 =',1PE12.4)                                                  001730
170   FORMAT(/5X,'PRES LIQUID =',1PE12.4,5X,'PRES GAS =',1PE12.4)       001740
172   FORMAT(/5X,'SPECIFIC GAS CONSTANT =',1PE12.4)                     001750
173   FORMAT(/5X,'CV =',1PE12.4,5X,'CP =',1PE12.4)                      001760
  175 FORMAT(//5X,'TINC =',1PE12.4)                                     001770
185   FORMAT(/5X,'MF =',I4,5X,'EPS =',1PE12.4,5X,'SREC =',1PE12.4)      001780
  186 FORMAT(/5X,'KWRITE =',I4,5X,'PRINT EACH TIME STEP')               001790
200   FORMAT(/5X,'CHEM ENERGY =',1PE12.4,5X,'GAM =',1PE12.4)            001800
```

```
  202 FORMAT(/5X,'SURFACE TENSION =',1PE12.4)
  205   FORMAT(/5X,'MOL WT GAS =',1PE12.4,5X,'COVOLUME =',1PE12.4)
C  ************************
C  INITIAL CONDITIONS.
C  Y(1) IS THE LIQUID VOLUME.
      Y(1)=V1I
C  Y(2) IS THE PISTON VELOCITY.
      Y(2)=0.0
C  Y(3) IS THE PISTON TRAVEL.
      Y(3)=0.0
C  Y(4) IS THE LIQUID DENSITY AT THE GIVEN PRESSURE.
      IF(RK2.EQ.0.0)Y(4)=RH1I*EXP(PLI/RK1)
      IF(RK2.NE.0.0)Y(4)=RH1I*((RK2*PLI/RK1+1.0)**(1.0/RK2))
C  Y(5) IS THE LIQUID PRESSURE.
      Y(5)=PLI
C  Y(6) IS THE COMBUSTION CHAMBER  VOLUME.
      Y(6)=V3I
C  Y(7) IS THE COMBUSTION CHAMBER DENSITY.
      RH3=PGI/(ENER*(GAM-1.0)+COV*PGI)
      Y(7)=RH3
C  Y(8) IS THE COMBUSTION CHAMBER PRESSURE.
      Y(8)=PGI
      MOD1=1
      MOD4=1
      IF(OFFSET.LT.0.01)MOD4=0
C  Y(9) IS THE TUBE VOLUME.
      Y(9)=OFFSET*A4
C  Y(10) IS THE PROJECTILE VELOCITY
      Y(10)=0.0
C  Y(11) IS THE PROJECTILE TRAVEL
      Y(11)=0.0
C  Y(12) IS THE TUBE DENSITY
      Y(12)=RH3
C  Y(13) IS THE TUBE PRESSURE
      Y(13)=PGI
C  THIS SECTION ONLY FOR DROPLET MODEL.
C  Y(14) IS THE LIQUID DENSITY IN REGION 3.
      Y(14)=Y(4)
C  Y(15) IS THE LIQUID MASS IN REGION 3.
      Y(15)=0.0
C  Y(16) IS THE LIQUID DENSITY IN REGION 4.
      Y(16)=Y(4)
C  Y(17) IS THE LIQUID MASS IN REGION 4.
      Y(17)=0.0
C  THIS SECTION ONLY FOR SHRINKING DROPLET MODEL.
C  Y(18) TO Y(27) ARE THE PARTITIONED LIQUID MASS IN REGION 3.
C  Y(28) TO Y(37) ARE THE PARTITIONED LIQUID MASS IN REGION 4.
C  THESE ARE THE MASSES OF THE DROPLETS WITH DIAMETER PODR(I).
      DO 295 I=18,37
  295 Y(I)=0.0
C  ********************
C  IF PRIMER OPTION 2, CHANGE THE INITIAL CONDITIONS.
      PRIMER=V3I*RH3+Y(9)*RH3
      IF(TPRIM(1).EQ.10HPRIM2      )CALL PRIM2(NPRIM,PRIMER,PGI,Y)
C  ************************
C  SET THE INTEGRATOR PARAMETERS.
      IF(TDROP(1).EQ.10HDROP1      )N=13
```

```
      IF(TDROP(1).EQ.10HDROP2      )N=17
      IF(TDROP(1).EQ.10HDROP3      )N=37
      T0=0.0
      TOUT=TINC
      TMS=0.0
      H0=1.0E-10
      IERROR=2
      NSTEP=0
      MFE=0
      NJE=0
      NMI=0
      INDEX=1
      LSKIP=0
      LPAGE=0
      DO 300 K=1,10
300   RMAX(K)=0.0
      CALL DIFFUN(N,0.0,Y,YDOT)
      PRIMER=RM3+RM4
      CHMASS=(RM1+RM3+RM4)/PJWT
      WRITE(6,225)RM1,PRIMER,CHMASS
225   FORMAT(/5X,'CHARGE =',1PE12.4,5X,'PRIMER =',1PE12.4,5X,
     * 'C/M =',1PE12.4)
      DENSL=(RM1+RM3+RM4)/VOL
      WRITE(6,240)DENSL
240   FORMAT(/5X,'LOADING DENSITY =',1PE12.4)
      RMINT=RMSUM
      EINT=ESUM
C  ************************
C  CALL THE INTEGRATOR.
      CALL CAPTION
      CALL OUT
390   CALL DRIVE(N,T0,H0,Y,TOUT,EPS,IERROR,MF,INDEX)
      IF(INDEX.NE.0)STOP
      CALL DIFFUN(N,TOUT,Y,YDOT)
      TMS=TOUT*1000.
      IF(SPJ.GE.TRAVEL)GO TO 500
      CALL OUT
      IF(MOD1.EQ.0.AND.VPJ.EQ.0.0)GO TO 550
      TOUT=TOUT+TINC
      GO TO 390
500   CONTINUE
C  FIND THE TIME CORRESPONDING TO THE PROJECTILE LEAVING THE MUZZLE.
      TINC=TINC/5.0
505   TOUT=TOUT-TINC
      CALL DRIVE(N,T0,H0,Y,TOUT,EPS,IERROR,MF,INDEX)
      CALL DIFFUN(N,TOUT,Y,YDOT)
      IF(SPJ.GT.TRAVEL)GO TO 505
      TINC=TINC/5.0
510   TOUT=TOUT+TINC
      CALL DRIVE(N,T0,H0,Y,TOUT,EPS,IERROR,MF,INDEX)
      CALL DIFFUN(N,TOUT,Y,YDOT)
      IF(SPJ.LT.TRAVEL)GO TO 510
      TMS=TOUT*1000.
      CALL OUT
550   CONTINUE
C  WRITE THE MAX VALUES.
C CHANGE MUZZLE VELOCITY TO M/SEC.
```

```
      VM=VPJ/100.
C  CHANGE MASS AND ENERGY ERRORS TO PERCENTAGES.
      RMAX(6)=RMAX(6)*100.
      RMAX(7)=RMAX(7)*100.
      WRITE(6,590)VM
      WRITE(6,600)RMAX(1)
      WRITE(6,610)RMAX(2)
      WRITE(6,620)RMAX(3)
      WRITE(6,630)RMAX(4)
      WRITE(6,640)RMAX(5)
      WRITE(6,650)RMAX(6)
      WRITE(6,660)RMAX(7)
  590 FORMAT(1H1,4X,'MUZZLE VEL (M/SEC)  ',F10.1)
  600 FORMAT(//5X,  'MAX P1 (MPA)        ',F10.1)
  610 FORMAT(/5X,   'MAX P3 (MPA)        ',F10.1)
  620 FORMAT(/5X,   'MAX PL (MPA)        ',F10.1)
  630 FORMAT(/5X,   'MAX PR (MPA)        ',F10.1)
  640 FORMAT(//5X,  'MAX ACC (K-G)       ',F10.1)
  650 FORMAT(//5X,  'MAX MASS ERROR      ',F10.2)
  660 FORMAT(/5X,   'MAX ENERGY ERROR    ',F10.2)
      GT=SECOND(CP)
      HT=GT-FT
      WRITE(6,575)HT,NSTEP
  575 FORMAT(//5X,'RUN TIME =',F12.1,5X,'NSTEP =',I8)
      STOP
      END
```

```
      SUBROUTINE VENT1(ISW)                                             003210
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                                003220
      COMMON/TAB2/RK1,RK2,RHII,VII,VPER                                 003230
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                          003240
      COMMON/TAB4/PSWT,PSTPAV,MOD1,MHACK,MOD4                           003250
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,PM1,RM3,RM4,RMSUM,PMINT   003260
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     003270
      COMMON/TAB7/PJWT,TRAVEL                                           003280
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                  003290
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,FSUM,EINT                  003300
      COMMON/TAB10/RH,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         003310
      COMMON/TAB11/LSKIP,LPAGE,TMS,PMAX(10)                             003320
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                       003330
      COMMON/TAB13/PI                                                   003340
      COMMON/TAB14/DDR,ADR,BDR                                          003350
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,PMG3,PML3,PMG4,PML4              003360
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4                     003370
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            003380
      COMMON/TAB18/DINC,PODR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    003390
      COMMON/TAB19/SIGMA,U13,WE                                         003400
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)003410
      DIMENSION SVT(30),AVT(30)                                         003420
C  READ IN A TABLE OF VENT OPENINGS.                                    003430
C  LINEAR INTERPOLATION TO GET VENT OPENING                             003440
C  FOR ANY DESIRED PISTON TRAVEL.                                       003450
C  VENT AREA CHANGES BY 'MAGIC' IN PISTON FACE.                         003460
C  DOES NOT AFFECT THE LIQUID RESERVOIR VOLUME.                         003470
      IF(ISW.EQ.2)GO TO 100                                             003480
C  READ IN THE REQUIRED PARAMETERS.                                     003490
      READ(5,15)NVENT                                                   003500
   15 FORMAT(10I4)                                                      003510
      WRITE(6,20)NVENT                                                  003520
   20 FORMAT(/5X,'NVENT =',I4/)                                         003530
      DO 30 I=1,NVENT                                                   003540
      READ(5,25)SVT(I),AVT(I)                                           003550
   25 FORMAT(1P5E12.4)                                                  003560
      HYD=(A3-AVT(I))/(A1-AVT(I))                                       003570
      WRITE(6,28)SVT(I),AVT(I),HYD                                      003580
   28 FORMAT(5X,'PIS TRAV =',1PE12.4,5X,'VENT AREA =',1PE12.4,5X,       003590
     * 'HYDROLIC DIFF =',1PE12.4)                                       003600
   30 CONTINUE                                                          003610
C  FIND THE MAX PISTON TRAVEL.                                          003620
      PSTPAV=V1I/A1                                                     003630
      WRITE(6,40)PSTRAV                                                 003640
   40 FORMAT(/5X,'MAX PISTON TRAVEL =',1PE12.4/)                        003650
      SPS=0.0                                                           003660
  100 CONTINUE                                                          003670
C  FIND THE VENT AREA FROM THE TABLE.                                   003680
      DO 120 I=2,NVENT                                                  003690
      IM=I-1                                                            003700
      IF(SPS.LT.SVT(IM).OR.SPS.GE.SVT(I))GO TO 115                      003710
      PP=(SPS-SVT(IM))/(SVT(I)-SVT(IM))                                 003720
      AVENT=AVT(IM)+PP*(AVT(I)-AVT(IM))                                 003730
  115 CONTINUE                                                          003740
  120 CONTINUE                                                          003750
      IF(SPS.GT.SVT(NVENT))AVENT=AVT(NVENT)                             003760
      RETURN                                                            003770
```

```
      END                                                       003790
```

```
      SUBROUTINE VENT2(ISW)                                             003790
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                                003800
      COMMON/TAB2/RK1,RK2,RHII,VII,VPER                                 003810
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                          003820
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                           003830
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT   003840
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     003850
      COMMON/TAB7/PJWT,TRAVEL                                           003860
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                  003870
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT                  003880
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         003890
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)                             003900
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                       003910
      COMMON/TAB13/PI                                                   003920
      COMMON/TAB14/DDR,ADR,BDR                                          003930
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,PMG3,PML3,PMG4,PML4              003940
      COMMON/TAB16/PHG3,RML3,RHG4,PHL4,EL3,EL4,EKL4                     003950
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            003960
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    003970
      COMMON/TAB19/SIGMA,U13,WE                                         003980
      COMMON/TABINT/KWRITE,KOUTD,SPEC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)003990
C  CONSTANT ACCELERATION OPTION.                                        004000
C  CHOOSE THE VENT AREA TO KEEP THE BASE PRESSURE CONSTANT.             004010
C  ACTUAL CALCULATION IS DONE AT THE END OF EACH TIME STEP              004020
C  AT THE END OF TSTEP.                                                 004030
C  AVMIN = MIN VENT AREA                                                004040
C  AVMAX = MAX VENT AREA                                                004050
C  AK = CONSTANT CONTROLLING THE RATE OF CHANGE OF THE VENT AREA        004060
C  ACS = DESIRED CONSTANT ACCELERATION IN KILO-G                        004070
      IF(ISW.EQ.2)GO TO 100                                             004080
      READ(5,20)AVMIN,AVMAX                                             004090
      READ(5,20)AK,ACS                                                  004100
   20 FORMAT(1P5E12.4)                                                  004110
      HYD=(A3-AVMIN)/(A1-AVMIN)                                         004120
      WRITE(6,30)AVMIN,HYD                                              004130
   30 FORMAT(/5X,'AVMIN =',1PE12.4,5X,'HYDROLIC DIFF =',1PE12.4)        004140
      HYD=(A3-AVMAX)/(A1-AVMAX)                                         004150
      WRITE(6,32)AVMAX,HYD                                              004160
   32 FORMAT(/5X,'AVMAX =',1PE12.4,5X,'HYDROLIC DIFF =',1PE12.4)        004170
      WRITE(6,40)AK,ACS                                                 004180
   40 FORMAT(/5X,'AK =',1PE12.4,5X,'ACS =',1PE12.4/)                    004190
  100 CONTINUE                                                          004200
      RETURN                                                            004210
      END                                                               004220
```

```
      SUBROUTINE VENT3(ISW)                                             004230
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                                004240
      COMMON/TAB2/RK1,RK2,RH1I,V1I,VPER                                 004250
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                          004260
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                           004270
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT   004280
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     004290
      COMMON/TAB7/PJWT,TRAVEL                                           004300
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                  004310
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT                  004320
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         004330
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)                             004340
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                       004350
      COMMON/TAB13/PI                                                   004360
      COMMON/TAB14/DDR,ADR,BDR                                          004370
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4              004380
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4                     004390
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            004400
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    004410
      COMMON/TAB19/SIGMA,U13,WE                                         004420
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)004430
      DIMENSION SVT(30),AVT(30),AROD(30),RROD(30)                       004440
C  CENTRAL ROD OR BOLT OPTION.                                          004450
C  ASSUME INFINITELY THIN PISTON.                                       004460
C  A1 IS THE AREA OF THE FUEL SIDE (INCLUDING BOLT HOLE).               004470
C  A3 IS THE AREA OF THE COMBUSTION SIDE (INCLUDING BOLT HOLE).         004480
C  READ IN THE AREA OF THE BOLT HOLE                                    004490
C  AND THE TABLE OF VENT AREAS.                                         004500
      IF(ISW.EQ.2)GO TO 100                                             004510
C  READ IN THE REQUIRED PARAMETERS.                                     004520
      READ(5,15)NVENT                                                   004530
   15 FORMAT(10I4)                                                      004540
      WRITE(6,20)NVENT                                                  004550
   20 FORMAT(/5X,'NVENT =',I4/)                                         004560
      DO 30 I=1,NVENT                                                   004570
      READ(5,25)SVT(I),AVT(I)                                           004580
   25 FORMAT(1P5E12.4)                                                  004590
      WRITE(6,28)SVT(I),AVT(I)                                          004600
   28 FORMAT(5X,'PISTON TRAV =',1PE12.4,5X,'VENT AREA =',1PE12.4)       004610
   30 CONTINUE                                                          004620
      READ(5,25)AHOLE                                                   004630
      WRITE(6,35)AHOLE                                                  004640
   35 FORMAT(/5X,'AREA OF CENTER HOLE =',1PE12.4)                       004650
      HYD=(A3-AHOLE)/(A1-AHOLE)                                         004660
      WRITE(6,38)HYD                                                    004670
   38 FORMAT(/5X,'HYDROLIC DIFF =',1PE12.4/)                            004680
C  COMPUTE THE AREAS OF THE ROD AND THE CORRESPONDING RADII.            004690
      DO 50 I=1,NVENT                                                   004700
      AROD(I)=AHOLE-AVT(I)                                              004710
   50 RROD(I)=SQRT(AROD(I)/PI)                                          004720
C  COMPUTE THE VOLUME OF THE LIQUID RESERVOIR.                          004730
      VL=0.0                                                            004740
      R1=SQRT(A1/PI)                                                    004750
      DO 60 I=2,NVENT                                                   004760
      IM=I-1                                                            004770
   60 VL=VL+PI*(SVT(I)-SVT(IM))*(R1*R1-(RROD(I)*RROD(I)+                004780
     * RROD(I)*RROD(IM)+RROD(IM)*RROD(IM))/3.0)                         004790
```

```
C  SCALE THE DISTANCES SO THAT THE LIQUID VOLUME MATCHES
C  THE INPUT LIQUID VOLUME (V1).
      SC=V1I/VL
      DO 62 I=1,NVENT
   62 SVT(I)=SVT(I)*SC
      DO 66 I=1,NVENT
      RVENT=SQRT((AVT(I)+AROD(I))/3.141592654)
      GAP=RVENT-RROD(I)
   66 WRITE(6,70)SVT(I),AVT(I),AROD(I),RROD(I),GAP
   70 FORMAT(5X,'PIS TRAV =',1PE12.4,5X,'AVT =',1PE12.4,5X,
     + 'AROD =',1PE12.4,5X,'RROD =',1PE12.4,5X,'GAP =',1PE12.4)
      PSTRAV=SVT(NVENT)
      WRITE(6,75)PSTRAV
   75 FORMAT(/5X,'MAX PISTON TRAVEL =',1PE12.4)
      SPS=0.0
  100 CONTINUE
C  FIND THE VENT AREA.
      DO 120 I=2,NVENT
      IM=I-1
      IF(SPS.LT.SVT(IM).OR.SPS.GE.SVT(I))GO TO 115
      PP=(SPS-SVT(IM))/(SVT(I)-SVT(IM))
C  INTERPOLATE THE ROD RADIUS.
      RR=RROD(IM)+PP*(RROD(I)-RROD(IM))
      RIM=RROD(IM)
      AR=PI*RR*RR
      AVENT=AHOLE-AR
C  FIND THE CORRECTION FOR THE RATE OF CHANGE OF V1 AND V3
C  DUE TO THE CENTER ROD.
C  DONE BY REPRESENTING VOLUME AS FUEL CHAMBER VOLUME MINUS
C  ROD VOLUME AND COMPUTING THE RATE OF CHANGE OF THE ROD VOLUME.
      RRDT=VPS*(RROD(I)-RROD(IM))/(SVT(I)-SVT(IM))
      VCOR=(PI/3.0)*(VPS*(RR*RR+RR*RIM+RIM*RIM)+
     + (SPS-SVT(IM))*RRDT*(RIM+2.0*RR))
  115 CONTINUE
  120 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DROP2                                                  005170
      COMMON/TABL4/DDR,ADR,BDR                                          005180
C  DROPLET - CONSTANT SIZE.                                             005190
C  READ IN THE INPUT PARAMETERS.                                        005200
      READ(5,10)DDR,ADR,BDR                                             005210
   10 FORMAT(1P5E12.4)                                                  005220
      WRITE(6,20)DDR,ADR,BDR                                            005230
   20 FORMAT(/5X,'DDR =',1PE12.4,5X,'ADR =',1PE12.4,5X,                 005240
     * 'BDR =',1PE12.4)                                                 005250
      RETURN                                                            005260
      END                                                               005270
```

```
      SUBROUTINE DROP3
      COMMON/TAB14/DDR,ADR,BDR
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)
      COMMON/TAB19/SIGMA,U13,WE
C  DROPLETS - SHRINK WHILE BURNING.
C  READ IN THE INPUT PARAMETERS.
      READ(5,10)DDR,ADR,BDR
   10 FORMAT(1P5F12.4)
      WRITE(6,20)DDR,ADR,BDR
   20 FORMAT(/5X,'DDR =',1PE12.4,5X,'ADR =',1PE12.4,5X,
     + 'BDR =',1PE12.4)
      I=1
      PDDR(1)=DDR*0.95
      DINC=DDR/10.0
      WRITE(6,50)I,PDDR(I)
      DO 30 I=2,10
      IM=I-1
      PDDR(I)=PDDR(IM)-DINC
      WRITE(6,60)I,PDDR(I)
   30 CONTINUE
   50 FORMAT(/5X,'PDDR',I4,' =',1PE12.4)
   60 FORMAT(5X,'PDDR',I4,' =',1PE12.4)
      RETURN
      END
```

```
      SUBROUTINE PISRES(ISW,SPS,PSRS)
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MPACK,MOD4
      DIMENSION SFRAC(10),SPIS(10),PRS(10)
      IF(ISW.EQ.2)GO TO 100
C  READ IN THE PISTON RESISTANCE.
      READ(5,10)NPIS
   10 FORMAT(10I4)
      WRITE(6,12)NPIS
   12 FORMAT(/5X,'NPIS =',I4,5X/)
      DO 25 I=1,NPIS
      READ(5,20)SFRAC(I),PRS(I)
   20 FORMAT(1P10E12.4)
   25 CONTINUE
C  NORMALIZE TO MAX PISTON TRAVEL.
      DO 35 I=1,NPIS
      SPIS(I)=(SFRAC(I)/SFRAC(NPIS))*PSTRAV
      WRITE(6,30)SFRAC(I),SPIS(I),PRS(I)
   30 FORMAT(5X,'FRAC PIS TRAV =',1PE12.4,5X,'PIS TRAV =',1PE12.4,
     + 5X,'PIS RESISTANCE =',1PE12.4)
   35 CONTINUE
      RETURN
  100 CONTINUE
C  FIND THE PISTON RESISTANCE PRESSURE.
      DO 110 I=2,NPIS
      IM=I-1
      IF(SPS.LT.SPIS(IM).OR.SPS.GE.SPIS(I))GO TO 108
      PP=(SPS-SPIS(IM))/(SPIS(I)-SPIS(IM))
      PSRS=PRS(IM)+PP*(PRS(I)-PRS(IM))
  108 CONTINUE
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISI(ISW)
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOP
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MRACK,MOD4
      COMMON/TAB6/V1,VPS,SPS,PH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,PH4,P4
      DIMENSION SFRAC(30),SPIS(30),DCF(30)
C  DISCHARGE COEFFICIENT AS A FUNCTION OF PISTON TRAVEL.
      IF(ISW.EQ.2)GO TO 100
C  READ IN THE DISCHARGE COEFFICIENT TABLE.
      READ(5,10)NDC
   10 FORMAT(10I4)
      WRITE(6,12)NDC
   12 FORMAT(/5X,'NDC =',I4,5X/)
      DO 25 I=1,NDC
      READ(5,20)SFRAC(I),DCF(I)
   20 FORMAT(1P10E12.4)
   25 CONTINUE
C  NORMALIZE TO MAX PISTON TRAVEL.
      DO 35 I=1,NDC
      SPIS(I)=(SFRAC(I)/SFRAC(NDC))*PSTPAV
      WRITE(6,30)SFRAC(I),SPIS(I),DCF(I)
   30 FORMAT(5X,'FRAC PIS TRAV =',1PE12.4,5X,'PIS TRAV =',1PE12.4,
     + 5X,'DIS COEFF =',1PE12.4)
   35 CONTINUE
      RETURN
  100 CONTINUE
C  FIND THE DISCHARGE COEFFICIENT.
      DO 110 I=2,NDC
      IM=I-1
      IF(SPS.LT.SPIS(IM).OR.SPS.GE.SPIS(I))GO TO 108
      PP=(SPS-SPIS(IM))/(SPIS(I)-SPIS(IM))
      DC=DCF(IM) + PP*(DCF(I)-DCF(IM))
  108 CONTINUE
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DIS2(ISW)
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4
      DIMENSION PLIQ(30),DCF(30)
C  DISCHARGE COEFFICIENT AS A FUNCTION OF LIQUID PRESSURE.
      IF(ISW.EQ.2)GO TO 100
C  READ IN THE DISCHARGE COEFFICIENT TABLE.
      READ(5,10)NDC
   10 FORMAT(10I4)
      WRITE(6,12)NDC
   12 FORMAT(/5X,'NDC =',I4,5X/)
      DO 25 I=1,NDC
      READ(5,20)PLIQ(I),DCF(I)
   20 FORMAT(1P10E12.4)
      WRITE(6,30)PLIQ(I),DCF(I)
   30 FORMAT(5X,'LIQUID PRESSURE =',1PE12.4,5X,'DIS COEFF =',1PE12.4)
   25 CONTINUE
      PMAX=0.0
      RETURN
  100 CONTINUE
C  FIND THE DISCHARGE COEFFICIENT.
      DO 110 I=2,NDC
      IM=I-1
      IF(P1.LT.PLIQ(IM).OR.P1.GE.PLIQ(I))GO TO 108
      PP=(P1-PLIQ(IM))/(PLIQ(I)-PLIQ(IM))
      DC=DCF(IM) + PP*(DCF(I)-DCF(IM))
  108 CONTINUE
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PROJRES(ISW,SPJ,PJRS)                                  006500
      DIMENSION STR(10),PTR(10)                                         006510
      IF(ISW.EQ.2)GO TO 100                                             006520
C  FIND THE PROJECTILE RESISTANCE PRESSURE.                             006530
      READ(5,10)NPROJ                                                   006540
   10 FORMAT(10I4)                                                      006550
      WRITE(6,12)NPROJ                                                  006560
   12 FORMAT(/5X,'NPROJ =',I4/)                                         006570
      DO 25 I=1,NPROJ                                                   006580
      READ(5,20)STR(I),PTR(I)                                           006590
   20 FORMAT(1P10E12.4)                                                 006600
      WRITE(6,22)STR(I),PTR(I)                                          006610
   22 FORMAT(5X,'TRAVEL =',1PE12.4,5X,'RESISTIVE PRESS =',1PE12.4)      006620
   25 CONTINUE                                                          006630
      RETURN                                                            006640
  100 CONTINUE                                                          006650
C  FIND THE RESISTANCE PRESSURE.                                        006660
      DO 110 I=2,NPROJ                                                  006670
      IM=I-1                                                            006680
      IF(SPJ.LT.STR(I-1).OR.SPJ.GE.STR(I))GO TO 108                     006690
      PP=(SPJ-STR(IM))/(STR(I)-STR(IM))                                 006700
      PJRS=PTR(IM) + PP*(PTR(I)-PTR(IM))                                006710
  108 CONTINUE                                                          006720
  110 CONTINUE                                                          006730
      IF(SPJ.GT.STR(NPROJ))PJRS=0.0                                     006740
      RETURN                                                            006750
      END                                                               006760
```

```
      SUBROUTINE PRIM2(NPRIM,PRIMER,PGI,Y)                              006770
      DIMENSION Y(50)                                                   006780
C  PRIMER BEGINS AS LIQUID DROPLETS.                                    006790
C  SET CHAMBER PRESSURES BACK TO ATMOSPHERIC PRESSURE.                  006800
      Y(8)=.10135                                                       006810
      Y(13)=.10135                                                      006820
C  SPLIT THE PRIMER UP BETWEEN REGION 3 AND REGION 4.                   006830
      V3=Y(6)                                                           006840
      V4=Y(9)                                                           006850
      Y(15)=PRIMER*V3/(V3+V4)                                           006860
      Y(17)=PRIMER*V4/(V3+V4)                                           006870
C  SUBTRACT OFF SOME OF THE MASS FOR THE GAS COMPONENT.                 006880
      Y(15)=Y(15)*(1.0-.10135/PGI)                                      006890
      Y(17)=Y(17)*(1.0-.10135/PGI)                                      006900
C  IF SHRINKING DROPLET OPTION, SET LIQUID INTO DESIRED CLASS.          006910
      I=17+NPRIM                                                        006920
      Y(17+I)=Y(15)                                                     006930
      I=27+NPRIM                                                        006940
      Y(I)=Y(17)                                                        006950
      RETURN                                                            006960
      END                                                               006970
```

```
      SUBROUTINE CAPTION                                                006980
      WRITE(6,500)                                                      006990
      WRITE(7,501)                                                      007000
      WRITE(8,502)                                                      007010
      WRITE(9,503)                                                      007020
      WRITE(10,504)                                                     007030
      WRITE(11,505)                                                     007040
      WRITE(12,506)                                                     007050
      WRITE(13,507)                                                     007060
      WRITE(14,508)                                                     007070
      WRITE(15,509)                                                     007080
      WRITE(16,510)                                                     007090
      WRITE(17,511)                                                     007100
  500 FORMAT(1H1,4X,'T (MS)',5X,'P1',9X,'P3',9X,'PL',9X,'P4',          007110
     * 9X,'PR',9X,'S PS',7X,'V PS',7X,'S PJ',7X,'V PJ',7X,'ACC K-G'/)  007120
  501 FORMAT(1H1,4X,'T (MS)',5X,'RH1',8X,'RHL3',7X,'RHG3',7X,          007130
     * 'RH3',8X,'RHL4',7X,'RHG4',7X,'RH4'/)                            007140
  502 FORMAT(1H1,4X,'T (MS)',5X,'V1',9X,'VL3',8X,'VG3',8X,'V3',        007150
     * 9X,'VL4',8X,'VG4',8X,'V4',9X,'EPS3',7X,'EPS4',7X,'AVENT'/)      007160
  503 FORMAT(1H1,4X,'T (MS)',5X,'RM1',8X,'RML3',7X,'RMG3',7X,'RM3',    007170
     * 8X,'RML4',7X,'RMG4',7X,'RM4',9X,'RMSUM'/)                       007180
  504 FORMAT(1H1,4X,'T (MS)',5X,'RMD13',6X,'RMD3',7X,'RMD34',6X,       007190
     * 'RMD4',7X,'HL1',8X,'HL3',8X,'H3',9X,'HL4',8X,'H4'/)             007200
  505 FORMAT(1H1,4X,'T (MS)',5X,'CL1',8X,'CL3',8X,'CG3',8X,'C3',       007210
     * 9X,'CL4',8X,'CG4',8X,'C4',9X,'TEMP3',6X,'TEMP4'/)               007220
  506 FORMAT(1H1,4X,'T (MS)',5X,'EL1',8X,'EL3',8X,'EG3',8X,'EL4',      007230
     * 8X,'EG4',8X,'EKPS',7X,'EKPJ',7X,'EKL4',7X,'EKG4',7X,'ESUM'/)    007240
  507 FORMAT(1H1,4X,'T (MS)',5X,'PARTITIONED RML3'/)                   007250
  508 FORMAT(1H1,4X,'T (MS)',5X,'PARTITIONED RMD3'/)                   007260
  509 FORMAT(1H1,4X,'T (MS)',5X,'PARTITIONED RML4'/)                   007270
  510 FORMAT(1H1,4X,'T (MS)',5X,'PARTITIONED RMD4'/)                   007280
  511 FORMAT(1H1,4X,'T (MS)',5X,'U13',8X,'WE',9X,'DC'/)                007290
      RETURN                                                            007300
      END                                                               007310
```

```
      SUBROUTINE OUT                                                    007320
      COMMON/EPCOM9/MUSED,NOUSED,NSTEP,NFE,NJE,NMI                      007330
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                                007340
      COMMON/TAB2/PK1,PK2,RH1I,V1I,VPER                                 007350
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOP                          007360
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                           007370
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT   007380
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     007390
      COMMON/TAB7/PJWT,TRAVEL                                           007400
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                  007410
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT                  007420
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         007430
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)                             007440
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                       007450
      COMMON/TAB13/PI                                                   007460
      COMMON/TAB14/DDR,ADR,BDR                                          007470
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4              007480
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4                     007490
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            007500
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    007510
      COMMON/TAB19/SIGMA,U13,WE                                         007520
      COMMON/TABINT/KWRITE,KOUTD,SPEC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)007530
C  WRITE OUT THE DATA AT THE END OF EACH TIME STEP.                     007540
      WRITE(6,510)TMS,P1,P3,PL,P4,PR,SPS,VPS,SPJ,VPJ,APJ                007550
      WRITE(7,512)TMS,RH1,RHL3,RHG3,RH3,RHL4,RHG4,RH4                   007560
      WRITE(8,510)TMS,V1,VL3,VG3,V3,VL4,VG4,V4,EPS3,EPS4,AVENT          007570
      WRITE(9,510)TMS,RM1,RML3,RMG3,RM3,RML4,RMG4,RM4,RMSUM             007580
      WRITE(10,510)TMS,RMD13,RMD3,RMD34,RMD4,HL1,HL3,H3,HL4,H4          007590
      WRITE(11,510)TMS,CL1,CL3,CG3,C3,CL4,CG4,C4,TEMP3,TEMP4            007600
      WRITE(12,510)TMS,EL1,EL3,EG3,EL4,EG4,EKPS,EKPJ,EKL4,EKG4,ESUM     007610
      WRITE(13,510)TMS,(PML3(I),I=1,10)                                 007620
      WRITE(14,510)TMS,(PMD3(I),I=1,10)                                 007630
      WRITE(15,510)TMS,(PML4(I),I=1,10)                                 007640
      WRITE(16,510)TMS,(PMD4(I),I=1,10)                                 007650
      WRITE(17,510)TMS,U13,WE,DC                                        007660
      CALL OUTGRA                                                       007670
  510 FORMAT(11F11.3)                                                   007680
  512 FORMAT(11F11.4)                                                   007690
      RMAX(1)=AMAX1(P1,RMAX(1))                                         007700
      RMAX(2)=AMAX1(P3,RMAX(2))                                         007710
      RMAX(3)=AMAX1(PL,RMAX(3))                                         007720
      RMAX(4)=AMAX1(PR,RMAX(4))                                         007730
      RMAX(5)=AMAX1(APJ,RMAX(5))                                        007740
      RMAX(6)=AMAX1(ABS(RMSUM-RMINT)/RMINT,RMAX(6))                     007750
      RMAX(7)=AMAX1(ABS(ESUM-EINT)/EINT,RMAX(7))                        007760
      LSKIP=LSKIP+1                                                     007770
      LPAGE=LPAGE+1                                                     007780
      IF(LPAGE.EQ.45)GO TO 550                                          007790
      IF(LSKIP.LT.5)RETURN                                              007800
      LSKIP=0                                                           007810
      WRITE(6,525)                                                      007820
      WRITE(7,525)                                                      007830
      WRITE(8,525)                                                      007840
      WRITE(9,525)                                                      007850
      WRITE(10,525)                                                     007860
      WRITE(11,525)                                                     007870
      WRITE(12,525)                                                     007880
```

```
      WRITE(13,525)                                                     007890
      WRITE(14,525)                                                     007900
      WRITE(15,525)                                                     007910
      WRITE(16,525)                                                     007920
      WRITE(17,525)                                                     007930
  525 FORMAT(10X)                                                       007940
      RETURN                                                            007950
  550 CALL CAPTION                                                      007960
      LPAGE=0                                                           007970
      LSKIP=0                                                           007980
      RETURN                                                            007990
      END                                                               008000
```

```
      SUBROUTINE OUTGRA
      COMMON/EPCOM9/HUSED,NQUSED,NSTEP,NFE,NJE,NMI
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET
      COMMON/TAB2/RK1,RK2,RH1I,V1I,VPER
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4
      COMMON/TAB7/PJWT,TRAVEL
      COMMON/TAB8/PSPS,PJRS,PL,PR,DC34
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS
      COMMON/TAB13/PI
      COMMON/TAB14/DDR,ADR,BDR
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4
      COMMON/TAB18/DINC,PDDP(10),PML3(10),PMD3(10),PML4(10),PMD4(10)
      COMMON/TAB19/SIGMA,U13,WE
      COMMON/TABINT/KWRITE,KOUTD,SPEC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)
      DIMENSION COLTIT(5),PV(5),TIM(5),DAY(5)
      DATA KOUNT/1/
C  CREATE UNFORMATTED GRAPH TRAJECTORY FILE ON TAPE17.
      IF(KOUNT.GT.1)GO TO 100
C  FILE HEADINGS.
      PV(1)=9HRLGD GUN
      PV(2)=4HCODE
      PV(3)=7HVERSION
      PV(4)=9H07/24/84
      PV(5)=1H
      CALL DATE(DAY(1))
      CALL CLOCK(TIM(1))
      KOLS=107
      COLS=FLOAT(KOLS)
      DUM=0.0
      WRITE(2)TITLE,(PV(5),J=1,12),DAY,TIM,DUM,DUM,COLS,PV,DUM,
     + DUM,DUM,DUM,DUM
      DO 40 I=1,KOLS
      COLMIN=-1.E4
      COLMAX=1.E4
      DO 30 J=1,5
      COLTIT(J)=5HXXXXX
   30 CONTINUE
      WRITE(2)COLMIN,COLMAX,COLTIT
   40 CONTINUE
      KOUNT=2
  100 CONTINUE
      VPSM=VPS/100.
      VPJM=VPJ/100.
      SPJM=SPJ/100.
      CL1M=CL1/100.
      CL3M=CL3/100.
      CL4M=CL4/100.
      CG3M=CG3/100.
      CG4M=CG4/100.
```

```
      C3M=C3/100.                                                       008540
      C4M=C4/100.                                                       008590
      U13M=U13/100.                                                     008600
      WRITE(2)TMS,P1,P3,PL,P4,PR,SPS,VPSM,SPJM,VPJM,APJ,                008610
     + RH1,RHL3,RHG3,RH3,RHL4,RHG4,RH4,V1,VL3,VG3,V3,VL4,VG4,V4,        008620
     + EPS3,EPS4,                                                       008630
     + AVENT,PH1,RML3,RMG3,RM3,RML4,RMG4,RM4,RMSUM,RMD13,RMD3,RMD34,    008640
     + RMD4,HL1,HL3,H3,HL4,H4,CL1M,CL3M,CG3M,C3M,CL4M,CG4M,C4M,         008650
     + TEMP3,TEMP4,                                                     008660
     + EL1,EL3,EG3,EL4,EG4,EXPS,EXPJ,EXL4,EXG4,ESUM,                    008670
     + (PML3(I),I=1,10),(PMD3(I),I=1,10),                               008680
     + (PML4(I),I=1,10),(PMD4(I),I=1,10),                               008690
     + U13M,WF,DC                                                       008700
      RETURN                                                            008710
      END                                                               008720
```

```
      SUBROUTINE DIFFUN(N,TIME,Y,YDOT)                                  008730
      COMMON/TARINT/KWRITE,KOUTD,SPEC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)008760
      IF(TDROP(1).EQ.10HDROP1      )CALL FDROP1(N,TIME,Y,YDOT)          008750
      IF(TDROP(1).EQ.10HDROP2      )CALL FDROP2(N,TIME,Y,YDOT)          008760
      IF(TDROP(1).EQ.10HDROP3      )CALL FDROP3(N,TIME,Y,YDOT)          008770
      RETURN                                                            008740
      END                                                               008790
```

```
      SUBROUTINE FDROP1(N,TIME,Y,YDOT)                                  008800
      DIMENSION Y(N),YDOT(N)                                            008810
      COMMON/FPCOM9/NUSED,NQUSED,NSTEP,NFE,NJF,NMI                      008820
      COMMON/TAB1/G0,GAM,COV,FNER,OFFSET                                008830
      COMMON/TAB2/RK1,RK2,RH1I,V1I,VPER                                 008840
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                          008850
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                           008860
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT   008870
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     008880
      COMMON/TAB7/PJWT,TRAVEL                                           008890
      COMMON/TAB8/PSPS,PJPS,PL,PR,DC34                                  008900
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,FINT                  008910
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         008920
      COMMON/TAB11/LSKIP,LPAGE,TMS,PMAX(10)                             008930
      COMMON/TAB12/NVENT,NVENT,AVMIN,AVMAX,AK,ACS                       008940
      COMMON/TAB13/PI                                                   008950
      COMMON/TAB14/DDP,ADP,BDP                                          008960
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RHG3,RML3,RMG4,RML4              008970
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,FL3,EL4,EKL4                     008980
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            008990
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    009000
      COMMON/TAB19/SIGMA,U13,WE                                         009010
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDPOP(8),TDIS(8)009020
C     ***********************************************                   009030
C     INSTANTANEOUS BURNING MODEL.                                      009040
C     ***********************************************                   009050
C     VOLUME OF LIQUID CHAMBER.                                         009060
      V1=Y(1)                                                           009070
C     PISTON VELOCITY.                                                  009080
      VPS=Y(2)                                                          009090
      IF(MOD1.EQ.0)VPS=0.0                                              009100
C     PISTON TRAVEL                                                     009110
      SPS=Y(3)                                                          009120
C     LIQUID DENSITY.                                                   009130
      RH1=Y(4)                                                          009140
C     LIQUID PRESSURE.                                                  009150
      P1=Y(5)                                                           009160
      IF(MOD1.EQ.0)P1=0.0                                               009170
C     VOLUME OF COMBUSTION CHAMBER.                                     009180
      V3=Y(6)                                                           009190
C     DENSITY IN COMBUSTION CHAMBER.                                    009200
      RH3=Y(7)                                                          009210
C     PRESSURE IN COMBUSTION CHAMBER.                                   009220
      P3=Y(8)                                                           009230
C     VOLUME OF TUBE.                                                   009240
      V4=Y(9)                                                           009250
C     PROJECTILE VELOCITY.                                              009260
      VPJ=Y(10)                                                         009270
C     PROJECTILE TRAVEL                                                 009280
      SPJ=Y(11)                                                         009290
C     DENSITY IN TUBE                                                   009300
      RH4=Y(12)                                                         009310
C     AVERAGE PRESSURE IN THE TUBE.                                     009320
      P4=Y(13)                                                          009330
C     FIND THE DISCHARGE COEFFICIENT.                                   009340
      IF(TDIS(1).EQ.10HDIS1      )CALL DIS1(2)                          009350
      IF(TDIS(1).EQ.10HDIS2      )CALL DIS2(2)                          009360
```

```
C  FIND THE PISTON RESISTANCE PRESSURE.
      CALL PISRES(2,SPS,PSRS)
C  FIND THE PROJECTILE RESISTANCE PRESSURE.
      CALL PROJRES(2,SPJ,PJRS)
      IF(TVENT(1).EQ.10HVENT1     )CALL VENT1(2)
      IF(TVENT(1).EQ.10HVENT2     )CALL VENT2(2)
      IF(TVENT(1).EQ.10HVENT3     )CALL VENT3(2)
C  RM1 IS THE MASS IN REGION 1.
      RM1=RH1*V1
C  RM3 IS THE MASS IN REGION 3.
      RM3=RH3*V3
C  RM4 IS THE MASS IN REGION 4.
      RM4=RH4*V4
C  RMSUM IS THE TOTAL MASS
      RMSUM=RM1+RM3+RM4
C  PRESSURE AT BASE OF PROJECTILE.
      PR=(P4+PJRS*RM4/(3.0*PJWT))/(1.0+RM4/(3.0*PJWT))
      IF(PR.LT.PJRS)PR=P4
C  PRESSURE AT LEFT OF TUBE.
      PL=PR*(1.0+RM4/(2.0*PJWT)) - PJRS*RM4/(2.0*PJWT)
      IF(PR.LT.PJRS)PL=P4
C  INTERNAL ENERGY PER GRAM FOR GAS IN REGION 3.
      ENER3=P3*(1.0-COV*RM3)/(RH3*(GAM-1.0))
C  AVERAGE INTERNAL ENERGY PER RAM FOR GAS IN REGION 4.
      ENER4=P4*(1.0-COV*RH4)/(RH4*(GAM-1.0))
      IF(MOD4.EQ.1)GO TO 30
      PL=P4
      PR=P4
30    CONTINUE
C  C3 IS THE SPEED OF SOUND IN THE COMBUSTION CHAMBER GAS.
      C3=SQRT(G0*GAM*P3/((1.0-COV*RH3)*RH3))
C  C4 IS THE SPEED OF SOUND IN THE TUBE GAS.
      C4=SQRT(G0*GAM*P4/((1.0-COV*RH4)*RH4))
C  CL1 IS THE SPEED OF SOUND IN THE LIQUID CHAMBER.
      CL1=SQRT(G0*(RK1+RK2*P1)/RH1)
C  RMD13 IS THE MASS FLUX FROM REGION 1 TO REGION 3.
      DLP=P1-P3
      IF(DLP.LT.0.0.AND.MBACK.EQ.1)RMD13=-DC*AVENT*SQRT(-2.0*RH3*G0*DLP)
      IF(DLP.LT.0.0.AND.MBACK.EQ.0)RMD13=0.0
      IF(DLP.GE.0.0)RMD13=DC*AVENT*SQRT(2.0*RH1*G0*DLP)
      IF(MOD1.EQ.0)RMD13=0.0
C  LIQUID INJECTION VELOCITY INTO REGION 3.
      U13=RMD13/(RH1*AVENT)
C  RMD34 IS THE MASS FLUX FROM REGION 3 TO REGION 4.
      DLP=P3-PL
      IF(DLP.LT.0.0)RMD34=-DC34*A4*SQRT(-2.0*RH4*G0*DLP)
      IF(DLP.GE.0.0)RMD34=DC34*A4*SQRT(2.0*RH3*G0*DLP)
      IF(MOD4.EQ.0)RMD34=0.0
C  EL1 IS THE TOTAL ENERGY IN REGION 1.
      EL1=ENER*RM1
C  EG3 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 3.
      EG3=ENER3*RM3
C  EG4 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 4.
      EG4=ENER4*RM4
C  EKPS IS THE KINETIC ENERGY OF THE PISTON.
      EKPS=0.5*PSWT*VPS*VPS/G0
C  EKPJ IS THE KINETIC ENERGY OF THE PROJECTILE.
```

```
      EKPJ=0.5*PJWT*VPJ*VPJ/G0
C  EKG4 IS THE KINETIC ENERGY OF THE GAS IN REGION 4.
C  ASSUME THAT THE DENSITY IS CONSTANT.
C  ASSUME THAT THE VELOCITY IS A LINEAR FUNCTION.
C  SO USE THE AVERAGE VELOCITY TO COMPUTE THE KINETIC ENERGY.
      EKG4=0.0
      IF(MOD4.EQ.0)GO TO 32
      UL=0.0
      UR=VPJ
      U=0.5*(UL+UR)
      EKG4=0.5*RM4*UR*UR/(3.0*G0)
   32 CONTINUE
C  ESUM IS THE SUM OF THE ENERGIES.
      ESUM=EL1+EG3+EG4+EKPS+EKPJ+EKG4
C  TEMP3 IS THE GAS TEMPERATURE IN REGION 3.
      TEMP3=P3*(1.0-COV*RM3)/(RM3*(GAM-1.0)*CV)
C  TEMP4 IS THE AVERAGE GAS TEMPERATURE IN REGION 4.
      TEMP4=P4*(1.0-COV*RM4)/(RM4*(GAM-1.0)*CV)
C  HL1 IS THE ENTHALPY OF THE LIQUID IN REGION 1.
      HL1=ENER + P1/RH1
C  H3 IS THE ENTHALPY OF THE GAS IN REGION 3.
      H3=CP*TEMP3 + P3*COV
C  H4 IS THE ENTHALPY OF THE GAS IN REGION 4.
      H4=CP*TEMP4 + P4*COV
C  ***********************************************
      IF(MOD1.EQ.0)GO TO 50
      IF(MOD4.EQ.0)GO TO 40
C  REGION 1 IS OPEN.
C  REGION 4 IS OPEN.
      YDOT(1)=-VPS*A1
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR
      FORC=P3*(A3-AVENT) - P1*(A1-AVENT)
      FORR=PSRS*(A1-AVENT)
      IF(TVENT(1).EQ.10HVENT3      )FORC=P3*(A3-AHOLE)-P1*(A1-AHOLE)
      IF(TVENT(1).EQ.10HVENT3      )FORR=PSRS*(A1-AHOLE)
      IF(FORC.GE.FORR)YDOT(2)=(G0/PSWT)*(FORC-FORR)
      IF(FORC.LT.FORR)YDOT(2)=0.0
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC
      YDOT(3)=VPS
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1
      YDOT(5)=CL1*CL1*YDOT(4)/G0
      YDOT(6)=VPS*A3
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR
      YDOT(7)=-RH3*YDOT(6)/V3 + RMD13/V3 -RMD34/V3
      YDOT(8)=C3*C3*YDOT(7)/G0
      IF(RMD13.GE.0.0)YDOT(8)=YDOT(8)+RMD13*(HL1-H3)*(GAM-1.0)/
     * (V3-COV*RM3)
      IF(RMD34.LE.0.0)YDOT(8)=YDOT(8)-RMD34*(H4-H3)*(GAM-1.0)/
     * (V3-COV*RM3)
      YDOT(9)=VPJ*A4
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.09*146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4
```

```
      YDOT(13)=C4*C4*YDOT(12)/G0                                        010510
      IF(RMD34.GE.0.0)YDOT(13)=YDOT(13)+RMD34*(H3-H4)*(GAM-1.0)/        010520
     * (V4-COV*RM4)                                                     010530
      RETURN                                                            010540
   40 CONTINUE                                                          010550
C  REGION 1 IS OPEN.                                                    010560
C  REGION 4 IS CLOSED.                                                  010570
C  VIEW REGION 3 AND REGION 4 AS ONE REGION.                            010580
      YDOT(1)=-VPS*A1                                                   010590
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR                010600
      FORC=P3*(A3-AVENT) - P1*(A1-AVENT)                                010610
      FORR=PSRS*(A1-AVENT)                                              010620
      IF(TVENT(1).EQ.10HVENT3      )FORC=P3*(A3-AHOLE)-P1*(A1-AHOLE)    010630
      IF(TVENT(1).EQ.10HVENT3      )FORR=PSRS*(A1-AHOLE)                010640
      IF(FORC.GE.FORR)YDOT(2)=(G0/PSWT)*(FORC-FORR)                     010650
      IF(FORC.LT.FORR)YDOT(2)=0.0                                       010660
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC                             010670
      YDOT(3)=VPS                                                       010680
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0                      010690
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0                      010700
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1                                010710
      YDOT(5)=CL1*CL1*YDOT 4)/G0                                        010720
      YDOT(6)=VPS*A3                                                    010730
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR                010740
      YDOT(9)=VPJ*A4                                                    010750
      V34=V3+V4                                                         010760
      RM34=RM3+RM4                                                      010770
      YDOT(7)=-RH3*(YDOT(6)+YDOT(9))/V34 + RMD13/V34                    010780
      YDOT(8)=C3*C3*YDOT(7)/G0                                          010790
      IF(RMD13.GE.0.0)YDOT(8)=YDOT(8)+RMD13*(HL1-H3)*(GAM-1.0)/         010800
     * (V34-COV*RM34)                                                   010810
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT                       010820
      IF(PR.LT.PJRS)YDOT(10)=0.0                                        010830
      APJ=YDOT(10)/(.098146*1.0E7)                                      010840
      YDOT(11)=VPJ                                                      010850
      YDOT(12)=YDOT(7)                                                  010860
      YDOT(13)=YDOT(8)                                                  010870
      RETURN                                                            010880
   50 CONTINUE                                                          010890
      IF(MOD4.EQ.1)GO TO 60                                             010900
C  REGION 1 IS CLOSED.                                                  010910
C  REGION 4 IS CLOSED.                                                  010920
      YDOT(1)=0.0                                                       010930
      YDOT(2)=0.0                                                       010940
      YDOT(3)=0.0                                                       010950
      YDOT(4)=0.0                                                       010960
      YDOT(5)=0.0                                                       010970
      YDOT(6)=0.0                                                       010980
      V34=V3+V4                                                         010990
      RM34=RM3+RM4                                                      011000
      YDOT(9)=VPJ*A4                                                    011010
      YDOT(7)=-RH3*(YDOT(6)+YDOT(9))/V34                                011020
      YDOT(8)=C3*C3*YDOT(7)/G0                                          011030
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT                       011040
      IF(PR.LT.PJRS)YDOT(10)=0.0                                        011050
      APJ=YDOT(10)/(.098146*1.0E7)                                      011060
      YDOT(11)=VPJ                                                      011070
```

```
      YDOT(12)=YDOT(7)                                                  011080
      YDOT(13)=YDOT(8)                                                  011090
      RETURN                                                            011100
   60 CONTINUE                                                          011110
C  REGION 1 IS CLOSED.                                                  011120
C  REGION 4 IS OPEN.                                                    011130
      YDOT(1)=0.0                                                       011140
      YDOT(2)=0.0                                                       011150
      YDOT(3)=0.0                                                       011160
      YDOT(4)=0.0                                                       011170
      YDOT(5)=0.0                                                       011180
      YDOT(6)=0.0                                                       011190
      YDOT(7)=-RH3*YDOT(6)/V3 -RMD34/V3                                 011200
      YDOT(8)=C3*C3*YDOT(7)/G0                                          011210
      IF(RMD34.LE.0.0)YDOT(8)=YDOT(8)-RMD34*(H4-H3)*(GAM-1.0)/          011220
     * (V3-COV*RM3)                                                     011230
      YDOT(9)=VPJ*A4                                                    011240
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT                       011250
      IF(PR.LT.PJRS)YDOT(10)=0.0                                        011260
      APJ=YDOT(10)/(.098146*1.0E7)                                      011270
      YDOT(11)=VPJ                                                      011280
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4                               011290
      YDOT(13)=C4*C4*YDOT(12)/G0                                        011300
      IF(RMD34.GE.0.0)YDOT(13)=YDOT(13)+RMD34*(H3-H4)*(GAM-1.0)/        011310
     * (V4-COV*RM4)                                                     011320
      RETURN                                                            011330
  100 CONTINUE                                                          011340
      RETURN                                                            011350
      END                                                               011360
```

```
      SUBROUTINE FDROP2(N,TIME,Y,YDOT)                                  011370
      DIMENSION Y(N),YDOT(N)                                            011380
      COMMON/EPCOM9/NUSED,NOUSED,NSTEP,NFE,NJE,NMI                      011390
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                                011400
      COMMON/TAB2/RK1,RK2,RHII,V1I,VPER                                 011410
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                          011420
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                           011430
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT   011440
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4     011450
      COMMON/TAB7/FJWT,TRAVEL                                           011460
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                  011470
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT                  011480
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4         011490
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)                             011500
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                       011510
      COMMON/TAB13/PI                                                   011520
      COMMON/TAB14/DDR,ADR,BDR                                          011530
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4              011540
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4                     011550
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                            011560
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)    011570
      COMMON/TAB19/SIGMA,U13,WE                                         011580
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)011590
C     ****************************************                          011600
C     DROPLET BURNING - CONSTANT SIZE DROPLETS                          011610
C     ****************************************                          011620
C     VOLUME OF LIQUID CHAMBER.                                         011630
      V1=Y(1)                                                           011640
C     PISTON VELOCITY.                                                  011650
      VPS=Y(2)                                                          011660
      IF(MOD1.EQ.0)VPS=0.0                                              011670
C     PISTON TRAVEL                                                     011680
      SPS=Y(3)                                                          011690
C     LIQUID DENSITY.                                                   011700
      RH1=Y(4)                                                          011710
C     LIQUID PRESSURE.                                                  011720
      P1=Y(5)                                                           011730
      IF(MOD1.EQ.0)P1=0.0                                               011740
C     VOLUME OF COMBUSTION CHAMBER.                                     011750
      V3=Y(6)                                                           011760
C     DENSITY IN COMBUSTION CHAMBER.                                    011770
      RH3=Y(7)                                                          011780
C     PRESSURE IN COMBUSTION CHAMBER.                                   011790
      P3=Y(8)                                                           011800
C     VOLUME OF TUBE.                                                   011810
      V4=Y(9)                                                           011820
C     PROJECTILE VELOCITY.                                              011830
      VPJ=Y(10)                                                         011840
C     PROJECTILE TRAVEL                                                 011850
      SPJ=Y(11)                                                         011860
C     DENSITY IN TUBE                                                   011870
      RH4=Y(12)                                                         011880
C     AVERAGE PRESSURE IN THE TUBE.                                     011890
      P4=Y(13)                                                          011900
C     LIQUID DENSITY IN REGION 3.                                       011910
      RHL3=Y(14)                                                        011920
C     LIQUID MASS IN REGION 3.                                          011930
```

```
      RML3=Y(15)                                                        011940
C  LIQUID DENSITY IN REGION 4.                                          011950
      RHL4=Y(16)                                                        011960
C  LIQUID MASS IN REGION 4.                                             011970
      RML4=Y(17)                                                        011980
C  FIND THE DISCHARGE COEFFICIENT.                                      011990
      IF(TDIS(1).EQ.10HDIS1      )CALL DIS1(2)                          012000
      IF(TDIS(1).EQ.10HDIS2      )CALL DIS2(2)                          012010
C  FIND THE PISTON RESISTANCE PRESSURE.                                 012020
      CALL PISRES(2,SPS,PSRS)                                           012030
C  FIND THE PROJECTILE RESISTANCE PRESSURE.                             012040
      CALL PROJRES(2,SPJ,PJRS)                                          012050
      IF(TVENT(1).EQ.10HVENT1    )CALL VENT1(2)                         012060
      IF(TVENT(1).EQ.10HVENT2    )CALL VENT2(2)                         012070
      IF(TVENT(1).EQ.10HVENT3    )CALL VENT3(2)                         012080
C  RM1 IS THE MASS IN REGION 1.                                         012090
      RM1=RH1*V1                                                        012100
C  VL3 IS THE LIQUID VOLUME OF REGION 3.                                012110
      VL3=RML3/RHL3                                                     012120
C  VG3 IS THE GAS VOLUME IN REGION 3.                                   012130
      VG3=V3-VL3                                                        012140
C  EPS3 IS THE POROSITY IN REGION 3.                                    012150
      EPS3=VG3/V3                                                       012160
C  RHG3 IS THE GAS DENSITY IN REGION 3.                                 012170
      RHG3=(RH3-(1.0-EPS3)*RHL3)/EPS3                                   012180
C  RMG3 IS THE GAS MASS IN REGION 3.                                    012190
      RMG3=RHG3*VG3                                                     012200
C  RM3 IS THE MASS IN REGION 3.                                         012210
      RM3=RMG3+RML3                                                     012220
C  VL4 IS THE LIQUID VOLUME OF REGION 4.                                012230
      VL4=RML4/RHL4                                                     012240
C  VG4 IS THE GAS VOLUME IN REGION 4.                                   012250
      VG4=V4-VL4                                                        012260
C  EPS4 IS THE POROSITY OF REGION 4.                                    012270
      EPS4=EPS3                                                         012280
      IF(V4.GT.1.0E-4)EPS4=VG4/V4                                       012290
C  RHG4 IS THE GAS DENSITY OF REGION 4.                                 012300
      RHG4=(RH4-(1.0-EPS4)*RHL4)/EPS4                                   012310
C  RMG4 IS THE GAS MASS IN REGION 4.                                    012320
      RMG4=RHG4*VG4                                                     012330
C  RM4 IS THE MASS IN REGION 4.                                         012340
      RM4=RMG4+RML4                                                     012350
C  RMSUM IS THE TOTAL MASS                                              012360
      RMSUM=RM1+RM3+RM4                                                 012370
C  PRESSURE AT BASE OF PROJECTILE.                                      012380
      PR=(P4+PJRS*RM4/(3.0*PJWT))/(1.0+RM4/(3.0*PJWT))                  012390
      IF(PR.LT.PJRS)PR=P4                                               012400
C  PRESSURE AT LEFT OF TURE.                                            012410
      PL=PR*(1.0+RM4/(2.0*PJWT)) - PJRS*RM4/(2.0*PJWT)                  012420
      IF(PR.LT.PJRS)PL=P4                                               012430
C  INTERNAL ENERGY PER GRAM FOR GAS IN REGION 3.                        012440
      ENER3=P3*(1.0-COV*RHG3)/(RHG3*(GAM-1.0))                          012450
C  AVERAGE INTERNAL ENERGY PER RAM FOR GAS IN REGION 4.                 012460
      ENER4=P4*(1.0-COV*RHG4)/(RHG4*(GAM-1.0))                          012470
      IF(MOD4.EQ.1)GO TO 30                                             012480
      PL=P4                                                             012490
      PR=P4                                                             012500
```

```
   30    CONTINUE
C  CG3 IS THE SPEED OF SOUND IN THE COMBUSTION CHAMBER GAS.
      CG3=SQRT(G0*GAM*P3/((1.0-COV*RHG3)*RHG3))
C  CG4 IS THE SPEED OF SOUND IN THE TUBE GAS.
      CG4=SQRT(G0*GAM*P4/((1.0-COV*RHG4)*RHG4))
C  CL1 IS THE SPEED OF SOUND IN THE LIQUID CHAMBER.
      CL1=SQRT(G0*(RK1+RK2*P1)/RH1)
C  CL3 IS THE SPEED OF SOUND IN THE LIQUID IN REGION 3.
      CL3=SQRT(G0*(RK1+RK2*P3)/RHL3)
C  CL4 IS THE SPEED OF SOUND IN THE LIQUID IN REGION 4.
      CL4=SQRT(G0*(RK1+RK2*P4)/RHL4)
C  C3 IS THE MIXTURE SPEED OF SOUND IN REGION 3.
      C3=SQRT(1.0/(RH3*(EPS3/(RHG3*CG3*CG3) +
     + (1.0-EPS3)/(RHL3*CL3*CL3))))
C  C4 IS THE MIXTURE SPEED OF SOUND IN REGION 4.
      C4=SQRT(1.0/(RH4*(EPS4/(RHG4*CG4*CG4) +
     + (1.0-EPS4)/(RHL4*CL4*CL4))))
C  RMD13 IS THE MASS FLUX FROM REGION 1 TO REGION 3.
      DLP=P1-P3
      IF(DLP.LT.0.0.AND.MBACK.EQ.1)RMD13=-DC*AVENT*SQRT(-2.0*RH3*G0*DLP)
      IF(DLP.LT.0.0.AND.MBACK.EQ.0)RMD13=0.0
      IF(DLP.GE.0.0)RMD13=DC*AVENT*SQRT(2.0*RH1*G0*DLP)
      IF(MOD1.EQ.0)RMD13=0.0
C  LIQUID INJECTION VELOCITY INTO REGION 3.
      U13=RMD13/(RH1*AVENT)
C  WEBER NUMBER
      WE=RHG3*U13*U13*DDR/(2.0*SIGMA)
C  RMD34 IS THE MASS FLUX FROM REGION 3 TO REGION 4.
      DLP=P3-PL
      IF(DLP.LT.0.0)RMD34=-DC34*A4*SQRT(-2.0*RH4*G0*DLP)
      IF(DLP.GE.0.0)RMD34=DC34*A4*SQRT(2.0*RH3*G0*DLP)
      IF(MOD4.EQ.0)RMD34=0.0
C  EL1 IS THE TOTAL ENERGY IN REGION 1.
      EL1=ENER*RM1
C  EL3 IS THE LIQUID ENERGY IN REGION 3.
      EL3=ENER*RML3
C  EG3 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 3.
      EG3=ENER3*RMG3
C  EL4 IS THE LIQUID ENERGY IN REGION 4.
      EL4=ENER*RML4
C  EG4 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 4.
      EG4=ENER4*RMG4
C  EKPS IS THE KINETIC ENERGY OF THE PISTON.
      EKPS=0.5*PSWT*VPS*VPS/G0
C  EKPJ IS THE KINETIC ENERGY OF THE PROJECTILE.
      EKPJ=0.5*PJWT*VPJ*VPJ/G0
C  EKG4 IS THE KINETIC ENERGY OF THE GAS IN REGION 4.
C  ASSUME THAT THE DENSITY IS CONSTANT.
C  ASSUME THAT THE VELOCITY IS A LINEAR FUNCTION.
C  SO USE THE AVERAGE VELOCITY TO COMPUTE THE KINETIC ENERGY.
      EKG4=0.0
      IF(MOD4.EQ.0)GO TO 32
      UL=0.0
      UR=VPJ
      U=0.5*(UL+UR)
      EKG4=0.5*RMG4*UR*UR/(3.0*G0)
C  EKL4 IS THE KINETIC ENERGY OF THE LIQUID IN REGION 4.
```

```
C  IGNORE DRAG.
      EKL4=0.5*RML4*UR*UR/(3.0*G0)
32    CONTINUE
C  ESUM IS THE SUM OF THE ENERGIES.
      ESUM=EL1+EL3+EG3+EL4+EG4+EKPS+EKPJ+EKG4+EKL4
C  TEMP3 IS THE GAS TEMPERATURE IN REGION 3.
      TEMP3=P3*(1.0-COV*RHG3)/(RHG3*(GAM-1.0)*CV)
C  TEMP4 IS THE AVERAGE GAS TEMPERATURE IN REGION 4.
      TEMP4=P4*(1.0-COV*RHG4)/(RHG4*(GAM-1.0)*CV)
C  HL1 IS THE ENTHALPY OF THE LIQUID IN REGION 1.
      HL1=ENER + P1/RH1
C  HL3 IS THE ENTHALPY OF THE LIQUID IN REGION 3.
      HL3=ENER + P3/RHL3
C  HL4 IS THE ENTHALPY OF THE LIQUID IN REGION 4.
      HL4=ENER + P4/RHL4
C  H3 IS THE ENTHALPY OF THE GAS IN REGION 3.
      H3=CP*TEMP3 + P3*COV
C  H4 IS THE ENTHALPY OF THE GAS IN REGION 4.
      H4=CP*TEMP4 + P4*COV
C  ***********************************************
C  DROPLET FORMATION.
C  ASSUME THAT THE GAS CANNOT FLOW FROM REGION 4 TO REGION 3.
      IF(RMD34.LT.0.0)RMD34=0.0
C  RMD3 IS THE RATE OF CHANGE OF LIQUID TO GAS IN REGION 3.
      RMD3=RML3*(6.0/DDR)*ADR*(P3**BDR)
C  RMD4 IS THE RATE OF CHANGE OF LIQUID TO GAS IN REGION 4.
      RMD4=RML4*(6.0/DDR)*ADR*(P4**BDR)
      IF(MOD1.EQ.0)GO TO 150
      IF(MOD4.EQ.0)GO TO 140
C  REGION 1 IS OPEN.
C  REGION 4 IS OPEN.
      YDOT(1)=-VPS*A1
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR
      FORC=P3*(A3-AVENT) - P1*(A1-AVENT)
      FORR=PSRS*(A1-AVENT)
      IF(TVENT(1).EQ.10HVENT3      )FORC=P3*(A3-AHOLE)-P1*(A1-AHOLE)
      IF(TVENT(1).EQ.10HVENT3      )FORR=PSRS*(A1-AHOLE)
      IF(FORC.GE.FORR)YDOT(2)=(G0/PSWT)*(FORC-FORR)
      IF(FORC.LT.FORR)YDOT(2)=0.0
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC
      YDOT(3)=VPS
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1
      YDOT(5)=CL1*CL1*YDOT(4)/G0
      YDOT(6)=VPS*A3
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR
      YDOT(7)=-RH3*YDOT(6)/V3 + RMD13/V3 -RMD34/V3
      YDOT(8)=(RH3*C3*C3/G0) * (-YDOT(6)/V3  +
     + RMD13/(RHL3*V3) + (RMD3/(RHG3*V3)) *
     + (1.0 - RHG3/RHL3 + G0*(HL3-H3)*(GAM-1.0)/
     + (CG3*CG3*(1.0-COV*RHG3))) - RMD34/RH3)
      YDOT(9)=VPJ*A4
      IF(PP.GE.PJRS)YDOT(10)=(PP-PJRS)*A4*G0/PJWT
      IF(PP.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098140*1.057)
      YDOT(11)=VPJ
```

```
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4                                013650
      YDOT(13)=(RH4*C4*C4/(G0*V4)) * (-YDOT(9) +                         013660
     + (RMD4/RHG4) * (1.0-RHG4/RHL4 +                                    013670
     + G0*(HL4-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) +                 013680
     + (RMD34/(RHG4*RH3)) * (RML3*RHG4/RHL4 + RMG3 +                     013690
     + G0*RMG3*(H3-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) )             013700
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)                                      013710
      YDOT(15)=RMD13 - RML3*RMD34/RM3 - RMD3                             013720
      YDOT(16)=G0*YDOT(13)/(CL4*CL4)                                     013730
      YDOT(17)=RML3*RMD34/RM3 - RMD4                                     013740
      RETURN                                                             013750
  140 CONTINUE                                                           013760
C  REGION 1 IS OPEN.                                                     013770
C  REGION 4 IS CLOSED.                                                   013780
C  VIEW REGION 3 AND REGION 4 AS ONE REGION.                             013790
      YDOT(1)=-VPS*A1                                                    013900
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR                 013910
      FORC=P3*(A3-AVENT) - P1*(A1-AVENT)                                 013820
      FORR=PSRS*(A1-AVENT)                                               013830
      IF(TVENT(1).EQ.10HVENT3      )FORC=P3*(A3-AHOLE)-P1*(A1-AHOLE)     013840
      IF(TVENT(1).EQ.10HVENT3      )FORR=PSRS*(A1-AHOLE)                 013850
      IF(FORC.GE.FORR)YDOT(2)=(G0/PSWT)*(FORC-FORR)                      013860
      IF(FORC.LT.FORR)YDOT(2)=0.0                                        013870
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC                              013880
      YDOT(3)=VPS                                                        013990
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0                       013900
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0                       013910
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1                                 013920
      YDOT(5)=CL1*CL1*YDOT(4)/G0                                         013930
      YDOT(6)=VPS*A3                                                     013940
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR                 013950
      YDOT(9)=VPJ*A4                                                     013960
      V34=V3+V4                                                          013970
      RM34=RM3+RM4                                                       013980
      YDOT(7)=-RH3*(YDOT(6)+YDOT(9))/V34 + RMD13/V34                     013990
      YDOT(8)=(RH3*C3*C3/G0)*(-(YDOT(6)+YDOT(9))/V34 +                   014000
     + (RMD13/(RML3*V34)) + ((RMD3+RMD4)/(RMG3*V34))*                    014010
     + (1.0 - RMG3/RML3 + G0*(HL3-H3)*(GAM-1.0)/                         014020
     + (CG3*CG3*(1.0-COV*RHG3))) )                                       014030
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT                        014040
      IF(PR.LT.PJRS)YDOT(10)=0.0                                         014050
      APJ=YDOT(10)/(.098146*1.0E7)                                       014060
      YDOT(11)=VPJ                                                       014070
      YDOT(12)=YDOT(7)                                                   014080
      YDOT(13)=YDOT(8)                                                   014090
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)                                      014100
      YDOT(15)=(RMD13-RMD3-RMD4)*V3/V34                                  014110
      YDOT(16)=YDOT(14)                                                  014120
      YDOT(17)=YDOT(15)*V4/V3                                            014130
      RETURN                                                             014140
  150 CONTINUE                                                           014150
      IF(MOD4.EQ.1)GO TO 160                                             014160
C  REGION 1 IS CLOSED.                                                   014170
C  REGION 4 IS CLOSED.                                                   014180
      YDOT(1)=0.0                                                        014190
      YDOT(2)=0.0                                                        014200
      YDOT(3)=0.0                                                        014210
```

```
      YDOT(4)=0.0
      YDOT(5)=0.0
      YDOT(6)=0.0
      V34=V3+V4
      YDOT(9)=VPJ*A4
      YDOT(7)=-RH3*(YDOT(6)+YDOT(9))/V34
      YDOT(8)=(RH3*C3*C3/G0)*(-(YDOT(6)+YDOT(9))/V3 +
     + ((RMD3+RMD4)/(RHG3*V34))*(1.0 - RHG3/RHL3 +
     + G0*(HL3-H3)*(GAM-1.0)/(CG3*CG3*(1.0-COV*RHG3))) )
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=YDOT(7)
      YDOT(13)=YDOT(8)
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=(-RMD3-RMD4)*V3/V34
      YDOT(16)=YDOT(14)
      YDOT(17)=YDOT(15)*V4/V3
      RETURN
  160 CONTINUE
C  REGION 1 IS CLOSED.
C  REGION 4 IS OPEN.
      YDOT(1)=0.0
      YDOT(2)=0.0
      YDOT(3)=0.0
      YDOT(4)=0.0
      YDOT(5)=0.0
      YDOT(6)=0.0
      YDOT(7)=-RH3*YDOT(6)/V3 -RMD34/V3
      YDOT(8)=(RH3*C3*C3/G0)*(-YDOT(6)/V3 +
     + (RMD3/(RHG3*V3))*(1.0 - RHG/RHL3 +
     + G0*(HL3-H3)*(GAM-1.0)/(CG3*CG3*(1.0-COV*RHG3))) -
     + RMD34/RM3)
      YDOT(9)=VPJ*A4
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4
      YDOT(13)=(RH4*C4*C4/(G0*V4)) * (-YDOT(9) +
     + (RMD4/RHG4) * (1.0-RHG4/RHL4 +
     + G0*(HL4-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) +
     + (RMD34/(RHG4*RM3)) * (RML3*RHG4/RHL4 + RMG3 +
     + G0*RMG3*(H3-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) )
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=-RML3*RMD34/RM3 - RMD3
      YDOT(16)=G0*YDOT(13)/(CL4*CL4)
      YDOT(17)=RML3*RMD34/RM3 - RMD4
      RETURN
      END
```

```
      SUBROUTINE FDROP3(N,TIME,Y,YDOT)
      DIMENSION Y(N),YDOT(N)
      COMMON/EPCOM9/HUSED,NQUSED,NSTEP,NFE,NJE,NMI
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET
      COMMON/TAB2/RK1,RK2,RHII,VII,VPER
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT
      COMMON/TAB6/V1,VPS,SPS,RHI,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4
      COMMON/TAB7/PJWT,TRAVEL
      COMMON/TAB8/PSPS,PJPS,PL,PR,DC34
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,EKG4,ESUM,EINT
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS
      COMMON/TAB13/PI
      COMMON/TAB14/DDR,ADR,BDR
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)
      COMMON/TAB19/SIGMA,U13,WE
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)
C  VOLUME OF LIQUID CHAMBER.
      V1=Y(1)
C  PISTON VELOCITY.
      VPS=Y(2)
      IF(MOD1.EQ.0)VPS=0.0
C  PISTON TRAVEL
      SPS=Y(3)
C  LIQUID DENSITY.
      RH1=Y(4)
C  LIQUID PRESSURE.
      P1=Y(5)
      IF(MOD1.EQ.0)P1=0.0
C  VOLUME OF COMBUSTION CHAMBER.
      V3=Y(6)
C  DENSITY IN COMBUSTION CHAMBER.
      RH3=Y(7)
C  PRESSURE IN COMBUSTION CHAMBER.
      P3=Y(8)
C  VOLUME OF TUBE.
      V4=Y(9)
C  PROJECTILE VELOCITY.
      VPJ=Y(10)
C  PROJECTILE TRAVEL
      SPJ=Y(11)
C  DENSITY IN TUBE
      RH4=Y(12)
C  AVERAGE PRESSURE IN THE TUBE.
      P4=Y(13)
C  LIQUID DENSITY IN REGION 3.
      RHL3=Y(14)
C  LIQUID MASS IN REGION 3.
      RML3=Y(15)
C  LIQUID DENSITY IN REGION 4.
      RHL4=Y(16)
```

```
C  LIQUID MASS IN REGION 4.
      RML4=Y(17)
C  PARTIONED LIQUID MASS IN REGION 3.
C  PARTIONED LIQUID MASS IN REGION 4.
      DO 2 I=1,10
      IP=I+17
      PML3(I)=Y(IP)
      IP=I+27
    2 PML4(I)=Y(IP)
    5 CONTINUE
C  FIND THE DISCHARGE COEFFICIENT.
      IF(TDIS(1).EQ.10HDIS1      )CALL DIS1(2)
      IF(TDIS(1).EQ.10HDIS2      )CALL DIS2(2)
C  FIND THE PISTON RESISTANCE PRESSURE.
      CALL PISRES(2,SPS,PSRS)
C  FIND THE PROJECTILE RESISTANCE PRESSURE.
      CALL PROJRES(2,SPJ,PJRS)
C  FIND THE VENT AREA.
      IF(TVENT(1).EQ.10HVENT1     )CALL VENT1(2)
      IF(TVENT(1).EQ.10HVENT2     )CALL VENT2(2)
      IF(TVENT(1).EQ.10HVENT3     )CALL VENT3(2)
C  RM1 IS THE MASS IN REGION 1.
      RM1=RH1*V1
C  VL3 IS THE LIQUID VOLUME OF REGION 3.
      VL3=RML3/RHL3
C  VG3 IS THE GAS VOLUME IN REGION 3.
      VG3=V3-VL3
C  EPS3 IS THE POROSITY IN REGION 3.
      EPS3=VG3/V3
C  RHG3 IS THE GAS DENSITY IN REGION 3.
      RHG3=(RH3-(1.0-EPS3)*RHL3)/EPS3
C  RMG3 IS THE GAS MASS IN REGION 3.
      RMG3=RHG3*VG3
C  RM3 IS THE MASS IN REGION 3.
      RM3=RMG3+RML3
C  VL4 IS THE LIQUID VOLUME OF REGION 4.
      VL4=RML4/RHL4
C  VG4 IS THE GAS VOLUME IN REGION 4.
      VG4=V4-VL4
C  EPS4 IS THE POROSITY OF REGION 4.
      EPS4=EPS3
      IF(V4.GT.1.0E-4)EPS4=VG4/V4
C  RHG4 IS THE GAS DENSITY OF REGION 4.
      RHG4=(RH4-(1.0-EPS4)*RHL4)/EPS4
C  RMG4 IS THE GAS MASS IN REGION 4.
      RMG4=RHG4*VG4
C  RM4 IS THE MASS IN REGION 4.
      RM4=RMG4+RML4
C  RMSUM IS THE TOTAL MASS
      RMSUM=RM1+RM3+RM4
C  PRESSURE AT BASE OF PROJECTILE.
      PR=(P4+PJRS*RM4/(3.0*PJWT))/(1.0+RM4/(3.0*PJWT))
      IF(PR.LT.PJRS)PR=P4
C  PRESSURE AT LEFT OF TUBE.
      PL=PR*(1.0+RM4/(2.0*PJWT)) - PJRS*RM4/(2.0*PJWT)
      IF(PR.LT.PJRS)PL=P4
C  INTERNAL ENERGY PER GRAM FOR GAS IN REGION 3.
```

```
      ENER3=P3*(1.0-COV*RHG3)/(RHG3*(GAM-1.0))
C  AVERAGE INTERNAL ENERGY PER RAM FOR GAS IN REGION 4.
      ENER4=P4*(1.0-COV*RHG4)/(RHG4*(GAM-1.0))
      IF(MOD4.EQ.1)GO TO 30
      PL=P4
      PR=P4
30    CONTINUE
C  CG3 IS THE SPEED OF SOUND IN THE COMBUSTION CHAMBER GAS.
      CG3=SQRT(G0*GAM*P3/((1.0-COV*RHG3)*RHG3))
C  CG4 IS THE SPEED OF SOUND IN THE TUBE GAS.
      CG4=SQRT(G0*GAM*P4/((1.0-COV*RHG4)*RHG4))
C  CL1 IS THE SPEED OF SOUND IN THE LIQUID CHAMBER.
      CL1=SQRT(G0*(RK1+RK2*P1)/RH1)
C  CL3 IS THE SPEED OF SOUND IN THE LIQUID IN REGION 3.
      CL3=SQRT(G0*(RK1+RK2*P3)/RHL3)
C  CL4 IS THE SPEED OF SOUND IN THE LIQUID IN REGION 4.
      CL4=SQRT(G0*(RK1+RK2*P4)/RHL4)
C  C3 IS THE MIXTURE SPEED OF SOUND IN REGION 3.
      C3=SQRT(1.0/(RH3*(EPS3/(RHG3*CG3*CG3) +
     + (1.0-EPS3)/(RHL3*CL3*CL3))))
C  C4 IS THE MIXTURE SPEED OF SOUND IN REGION 4.
      C4=SQRT(1.0/(RH4*(EPS4/(RHG4*CG4*CG4) +
     + (1.0-EPS4)/(RHL4*CL4*CL4))))
C  RMD13 IS THE MASS FLUX FROM REGION 1 TO REGION 3.
      DLP=P1-P3
      IF(DLP.LT.0.0.AND.MBACK.EQ.1)RMD13=-DC*AVENT*SQRT(-2.0*RH3*G0*DLP)
      IF(DLP.LT.0.0.AND.MBACK.EQ.0)RMD13=0.0
      IF(DLP.GE.0.0)RMD13=DC*AVENT*SQRT(2.0*RH1*G0*DLP)
      IF(MOD1.EQ.0)RMD13=0.0
C  LIQUID INJECTION VELOCITY INTO REGION 3.
      U13=RMD13/(RH1*AVENT)
C  WEBER NUMBER
      WE=RHG3*U13*U13*DOR/(2.0*SIGMA)
C  RMD34 IS THE MASS FLUX FROM REGION 3 TO REGION 4.
      DLP=P3-PL
      IF(DLP.LT.0.0)RMD34=-DC34*A4*SQRT(-2.0*RH4*G0*DLP)
      IF(DLP.GE.0.0)RMD34=DC34*A4*SQRT(2.0*RH3*G0*DLP)
      IF(MOD4.EQ.0)RMD34=0.0
C  EL1 IS THE TOTAL ENERGY IN REGION 1.
      EL1=ENER*RM1
C  EL3 IS THE LIQUID ENERGY IN REGION 3.
      EL3=ENER*RML3
C  EG3 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 3.
      EG3=ENER3*RMG3
C  EL4 IS THE LIQUID ENERGY IN REGION 4.
      EL4=ENER*RML4
C  EG4 IS THE TOTAL INTERNAL GAS ENERGY IN REGION 4.
      EG4=ENER4*RMG4
C  EKPS IS THE KINETIC ENERGY OF THE PISTON.
      EKPS=0.5*PSWT*VPS*VPS/G0
C  EKPJ IS THE KINETIC ENERGY OF THE PROJECTILE.
      EKPJ=0.5*PJWT*VPJ*VPJ/G0
C  EKG4 IS THE KINETIC ENERGY OF THE GAS IN REGION 4.
C  ASSUME THAT THE DENSITY IS CONSTANT.
C  ASSUME THAT THE VELOCITY IS A LINEAR FUNCTION.
C  SO USE THE AVERAGE VELOCITY TO COMPUTE THE KINETIC ENERGY.
      EKG4=0.0
```

```
      IF(MOD4.EQ.0)GO TO 32                                          016440
      UL=0.0                                                         016450
      UR=VPJ                                                         016460
      U=0.5*(UL+UR)                                                  016470
      EKG4=0.5*RMG4*UR*UR/(3.0*G0)                                   016480
C  EKL4 IS THE KINETIC ENERGY OF THE LIQUID IN REGION 4.             016490
C  IGNORE DRAG.                                                      016500
      EKL4=0.5*RML4*UR*UR/(3.0*G0)                                   016510
32    CONTINUE                                                       016520
C  ESUM IS THE SUM OF THE ENERGIES.                                  016530
      ESUM=EL1+EL3+EG3+EL4+EG4+EKPS+EKPJ+EKG4+EKL4                   016540
C  TEMP3 IS THE GAS TEMPERATURE IN REGION 3.                         016550
      TEMP3=P3*(1.0-COV*RMG3)/(RMG3*(GAM-1.0)*CV)                    016560
C  TEMP4 IS THE AVERAGE GAS TEMPERATURE IN REGION 4.                 016570
      TEMP4=P4*(1.0-COV*RMG4)/(RMG4*(GAM-1.0)*CV)                    016580
C  HL1 IS THE ENTHALPY OF THE LIQUID IN REGION 1.                    016590
      HL1=ENER + P1/RH1                                              016600
C  HL3 IS THE ENTHALPY OF THE LIQUID IN REGION 3.                    016610
      HL3=ENER + P3/RHL3                                             016620
C  HL4 IS THE ENTHALPY OF THE LIQUID IN REGION 4.                    016630
      HL4=ENER + P4/RHL4                                             016640
C  H3 IS THE ENTHALPY OF THE GAS IN REGION 3.                        016650
      H3=CP*TEMP3 + P3*COV                                           016660
C  H4 IS THE ENTHALPY OF THE GAS IN REGION 4.                        016670
      H4=CP*TEMP4 + P4*COV                                           016680
C  ***********************************************                  016690
C  DROPLET FORMATION.                                                016700
C  ASSUME THAT THE GAS CANNOT FLOW FROM REGION 4 TO REGION 3.        016710
      IF(RMD34.LT.0.0)RMD34=0.0                                      016720
C  RMD3 IS THE RATE OF CHANGE OF LIQUID TO GAS IN REGION 3.          016730
      RMD3=0.0                                                       016740
      DO 105 I=1,10                                                  016750
      PMD3(I)=PML3(I)*(6.0/PDDR(I))*ADR*(P3**BDR)                    016760
      RMD3=RMD3+PMD3(I)                                              016770
  105 CONTINUE                                                       016780
C  DD3 IS THE RATE OF CHANGE OF THE DIAMETER IN REGION 3.            016790
      DD3=2.0*ADR*(P3**BDR)                                          016800
C  RMD4 IS THE RATE OF CHANGE OF LIQUID TO GAS IN REGION 4.          016810
      RMD4=0.0                                                       016820
      DO 110 I=1,10                                                  016830
      PMD4(I)=PML4(I)*(6.0/PDDR(I))*ADR*(P4**BDR)                    016840
      RMD4=RMD4+PMD4(I)                                              016850
  110 CONTINUE                                                       016860
C  DD4 IS THE RATE OF CHANGE OF THE DIAMETER IN REGION 4.            016870
      DD4=2.0*ADR*(P4**BDR)                                          016880
      IF(MOD1.EQ.0)GO TO 150                                         016890
      IF(MOD4.EQ.0)GO TO 140                                         016900
C  REGION 1 IS OPEN.                                                 016910
C  REGION 4 IS OPEN.                                                 016920
      YDOT(1)=-VPS*A1                                                016930
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR             016940
      FOPC=P3*(A3-AVENT) - P1*(A1-AVENT)                             016950
      FOPP=PSRS*(A1-AVENT)                                           016960
      IF(TVENT(1).EQ.10HVENT3      )FOPC=P3*(A3-AHOLE)-P1*(A1-AHOLE) 016970
      IF(TVENT(1).EQ.10HVENT3      )FOPP=PSRS*(A1-AHOLE)             016980
      IF(FOPC.GE.FOPP)YDOT(2)=(G0/PSWT)*(FOPC-FOPP)                  016990
      IF(FOPC.LT.FOPP)YDOT(2)=0.0                                    017000
```

```
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC
      YDOT(3)=VPS
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1
      YDOT(5)=CL1*CL1*YDOT(4)/G0
      YDOT(6)=VPS*A3
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR
      YDOT(7)=-RH3*YDOT(6)/V3 + RMD13/V3 -RMD34/V3
      YDOT(8)=(RH3*C3*C3/G0) * (-YDOT(6)/V3  +
     + RMD13/(RHL3*V3) + (RMD3/(RHG3*V3)) *
     + (1.0 - RHG3/RHL3 + G0*(HL3-H3)*(GAM-1.0)/
     + (CG3*CG3*(1.0-COV*RHG3))) - RMD34/RM3)
      YDOT(9)=VPJ*A4
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4
      YDOT(13)=(RH4*C4*C4/(G0*V4)) * (-YDOT(9) +
     + (RMD4/RHG4) * (1.0-RHG4/RHL4 +
     + G0*(HL4-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) +
     + (RMD34/(RHG4*RM3)) * (RHL3*RHG4/RHL4 + RMG3 +
     + G0*RMG3*(H3-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) )
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=RMD13 - RHL3*RMD34/RM3 - RMD3
      YDOT(16)=G0*YDOT(13)/(CL4*CL4)
      YDOT(17)=RHL3*RMD34/RM3 - RMD4
      YDOT(18)=RMD13-PML3(1)*RMD34/RM3-RMD3(1)-
     + PML3(1)*DD3/DINC
      DO 135 I=2,9
      IM=I-1
      IDOT=I+17
      YDOT(IDOT)=-PML3(I)*RMD34/RM3 - RMD3(I) +
     + (PML3(IM)-PML3(I))*DD3/DINC
  135 CONTINUE
      YDOT(27)=-PML3(10)*RMD34/RM3 - RMD3(10) +
     + PML3(9)*DD3/DINC
      YDOT(28)=PML3(1)*RMD34/RM3-RMD4(1)-
     + PML4(1)*DD4/DINC
      DO 145 I=2,9
      IM=I-1
      IDOT=I+27
      YDOT(IDOT)=PML3(I)*RMD34/RM3 - RMD4(I) +
     + (PML4(IM)-PML4(I))*DD4/DINC
  145 CONTINUE
      YDOT(37)=PML3(10)*RMD34/RM3 - RMD4(10) +
     + PML4(9)*DD4/DINC
      RETURN
  140 CONTINUE
C  REGION 1 IS OPEN.
C  REGION 4 IS CLOSED.
C  VIEW REGION 3 AND REGION 4 AS ONE REGION.
      YDOT(1)=-VPS*A1
      IF(TVENT(1).EQ.10HVENT3      )YDOT(1)=YDOT(1)+VCOR
      FORC=P3*(A3-AVENT) - P1*(A1-AVENT)
      FORP=PSPS*(A1-AVENT)
```

```
      IF(TVENT(1).EQ.10HVENT3      )FORC=P3*(A3-AHOLE)-P1*(A1-AHOLE)
      IF(TVENT(1).EQ.10HVENT3      )FORR=PSRS*(A1-AHOLE)
      IF(FORC.GE.FORR)YDOT(2)=(G0/PSWT)*(FORC-FORR)
      IF(FORC.LT.FORR)YDOT(2)=0.0
      IF(FORC.LT.0.0)YDOT(2)=(G0/PSWT)*FORC
      YDOT(3)=VPS
      IF(SPS.LT.0.0.AND.YDOT(2).LT.0.0)YDOT(2)=0.0
      IF(SPS.LT.0.0.AND.YDOT(3).LT.0.0)YDOT(3)=0.0
      YDOT(4)=-RH1*YDOT(1)/V1 - RMD13/V1
      YDOT(5)=CL1*CL1*YDOT(4)/G0
      YDOT(6)=VPS*A3
      IF(TVENT(1).EQ.10HVENT3      )YDOT(6)=YDOT(6)-VCOR
      YDOT(9)=VPJ*A4
      V34=V3+V4
      RM34=RM3+RM4
      YDOT(7)=-RM3*(YDOT(6)+YDOT(9))/V34 + RMD13/V34
      YDOT(8)=(RM3*C3*C3/G0)*(-(YDOT(6)+YDOT(9))/V34 +
     + (RMD13/(RHL3*V34)) + ((RMD3+RMD4)/(RHG3*V34))*
     + (1.0 - RHG3/RHL3 + G0*(HL3-H3)*(GAM-1.0)/
     + (CG3*CG3*(1.0-COV*RHG3))) )
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=YDOT(7)
      YDOT(13)=YDOT(8)
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=(RMD13-RMD3-RMD4)*V3/V34
      YDOT(16)=YDOT(14)
      YDOT(17)=YDOT(15)*V4/V3
      YDOT(18)=(RMD13 -RMD3(1) - RMD4(1) -
     + (RML3(1)+RML4(1))*DD3/DINC)*V3/V34
      DO 155 I=2,9
      IM=I-1
      IDOT=I+17
      YDOT(IDOT)= (-RMD3(I) - RMD4(I) +
     + (RML3(IM)+RML4(IM)-RML3(I)-RML4(I))*DD3/DINC)*V3/V34
  155 CONTINUE
      YDOT(27)= (-RMD3(10) - RMD4(10) +
     + (RML3(9)+RML4(9))*DD4/DINC)*V3/V34
      DO 158 I=18,27
      IP=I+10
  158 YDOT(IP)=YDOT(I)*V4/V3
      RETURN
  150 CONTINUE
      IF(MOD4.EQ.1)GO TO 160
C  REGION 1 IS CLOSED.
C  REGION 4 IS CLOSED.
      YDOT(1)=0.0
      YDOT(2)=0.0
      YDOT(3)=0.0
      YDOT(4)=0.0
      YDOT(5)=0.0
      YDOT(6)=0.0
      V34=V3+V4
      YDOT(9)=VPJ*A4
      YDOT(7)=-RM3*(YDOT(6)+YDOT(9))/V34
```

```
      YDOT(8)=(RH3*C3*C3/G0)*(-(YDOT(6)+YDOT(9))/V3 +
     + ((RMD3+RMD4)/(RHG3*V34))*(1.0 - RHG3/RHL3 +
     + G0*(HL3-H3)*(GAM-1.0)/(CG3*CG3*(1.0-COV*RHG3))) )
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=YDOT(7)
      YDOT(13)=YDOT(8)
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=(-RMD3-RMD4)*V3/V34
      YDOT(16)=YDOT(14)
      YDOT(17)=YDOT(15)*V4/V3
      YDOT(18)=(-RMD3(1) - RMD4(1) -
     + (RML3(1)+RML4(1))*DD3/DINC)*V3/V34
      DO 165 I=2,9
      IM=I-1
      IDOT=I+17
      YDOT(IDOT)= (-RMD3(I) - RMD4(I) +
     + (RML3(IM)+RML4(IM)-RML3(I)-RML4(I))*DD3/DINC)*V3/V34
  165 CONTINUE
      YDOT(27)= (-RMD3(10) - RMD4(10) +
     + (RML3(9)+RML4(9))*DD3/DINC)*V3/V34
      DO 168 I=18,27
      IP=I+10
  168 YDOT(IP)=YDOT(I)*V4/V3
      RETURN
  160 CONTINUE
C  REGION 1 IS CLOSED.
C  REGION 4 IS OPEN.
      YDOT(1)=0.0
      YDOT(2)=0.0
      YDOT(3)=0.0
      YDOT(4)=0.0
      YDOT(5)=0.0
      YDOT(6)=0.0
      YDOT(7)=-RH34*YDOT(6)/V3 -RMD34/V3
      YDOT(8)=(RH3*C3*C3/G0)*(-YDOT(6)/V3 +
     + (RMD3/(RHG3*V3))*(1.0 - RHG/RHL3 +
     + G0*(HL3-H3)*(GAM-1.0)/(CG3*CG3*(1.0-COV*RHG3))) -
     + RMD34/RH3)
      YDOT(9)=VPJ*A4
      IF(PR.GE.PJRS)YDOT(10)=(PR-PJRS)*A4*G0/PJWT
      IF(PR.LT.PJRS)YDOT(10)=0.0
      APJ=YDOT(10)/(.098146*1.0E7)
      YDOT(11)=VPJ
      YDOT(12)=-RH4*YDOT(9)/V4 + RMD34/V4
      YDOT(13)=(RH4*C4*C4/(G0*V4)) * (-YDOT(9) +
     + (RMD4/RHG4) * (1.0-RHG4/RHL4 +
     + G0*(HL4-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) +
     + (RMD34/(RHG4*RH3)) * (RML3*RHG4/RHL4 + RHG3 +
     + G0*RHG3*(H3-H4)*(GAM-1.0)/(CG4*CG4*(1.0-COV*RHG4))) )
      YDOT(14)=G0*YDOT(8)/(CL3*CL3)
      YDOT(15)=-RML3*RMD34/RH3 - RMD3
      YDOT(16)=G0*YDOT(13)/(CL4*CL4)
      YDOT(17)=RML3*RMD34/RH3 - RMD4
      YDOT(18)=-RML3(1)*RMD34/RH3-RMD3(1)-
```

```
     + PML3(1)*DD3/DINC                                                 018720
      DO 175 I=2,9                                                      018730
      IM=I-1                                                            018740
      IDOT=I+17                                                         018750
      YDOT(IDOT)=-PML3(I)*RMD34/RM3 - PMD3(I) +                         018760
     + (PML3(IM)-PML3(I))*DD3/DINC                                      018770
  175 CONTINUE                                                          018780
      YDOT(27)=-PML3(10)*RMD34/RM3 - PMD3(10) +                         018790
     + PML3(9)*DD3/DINC                                                 018800
      YDOT(28)=PML3(1)*RMD34/RM3-PMD4(1)-                               018810
     + PML4(1)*DD4/DINC                                                 018820
      DO 178 I=2,9                                                      018830
      IM=I-1                                                            018840
      IDOT=I+27                                                         018850
      YDOT(IDOT)=PML3(I)*RMD34/RM3 - PMD4(I) +                          018860
     + (PML4(IM)-PML4(I))*DD4/DINC                                      018870
  178 CONTINUE                                                          018880
      YDOT(37)=PML3(10)*RMD34/RM3 - PMD4(10) +                          018890
     + PML4(9)*DD4/DINC                                                 018900
      RETURN                                                            018910
      END                                                               018920
```

```
1          SUBROUTINE PEDERV (N, T, Y, PD, N0)                    018930
           RETURN                                                 018940
           END                                                    018950
```

```
      SUBROUTINE DRIVE(N,T0,H0,Y0,TOUT,EPS,IERROR,MF,INDEX)
C THIS IS THE JUNE 24, 1975  VERSION OF
C EPISODE..  EXPERIMENTAL PACKAGE FOR INTEGRATION OF
C SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,
C    DY/DT = F(Y,T),  Y = (Y(1),Y(2),....Y(N)) TRANSPOSE,
C GIVEN THE INITIAL VALUE OF Y.
C THIS CODE IS FOR THE IBM 370/195 AT ARGONNE NATIONAL LABORATORY
C AND IS A MODIFICATION OF EARLIER VERSIONS BY G.D.BYRNE
C AND A.C.HINDMARSH.
C                    REFERENCES
C 1.  G. D. BYRNE AND A. C. HINDMARSH, A POLYALGORITHM FOR THE
C       NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS,
C       UCRL-75652, LAWRENCE LIVERMORE LABORATORY, P. O. BOX 808,
C       LIVERMORE, CA 94550, APRIL 1974. ALSO IN ACM TRANSACTIONS
C       ON MATHEMATICAL SOFTWARE, 1 (1975), PP. 71-96.
C
C 2.  A. C. HINDMARSH AND G. D. BYRNE, EPISODE.. AN EXPERIMENTAL
C       PACKAGE FOR THE INTEGRATION OF SYSTEMS OF ORDINARY
C       DIFFERENTIAL EQUATIONS, UCID-30112, L.L.L., MAY, 1975.
C
C 3.  A. C. HINDMARSH, GEAR.. ORDINARY DIFFERENTIAL EQUATION
C       SYSTEM SOLVER, UCID-30001, REV. 3, L.L.L., DECEMBER, 1974.
C
C-----------------------------------------------------------------------
C DRIVE IS A DRIVER SUBROUTINE FOR THE EPISODE PACKAGE.
C DRIVE IS TO BE CALLED ONCE FOR EACH OUTPUT VALUE OF T.
C IT THEN MAKES REPEATED CALLS TO THE CORE INTEGRATOR
C SUBROUTINE, TSTEP.
C
C THE INPUT PARAMETERS ARE AS FOLLOWS.
C   N     =  THE NUMBER OF DIFFERENTIAL EQUATIONS (USED ONLY ON
C              FIRST CALL, UNLESS INDEX = -1).  N MUST NEVER BE
C              INCREASED DURING A GIVEN PROBLEM.
C   T0    =  THE INITIAL VALUE OF T, THE INDEPENDENT VARIABLE
C              (USED FOR INPUT ONLY ON FIRST CALL).
C   H0    =  THE STEP SIZE IN T (USED FOR INPUT ONLY ON THE
C              FIRST CALL, UNLESS INDEX = 3 ON INPUT).  WHEN
C              INDEX = 3, H0 IS THE MAXIMUM ABSOLUTE VALUE OF
C              THE STEP SIZE TO BE USED.
C   Y0    =  A VECTOR OF LENGTH N CONTAINING THE INITIAL VALUES OF
C              Y (USED FOR INPUT ONLY ON FIRST CALL).
C   TOUT  =  THE VALUE OF T AT WHICH OUTPUT IS DESIRED NEXT.
C              INTEGRATION WILL NORMALLY GO BEYOND TOUT AND
C              INTERPOLATE TO T = TOUT.  (USED ONLY FOR INPUT.)
C   EPS   =  THE RELATIVE ERROR BOUND (USED ONLY ON FIRST CALL,
C              UNLESS INDEX = -1).  THIS BOUND IS USED AS FOLLOWS.
C              LET R(I) DENOTE THE ESTIMATED RELATIVE LOCAL ERROR
C              IN Y(I), I.E. THE ERROR RELATIVE TO YMAX(I), AS
C              MEASURED PER STEP (OF SIZE H) OR PER SS UNITS OF T.
C              THEN EPS IS A BOUND ON THE ROOT-MEAN-SQUARE NORM
C              OF THE VECTOR R, I.E.
C                          N
C              SQRT ( SUM ( R(I)**2 )/N ) .LT. EPS.
C                         I=1
C              THE VECTOR YMAX IS COMPUTED IN DRIVE AS DESCRIBED
C              UNDER IERROR BELOW.
C              IF ERROR CONTROL PER SS UNITS OF T IS DESIRED, SET SS
```

```
C              TO A POSITIVE NUMBER AFTER STATEMENT 10 (WHERE IT IS      019530
C              NOW SET TO ZERO) AND UPDATE IT AFTER STATEMENT 60.        019540
C              SEE ALSO THE COMMENTS ON SS AND YMAX BELOW.               019550
C  IERROR =  THE ERROR FLAG WITH VALUES AND MEANINGS AS FOLLOW.         019560
C         1    ABSOLUTE ERROR IS CONTROLLED.  YMAX(I) = 1.0.            019570
C         2    ERROR RELATIVE TO ABS(Y) IS CONTROLLED.  IF Y(I) = 0.0   019580
C              A DIVIDE ERROR WILL OCCUR.  YMAX(I) = ABS(Y(I)).         019590
C         3    ERROR RELATIVE TO THE LARGEST VALUE OF ABS(Y(I)) SEEN    019600
C              SO FAR IS CONTROLLED.  IF THE INITIAL VALUE OF Y(I) IS   019610
C              0.0, THEN YMAX(I) IS SET TO 1.0 INITIALLY AND REMAINS    019620
C              AT LEAST 1.0.                                            019630
C   MF     =  THE METHOD FLAG (USED ONLY ON FIRST CALL, UNLESS         019640
C              INDEX = -1).  ALLOWED VALUES ARE 10, 11, 12, 13,         019650
C              20, 21, 22, 23.  MF IS AN INTEGER WITH TWO DECIMAL       019660
C              DIGITS, METH AND MITER (MF = 10*METH + MITER).  (MF      019670
C              CAN BE THOUGHT OF AS THE ORDERED PAIR (METH,MITER).)     019680
C              METH IS THE BASIC METHOD INDICATOR.                      019690
C                METH = 1 INDICATES VARIABLE-STEP SIZE, VARIABLE-       019700
C                         ORDER ADAMS METHOD, SUITABLE FOR NON-         019710
C                         STIFF PROBLEMS.                               019720
C                METH = 2 INDICATES VARIABLE-STEP SIZE, VARIABLE-       019730
C                         ORDER BACKWARD DIFFERENTIATION METHOD,        019740
C                         SUITABLE FOR STIFF PROBLEMS.                  019750
C              MITER INDICATES THE METHOD OF ITERATIVE CORRECTION       019760
C                (NONLINEAR SYSTEM SOLUTION).                           019770
C                MITER = 0 INDICATES FUNCTIONAL ITERATION (NO           019780
C                          PARTIAL DERIVATIVES NEEDED).                 019790
C                MITER = 1 INDICATES A CHORD OR SEMI-STATIONARY         019800
C                          NEWTON METHOD WITH CLOSED FORM (EXACT)       019810
C                          JACOBIAN, WHICH IS COMPUTED IN THE           019820
C                          USER SUPPLIED SUBROUTINE                     019830
C                          PEDERV(N,T,Y,PD,NO) DESCRIBED BELOW.         019840
C                MITER = 2 INDICATES A CHORD  OR SEMI-STATIONARY        019850
C                          NEWTON METHOD WITH AN INTERNALLY             019860
C                          COMPUTED FINITE DIFFERENCE APPROXIMATION     019870
C                          TO THE JACOBIAN.                             019880
C                MITER = 3 INDICATES A CHORD OR SEMI-STATIONARY         019890
C                          NEWTON METHOD WITH AN INTERNALLY             019900
C                          COMPUTED DIAGONAL MATRIX APPROXIMATION       019910
C                          TO THE JACOBIAN, BASED ON A DIRECTIONAL      019920
C                          DERIVATIVE.                                  019930
C   INDEX =  INTEGER USED ON INPUT TO INDICATE TYPE OF CALL.            019940
C              WITH THE FOLLOWING VALUES AND MEANINGS..                 019950
C         1    THIS IS THE FIRST CALL FOR THIS PROBLEM.                 019960
C         0    THIS IS NOT THE FIRST CALL FOR THIS PROBLEM,             019970
C              AND INTEGRATION IS TO CONTINUE.                          019980
C        -1    THIS IS NOT THE FIRST CALL FOR THE PROBLEM,              019990
C              AND THE USER HAS RESET N, EPS, AND/OR MF.                020000
C         2    SAME AS 0 EXCEPT THAT TOUT IS TO BE HIT                  020010
C              EXACTLY (NO INTERPOLATION IS DONE).                      020020
C              ASSUMES TOUT .GE. THE CURRENT T.                         020030
C         3    SAME AS 0 EXCEPT CONTROL RETURNS TO CALLING              020040
C              PROGRAM AFTER ONE STEP.  TOUT IS IGNORED.                020050
C            SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,               020060
C            IT NEED NOT BE RESET FOR NORMAL CONTINUATION.              020070
C                                                                       020080
C AFTER THE INITIAL CALL, IF A NORMAL RETURN OCCURRED AND A NORMAL      020090
```

```
C CONTINUATION IS DESIRED, SIMPLY RESET TOUT AND CALL AGAIN.
C ALL OTHER PARAMETERS WILL BE READY FOR THE NEXT CALL.
C A CHANGE OF PARAMETERS WITH INDEX = -1 CAN BE MADE AFTER
C EITHER A SUCCESSFUL OR AN UNSUCCESSFUL RETURN.
C
C THE OUTPUT PARAMETERS ARE AS FOLLOWS.
C   T0    =  THE OUTPUT VALUE OF T.  IF INTEGRATION WAS SUCCESSFUL,
C              T0 = TOUT.  OTHERWISE, T0 IS THE LAST VALUE OF T
C              REACHED SUCCESSFULLY.
C   H0    =  THE STEP SIZE H USED LAST, WHETHER SUCCESSFULLY OR NOT.
C   Y0    =  THE COMPUTED VALUES OF Y AT T = T0.
C   INDEX =  INTEGER USED ON OUTPUT TO INDICATE RESULTS.
C                WITH THE FOLLOWING VALUES AND MEANINGS..
C          0     INTEGRATION WAS COMPLETED TO TOUT OR BEYOND.
C         -1     THE INTEGRATION WAS HALTED AFTER FAILING TO PASS THE
C                ERROR TEST EVEN AFTER REDUCING H BY A FACTOR OF
C                1.E10 FROM ITS INITIAL VALUE.
C         -2     AFTER SOME INITIAL SUCCESS, THE INTEGRATION WAS
C                HALTED EITHER BY REPEATED ERROR TEST FAILURES OR
C                BY A TEST ON EPS.  POSSIBLY TOO MUCH ACCURACY HAS
C                BEEN REQUESTED, OR A BAD CHOICE OF MF WAS MADE.
C         -3     THE INTEGRATION WAS HALTED AFTER FAILING TO ACHIEVE
C                CORRECTOR CONVERGENCE EVEN AFTER REDUCING H BY A
C                FACTOR OF 1.E10 FROM ITS INITIAL VALUE.
C         -4     IMMEDIATE HALT BECAUSE OF ILLEGAL VALUES OF INPUT
C                PARAMETERS.  SEE PRINTED MESSAGE.
C         -5     INDEX WAS -1 ON INPUT, BUT THE DESIRED CHANGES OF
C                PARAMETERS WERE NOT IMPLEMENTED BECAUSE TOUT
C                WAS NOT BEYOND T.  INTERPOLATION TO T = TOUT WAS
C                PERFORMED AS ON A NORMAL RETURN.  TO CONTINUE,
C                SIMPLY CALL AGAIN WITH INDEX = -1 AND A NEW TOUT.
C         -6     INDEX WAS 2 ON INPUT, BUT TOUT WAS NOT BEYOND T.
C                NO ACTION WAS TAKEN.
C
C IN ADDITION TO DRIVE, THE FOLLOWING SUBROUTINES ARE USED BY AND
C PROVIDED IN THIS PACKAGE
C   INTERP(TOUT,Y,N0,Y0)  INTERPOLATES TO GIVE OUTPUT VALUES AT
C                          T = TOUT  BY USING DATA IN THE Y ARRAY.
C   TSTEP(Y,N0)  IS THE CORE INTEGRATION SUBROUTINE, WHICH INTEGRATES
C                OVER A SINGLE STEP AND DOES ASSOCIATED ERROR
C                CONTROL.
C   COSET  SETS COEFFICIENTS FOR USE IN TSTEP.
C   ADJUST(Y,N0)  ADJUSTS THE HISTORY ARRAY Y ON REDUCTION OF ORDER.
C   PSET(Y,N0,CON,MITER,IER)  COMPUTES AND PROCESSES THE JACOBIAN
C                             MATRIX, J = DF/DY.
C   DEC(N,N0,A,IP,IER)  PERFORMS THE LU DECOMPOSITION OF A MATRIX.
C   SOL(N,N0,A,B,IP)  SOLVES A LINEAR SYSTEM A*X = B, AFTER DEC
C                     HAS BEEN CALLED FOR THE MATRIX A.
C NOTE   PSET, DEC, AND SOL ARE CALLED IF AND ONLY IF MITER = 1
C        OR MITER = 2.
C
C THE USER MUST FURNISH THE FOLLOWING SUBROUTINES
C   DIFFUN(N,T,Y,YDOT)  COMPUTES THE FUNCTION  YDOT = F(Y,T),
C                       THE RIGHT HAND SIDE OF THE ORDINARY
C                       DIFFERENTIAL EQUATION SYSTEM, WHERE Y
C                       AND YDOT ARE VECTORS OF LENGTH N.
C   PEDERV(N,T,Y,PD,N0)  COMPUTES THE N BY N JACOBIAN MATRIX OF
```

```
C                          PARTIAL DERIVATIVES AND STORES IT IN PD AS
C                          AN NO BY NO ARRAY.   PD(I,J) IS TO BE SET
C                          TO THE PARTIAL DERIVATIVE OF YDOT(I) WITH
C                          RESPECT TO Y(J).  PEDERV IS CALLED IF AND
C                          ONLY IF MITER = 1. FOR OTHER VALUES OF
C                          MITER, PEDERV CAN BE A DUMMY SUBROUTINE.
C
C CAUTION    AT THE PRESENT TIME THE MAXIMUM NUMBER OF DIFFERENTIAL
C            EQUATIONS, WHICH CAN BE SOLVED BY EPISODE, IS 20.  TO
C            CHANGE THIS NUMBER TO A NEW VALUE, SAY NMAX, CHANGE
C            Y(20,13) TO Y(NMAX,13), YMAX(20) TO YMAX(NMAX),
C            ERROR(20) TO ERROR(NMAX), SAVE1(20) TO SAVE1(NMAX),
C            SAVE2(20) TO SAVE2(NMAX), PW(400) TO PW(NMAX*NMAX),
C            AND IPIV(20) TO IPIV(NMAX) IN THE COMMON AND DIMENSION
C            STATEMENTS BELOW.  ALSO CHANGE THE ARGUMENT IN THE
C            IF...GO TO 440 STATEMENT (AFTER THE COMMON STATEMENTS)
C            FROM 20 TO NMAX.  NO OTHER CHANGES NEED TO BE MADE TO
C            ANY OTHER SUBROUTINE IN THIS PACKAGE WHEN THE MAXIMUM
C            NUMBER OF EQUATIONS IS CHANGED.  ELSEWHERE, THE COLUMN
C            LENGTH OF THE Y ARRAY IS NO INSTEAD OF 20.  THE ROW
C            LENGTH OF Y CAN BE REDUCED FROM 13 TO 6 IF METH = 2.
C            THE ARRAY IPIV IS USED IF AND ONLY IF MITER = 1 OR
C           MITER = 2.  THE SIZE OF THE PW ARRAY CAN BE REDUCED
C           TO 1 IF MITER = 0 OR TO N IF MITER = 3.
C
C THE COMMON BLOCK EPCOM9 CAN BE ACCESSED EXTERNALLY BY THE USER,
C IF HE DESIRES.  IT CONTAINS THE STEP SIZE LAST USED SUCCESSFULLY
C (HUSED), THE ORDER LAST USED SUCCESSFULLY (NQUSED), THE
C NUMBER OF STEPS TAKEN SO FAR (NSTEP), THE NUMBER OF FUNCTION
C EVALUATIONS (DIFFUN CALLS) SO FAR (NFE), AND THE NUMBER OF
C JACOBIAN EVALUATIONS SO FAR (NJE).
C
C IN A DATA STATEMENT BELOW, LOUT IS SET TO THE LOGICAL UNIT NUMBER
C FOR THE OUTPUT OF MESSAGES DURING INTEGRATION.  CURRENTLY, LOUT
C = 3.
C-----------------------------------------------------------------------
C
      DIMENSION Y(50,13),Y0(50)
      DIMENSION YMAX(50),ERROR(50),SAVE1(50),SAVE2(50),SAVE3(50)
      DIMENSION PW(2500),IPIV(50)
C
      COMMON /EPCOM1/T,H,HMIN,HMAX,EPSC,SS,UROUND,NC,MFC,KFLAG,JSTART
      COMMON/EPCOM8/EPSJ,NSQR
      COMMON/EPCOM9/HUSED,NQUSED,NSTEP,NFE,NJE,NMI
      COMMON/TABINT/KWRITE,KOUTD,SREC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)
C
      DATA LOUT /6/
C
      HCUT=.1
      FOUR=4.0
      HUNDRD=100.0
      ONE=1.0
      TEN=10.0
      ZERO=0.0
      IF (INDEX .EQ. 0) GO TO 20
      IF (INDEX .EQ. 2) GO TO 25
      IF (INDEX .EQ. -1) GO TO 30
```

```
      IF (INDEX .EQ. 3) GO TO 40
      IF (INDEX .NE. 1) GO TO 430
      IF (EPS .LE. ZERO) GO TO 400
      IF (N .LE. 0) GO TO 410
      IF ((T0-TOUT)*H0 .GE. ZERO) GO TO 420
C-----------------------------------------------------------------------
C IF INITIAL VALUES FOR YMAX OTHER THAN THOSE BELOW ARE DESIRED,
C THEY SHOULD BE SET HERE.  ALL YMAX(I) MUST BE POSITIVE.  IF
C VALUES FOR HMIN OR HMAX, THE BOUNDS ON THE ABSOLUTE VALUE OF H,
C OTHER THAN THOSE BELOW, ARE DESIRED, THEY ALSO SHOULD BE SET HERE.
C IF ERROR PER SS UNITS OF T IS TO BE CONTROLLED, SS SHOULD BE SET
C TO A POSITIVE VALUE BELOW.  ERROR PER UNIT STEP IS CONTROLLED
C WHEN SS = 1.  THE DEFAULT VALUE FOR SS IS 0 AND YIELDS CONTROL
C OF ERROR PER STEP.
C-----------------------------------------------------------------------
C SET UROUND, THE MACHINE ROUNDOFF CONSTANT, HERE.
C USE STATEMENT BELOW FOR SHORT PRECISION ON IBM 360 OR 370.
C     UROUND = 9.53674E-7
C USE STATEMENT BELOW FOR SINGLE PRECISION ON CDC 7600 OR 6600.
C     UROUND = 7.105427406E-15
C USE STATEMENT BELOW FOR LONG PRECISION ON IBM 360 OR 370.
C-----------------------------------------------------------------------
      UROUND=7.105427406E-15
      DO 10 I = 1,N
        GO TO (5, 6, 7), IERROR
C IERROR   =   1, 2, 3  ---------------------------------------------
    5   YMAX(I) = ONE
        GO TO 10
C
    6 YMAX(I)=AMAX1(Y0(I),SPEC)
      GO TO 10
    7 YMAX(I)=ABS(Y0(I))
        IF (YMAX(I) .EQ. ZERO) YMAX(I) = ONE
   10   Y(I,1) = Y0(I)
      NC = N
      T = T0
      H = H0
      IF ((T+H) .EQ. T) WRITE(6,15) T
C
   15 FORMAT(/46H---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE,,
     1       24H THE O.D.E. SOLVER.  ---/22H WARNING.. T + H = T =,
     2E18.8,18H IN THE NEXT STEP./)
      HMIN=ABS(H0)
      HMAX=ABS(T0-TOUT)*TEN
      EPSC = EPS
      MFC = MF
      JSTART = 0
      SS = ZERO
      N0 = N
      NS0 = N0*N0
      EPSJ=SQRT(UROUND)
      NBCUT = 0
      GO TO 50
C TOP IS THE PREVIOUS OUTPUT VALUE OF T0 FOR USE IN HMAX. ------------
   20 HMAX=ABS(TOUT-TOP)*TEN
      GO TO 80
   25 HMAX=ABS(TOUT-TOP)*TEN
```

```
      IF ((T-TOUT)*H .GE. ZERO) GO TO 450
      GO TO 85
C
 30   IF ((T-TOUT)*H .GE. ZERO) GO TO 450
      IF (MF .NE. MFC) JSTART = -1
      NC = N
      EPSC = EPS
      MFC = MF
      GO TO 45
C
 40   HMAX = H0
C
 45   IF ((T+H) .EQ. T) WRITE(6,15) T
C
50    CALL TSTEP(N0,NS0,Y,YMAX,ERROR,SAVE1,SAVE2,SAVE3,PW,IPIV)
C
      KGO = 1 - KFLAG
      GO TO (60, 100, 200, 300), KGO
C KFLAG  =   0,  -1,  -2,  -3  ------------------------------------
C
 60   CONTINUE
C-----------------------------------------------------------------
C NORMAL RETURN FROM TSTEP.
C
C THE WEIGHTS YMAX(I) ARE UPDATED.  IF DIFFERENT VALUES ARE DESIRED,
C THEY SHOULD BE SET HERE.  IF SS IS TO BE UPDATED FOR CONTROL OF
C ERROR PER SS UNITS OF T, IT SHOULD ALSO BE DONE HERE.  A TEST IS
C MADE TO DETERMINE IF EPS IS TOO SMALL FOR MACHINE PRECISION.
C
C ANY OTHER TESTS OR CALCULATIONS THAT ARE REQUIRED AFTER EACH STEP
C SHOULD BE INSERTED HERE.
C
C IF INDEX = 3, Y0 IS SET TO THE CURRENT Y VALUES ON RETURN.
C IF INDEX = 2, H IS CONTROLLED TO HIT TOUT (WITHIN ROUNDOFF
C ERROR), AND THEN THE CURRENT Y VALUES ARE PUT IN Y0 ON
C RETURN.  FOR ANY OTHER VALUE OF INDEX, CONTROL RETURNS TO
C THE INTEGRATOR UNLESS TOUT HAS BEEN REACHED.  THEN
C INTERPOLATED VALUES OF Y ARE COMPUTED AND STORED IN Y0 ON
C RETURN.
C IF INTERPOLATION IS NOT DESIRED, THE CALL TO INTERP SHOULD
C BE DELETED AND CONTROL TRANSFERRED TO STATEMENT 500 INSTEAD
C OF 520.
C-----------------------------------------------------------------
      D = ZERO
      DO 70 I = 1,N
      AYI=ABS(Y(I,1))
        GO TO (70, 66, 67), IERROR
C IERROR  =     1,  2,  3  ---------------------------------------
66    YMAX(I)=AMAX1(AYI,SREC)
        GO TO 70
67    YMAX(I)=AMAX1(YMAX(I),AYI)
70    D=D+(AYI/YMAX(I))**2
      D=D*(UROUND/EPS)**2
      IF(D.GT.FLOAT(N))GO TO 250
      IF (INDEX .EQ. 3) GO TO 500
      IF (INDEX .EQ. 2) GO TO 85
```

```
 80   IF ((T-TOUT)*H .LT. ZERO) GO TO 45
      CALL INTERP (TOUT, Y, N0, Y0)
      T0 = TOUT
      GO TO 520
 85   IF (((T+H)-TOUT)*H .LE. ZERO) GO TO 45
      IF(ABS(T-TOUT).LE.HUNDRD*UROUND*HMAX)GO TO 500
      IF ((T-TOUT)*H .GE. ZERO) GO TO 500
      H = (TOUT - T)*(ONE - FOUR*UROUND)
      JSTART = -1
      GO TO 45
C-----------------------------------------------------------------------
C ON AN ERROR RETURN FROM TSTEP, AN IMMEDIATE RETURN OCCURS IF
C KFLAG = -2, AND RECOVERY ATTEMPTS ARE MADE OTHERWISE.
C TO RECOVER, H AND HMIN ARE REDUCED BY A FACTOR OF .1 UP TO 10
C TIMES BEFORE GIVING UP.
C-----------------------------------------------------------------------
 100  WRITE (6,101)
 101  FORMAT (/46H---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE,,
     1        24H THE O.D.E. SOLVER.  ---/)
      WRITE(6,4400)J1
4400  FORMAT(5X,'J1 =',I5)
      WRITE(6,105) T,HMIN
 105  FORMAT(//35H KFLAG = -1 FROM INTEGRATOR AT T = ,E18.8/
     140H  ERROR TEST FAILED WITH ABS(H) = HMIN =,E18.8/)
 110  IF (NHCUT .EQ. 10) GO TO 150
      NHCUT = NHCUT + 1
      HMIN = HCUT*HMIN
      H = HCUT*H
      WRITE (6,115) H
 115  FORMAT(24H  H HAS BEEN REDUCED TO ,E18.8,
     1  26H  AND STEP WILL BE RETRIED//)
      JSTART = -1
      GO TO 45
C
 150  WRITE (6,155)
 155  FORMAT(//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)
      GO TO 500
C
 200  WRITE (6,101)
      WRITE (6,205) T,H,EPS
 205  FORMAT(//16H KFLAG = -2  T =,E18.8,4H H =,E18.8,6H EPS =,E19.8/
     1       50H  THE REQUESTED ERROR IS TOO SMALL FOR INTEGRATOR.//)
      GO TO 500
C
 250  WRITE (6,101)
      WRITE (6,255) T,EPS
C
  255 FORMAT(//46H INTEGRATION HALTED BY SUBROUTINE DRIVE AT T =,
     1 E18.8/43H EPS IS TOO SMALL FOR MACHINE PRECISION AND/
     2 29H PROBLEM BEING SOLVED.  EPS =,E18.8//)
      KFLAG = -2
      GO TO 500
C
 300  WRITE (6,101)
      WRITE (6,305) T
 305  FORMAT(//34H KFLAG = -3 FROM INTEGRATOR AT T =,E18.8/
     1  45H  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED/)
```

```
      GO TO 110
C
 400  WRITE (6,101)
      WRITE (6,405) EPS
 405  FORMAT(//35H ILLEGAL INPUT.. EPS .EE. 0. EPS = ,E18.8//)
      INDEX = -4
      RETURN
C
 410  WRITE (6,101)
      WRITE (6,415) N
 415  FORMAT(//31H ILLEGAL INPUT.. N .LE. 0. N = ,I8//)
      INDEX = -4
      RETURN
C
 420  WRITE (6,101)
      WRITE (6,425) T0,TOUT,H0
 425  FORMAT(//39H ILLEGAL INPUT.. (T0 - TOUT)*H0 .GE. 0./
     1  5H T0 =,E18.8,7H TOUT =,E18.8,5H H0 =,E18.8//)
      INDEX = -4
      RETURN
C
 430  WRITE (6,101)
      WRITE (6,435) INDEX
 435  FORMAT(//24H ILLEGAL INPUT.. INDEX =,I8//)
      INDEX = -4
      RETURN
C
 440  WRITE (6,101)
      WRITE (6,445) N
 445  FORMAT (//39H ILLEGAL INPUT.  THE NUMBER OF ORDINARY/
     1        43H DIFFERENTIAL EQUATIONS BEING SOLVED IS N =, I6/
     2        42H STORAGE ALLOCATION IN SUBROUTINE DRIVE IS/
     3        46H TOO SMALL.  SEE COMMENTS IN SUBROUTINE DRIVE./)
      INDEX = -4
      RETURN
C
 450  WRITE (6,101)
      WRITE (6,455) T,TOUT,H
C
 455  FORMAT(//46H INDEX = -1 ON INPUT WITH (T - TOUT)*H .GE. 0./
     1        44H INTERPOLATION WAS DONE AS ON NORMAL RETURN./
     2        41H DESIRED PARAMETER CHANGES WERE NOT MADE./
     3        4H T =,E18.8,7H TOUT =,E18.8,4H H =,E18.8//)
      CALL INTERP (TOUT, Y, N0, Y0)
      T0 = TOUT
      INDEX = -5
      RETURN
C
 460  WRITE (6,101)
      WRITE (6,465) T,TOUT,H
C
 465  FORMAT(//45H INDEX = 2 ON INPUT WITH (T - TOUT)*H .GE. 0./
     1   4H T =,E18.8,7H TOUT =,E18.8,4H H =,E18.8//)
      INDEX = -6
      RETURN
C
 500  T0 = T
```

```
      DO 510 I = 1,N
  510   Y0(I) = Y(I,1)
  520 INDEX = KFLAG
      TOP = T0
      H0 = HUSED
      IF (KFLAG .NE. 0) H0 = H
      RETURN
C----------------------- END OF SUBROUTINE DRIVE -------------------
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
253	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
333	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE INTERP (TOUT, Y, NO, YO)
C-----------------------------------------------------------------------
C SUBROUTINE INTERP COMPUTES INTERPOLATED VALUES OF THE DEPENDENT
C VARIABLE Y AND STORES THEM IN YO.  THE INTERPOLATION IS TO THE
C POINT T = TOUT AND USES THE NORDSIECK HISTORY ARRAY Y AS FOLLOWS..
C                                NQ
C                    YO(I)  =  SUM  Y(I,J+1)*S**J ,
C                              J=0
C WHERE S = -(T-TOUT)/H.
C-----------------------------------------------------------------------
C CAUTION   NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
C-----------------------------------------------------------------------
C
      DIMENSION YO(NO),Y(NO,13)
C
      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      ONE=1.0
C
      DO 10 I = 1,N
 10     YO(I) = Y(I,1)
      L = JSTART + 1
      S = (TOUT - T)/H
      S1 = ONE
      DO 30 J = 2,L
        S1 = S1*S
        DO 20 I = 1,N
 20       YO(I) = YO(I) + S1*Y(I,J)
 30     CONTINUE
      RETURN
C----------------------  END OF SUBROUTINE INTERP  -----------------
      END
```

```
      SUBROUTINE TSTEP(NO,NSO,Y,YMAX,ERROR,SAVE1,SAVE2,SAVE3,
     * PW,IPIV)
C-----------------------------------------------------------------------
C TSTEP PERFORMS ONE STEP OF THE INTEGRATION OF AN INITIAL VALUE
C PROBLEM FOR A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS.
C COMMUNICATION WITH TSTEP IS VIA THE FOLLOWING VARIABLES..
C
C   Y        AN NO BY LMAX ARRAY CONTAINING THE DEPENDENT VARIABLES
C              AND THEIR SCALED DERIVATIVES.  LMAX IS CURRENTLY 6 FOR
C              THE VARIABLE STEP BACKWARD DIFFERENTIATION FORMULAS,
C              AND 13 FOR THE VARIABLE STEP ADAMS FORMULAS.
C              (LMAX -1) = MAXDER, THE MAXIMUM ORDER USED.
C              SEE SUBROUTINE COSET.  Y(I,J+1) CONTAINS THE
C              J-TH DERIVATIVE OF Y(I), SCALED BY H**J/FACTORIAL(J)
C              FOR J = 0,1,...,NQ, WHERE NQ IS THE CURRENT ORDER.
C   NO       A CONSTANT INTEGER .GE. N, USED FOR DIMENSIONING
C              PURPOSES.
C   T        THE INDEPENDENT VARIABLE, UPDATED ON EACH STEP TAKEN.
C   H        THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.
C              H IS ALTERED BY THE ERROR CONTROL ALGORITHM DURING
C              THE SOLUTION OF THE PROBLEM. H CAN BE EITHER POSITIVE
C              OR NEGATIVE, BUT ITS SIGN MUST REMAIN CONSTANT
C              THROUGHOUT THE PROBLEM RUN.
C   HMIN,    THE MINIMUM AND MAXIMUM ABSOLUTE VALUES OF THE STEP
C    HMAX      SIZE TO BE USED FOR THE STEP. THESE MAY BE CHANGED AT
C              ANY TIME, BUT THE CHANGE WILL NOT TAKE EFFECT UNTIL THE
C              NEXT CHANGE IN H IS MADE.
C   EPS      THE RELATIVE ERROR BOUND.  SEE DESCRIPTION IN
C              SUBROUTINE DRIVE.
C   SS       THE SIZE OF THE TIME INTERVAL TO BE USED FOR ERROR
C              CONTROL.  A DEFAULT VALUE OF 0 IS USED TO PRODUCE
C              CONTROL OF ERROR PER STEP.  SEE SUBROUTINE DRIVE.
C   UROUND   THE UNIT OF ROUNDOFF FOR THE COMPUTER BEING USED.
C   N        THE NUMBER OF FIRST ORDER ORDINARY DIFFERENTIAL
C              EQUATIONS BEING SOLVED.
C   MF       THE METHOD FLAG.  SEE DESCRIPTION IN SUBROUTINE DRIVE.
C   KFLAG    A COMPLETION CODE WITH THE FOLLOWING MEANINGS..
C                      0  THE STEP WAS SUCCESFUL.
C                     -1  THE REQUESTED ERROR COULD NOT BE ACHIEVED
C                           WITH ABS(H) = HMIN.
C                     -2  THE REQUESTED ERROR IS SMALLER THAN CAN
C                           BE HANDLED FOR THIS PROBLEM.
C                     -3  CORRECTOR CONVERGENCE COULD NOT BE
C                           ACHIEVED FOR ABS(H) = HMIN.
C              ON A RETURN WITH KFLAG NEGATIVE, THE VALUES OF T AND
C              THE Y ARRAY ARE AS OF THE BEGINNING OF THE LAST
C              STEP AND H IS THE LAST STEP SIZE ATTEMPTED.
C   JSTART   AN INTEGER USED ON INPUT AND OUTPUT.
C              ON INPUT, IT HAS THE FOLLOWING VALUES AND MEANINGS..
C                      0  PERFORM THE FIRST STEP.
C                  .GT.0  TAKE A NEW STEP CONTINUING FROM THE LAST.
C                  .LT.0  TAKE THE NEXT STEP WITH A NEW VALUE OF
C                           H AND/OR MF.
C              ON EXIT, JSTART IS SET TO NQ, THE CURRENT ORDER OF THE
C              METHOD.
C   YMAX       AN ARRAY OF N ELEMENTS WITH WHICH THE ESTIMATED LOCAL
C                ERRORS IN Y ARE COMPARED.
```

```
C   ERROR      AN ARRAY OF N ELEMENTS.  ERROR(I)/TO(2) IS THE          024490
C                ESTIMATED LOCAL ERROR IN Y(I) PER SS UNITS OF         024500
C                T OR PER STEP (OF SIZE H).                            024510
C   SAVE1.     TWO ARRAYS FOR WORKING STORAGE.                         024520
C     SAVE2      EACH OF LENGTH N.                                     024530
C   PW         A BLOCK OF LOCATIONS USED FOR THE PARTIAL DERIVATIVES   024540
C                OF F WITH RESPECT TO Y, IF MITER IS NOT 0.  SEE       024550
C                DESCRIPTION IN SUBROUTINE DRIVE.                      024560
C   IPIV       AN INTEGER ARRAY OF LENGTH N, WHICH IS USED FOR PIVOT   024570
C                INFORMATION FOR THE LINEAR ALGEBRAIC SYSTEM IN THE    024580
C                CORRECTION PROCESS, WHEN MITER = 1 OR 2.              024590
C                                                                      024600
C THE COMMON BLOCK EPCM10, DECLARED BELOW, IS PRIMARILY INTENDED       024610
C FOR INTERNAL USE, BUT IT CAN BE ACCESSED EXTERNALLY.                 024620
C-----------------------------------------------------------------     024630
C                                                                      024640
      DIMENSION Y(N0,13)                                               024650
      DIMENSION YMAX(N0),ERROR(N0),SAVE1(N0),SAVE2(N0),PW(NS0),IPIV(N0) 024660
      DIMENSION SAVE3(N0)                                              024670
C                                                                      024680
      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART    024690
      COMMON/EPCOM9/HUSED,NQUSED,NSTEP,NFE,NJE,NMI                     024700
      COMMON/TABINT/KWRITE,KOUTD,SPEC,TITLE(8),TVENT(8),TDROP(8),TDIS(8)024710
      COMMON /EPCM10/ TAU(13),EL(13),TO(5),LMAX,METH,NQ,L,NQINDX       024720
      COMMON/TAB1/G0,GAM,COV,ENER,OFFSET                               024730
      COMMON/TAB2/RK1,RK2,RH1I,VII,VPER                                024740
      COMMON/TAB3/DC,AVENT,A1,A3,A4,AHOLE,VCOR                         024750
      COMMON/TAB4/PSWT,PSTRAV,MOD1,MBACK,MOD4                          024760
      COMMON/TAB5/C3,C4,RMD13,RMD34,RMD3,RMD4,RM1,RM3,RM4,RMSUM,RMINT  024770
      COMMON/TAB6/V1,VPS,SPS,RH1,P1,V3,RH3,P3,V4,VPJ,SPJ,APJ,RH4,P4    024780
      COMMON/TAB7/PJWT,TRAVEL                                          024790
      COMMON/TAB8/PSRS,PJRS,PL,PR,DC34                                 024800
      COMMON/TAB9/EL1,EG3,EG4,EKPS,EKPJ,FKG4,ESUM,EINT                 024810
      COMMON/TAB10/RU,WG,RS,CP,CV,TEMP3,TEMP4,HL1,HL3,HL4,H3,H4        024820
      COMMON/TAB11/LSKIP,LPAGE,TMS,RMAX(10)                            024830
      COMMON/TAB12/MVENT,NVENT,AVMIN,AVMAX,AK,ACS                      024840
      COMMON/TAB13/PI                                                  024850
      COMMON/TAB14/DDR,ADR,BDR                                         024860
      COMMON/TAB15/CL1,CL3,CG3,CL4,CG4,RMG3,RML3,RMG4,RML4             024870
      COMMON/TAB16/RHG3,RHL3,RHG4,RHL4,EL3,EL4,EKL4                    024880
      COMMON/TAB17/VG3,VL3,VG4,VL4,EPS3,EPS4                           024890
      COMMON/TAB18/DINC,PDDR(10),PML3(10),PMD3(10),PML4(10),PMD4(10)   024900
      COMMON/TAB19/SIGMA,U13,WE                                        024910
      DATA ISTEPJ/20/,KFC/-3/,KFH/-7/,MAXCOR/3/                        024920
C                                                                      024930
      DATA ADDON/1.0E-6/,BIAS1/2.5E1/,BIAS2/2.5E1/,BIAS3/1.0E2/,       024940
     1CRDOWN/0.1E0/,DELRC/0.3E0/,ETACF/0.25E0/,ETAMIN/0.1E0/,          024950
     2ETAMXF/0.2E0/,ETAMX1/1.0E4/,ETAMX2/1.0E1/,ETAMX3/1.5E0/,         024960
     3ONEPSM/1.00001E0/,SHORT/0.1E0/,THRESH/1.3E0/                     024970
      DATA ONE/1.0E0/,PT5/0.5E0/,ZERO/0.0E0/                           024980
      KFLAG = 0                                                        024990
      TOLD = T                                                         025000
      FLOTN=FLOAT(N)                                                   025010
      IF (JSTART .GT. 0) GO TO 200                                     025020
      IF (JSTART .NE. 0) GO TO 150                                     025030
C-----------------------------------------------------------------     025040
C ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL             025050
```

```
C DERIVATIVES ARE CALCULATED.  ETAMAX IS THE MAXIMUM RATIO BY
C WHICH H CAN BE INCREASED IN A SINGLE STEP.  IT IS 1.E04 FOR THE
C FIRST STEP TO COMPENSATE FOR THE SMALL INITIAL H, THEN 10 FOR
C THE NEXT 10 STEPS, AND THEN 1.5 THEREAFTER.  IF A FAILURE
C OCCURS (IN CORRECTOR CONVERGENCE OR ERROR TEST), ETAMAX IS SET AT 1
C FOR THE NEXT INCREASE.  ETAMIN = .1 IS THE MINIMUM RATIO BY WHICH
C H CAN BE REDUCED ON ANY RETRY OF A STEP.
C-----------------------------------------------------------------
      CALL DIFFUN(N,T,Y,SAVE1)
      DO 110 I = 1,N
 110    Y(I,2) = H*SAVE1(I)
      METH = MF/10
      MITER = MF - 10*METH
      MITER1 = MITER + 1
      MFOLD = MF
      NO = 1
      L = 2
      TAU(1) = H
      PRL1 = ONE
      RC = ZERO
      ETAMAX = ETAMX1
      NQINDX = 2
      NSTEP = 0
      NSTEPJ = 0
      NFE = 1
      NJE = 0
      TPRINT=.00005
      GO TO 200
C-----------------------------------------------------------------
C IF THE USER HAS CHANGED H, THEN Y MUST BE RESCALED.  IF THE
C USER HAS CHANGED MITER, THEN NEWJ IS SET TO MITER TO FORCE
C THE PARTIAL DERIVATIVEES TO BE UPDATED, IF THEY ARE BEING USED.
C-----------------------------------------------------------------
 150  IF (MF .EQ. MFOLD) GO TO 170
      MIO = MITER
      METH = MF/10
      MITER = MF - 10*METH
      MFOLD = MF
      IF (MITER .EQ. MIO) GO TO 170
      NEWJ = MITER
      MITER1 = MITER + 1
 170  IF (H .EQ. HOLD) GO TO 200
      ETA = H/HOLD
      H = HOLD
      IREDO = 3
      GO TO 185
 180  ETA=AMAX1(ETA,HMIN/ABS(H),ETAMIN)
 185  ETA=AMIN1(ETA,HMAX/ABS(H),ETAMAX)
      R1 = ONE
      DO 190 J = 2,L
        R1 = R1*ETA
        DO 190 I = 1,N
 190      Y(I,J) = Y(I,J)*R1
      H = H*ETA
      RC = RC*ETA
      IF (IREDO .EQ. 0) GO TO 690
C-----------------------------------------------------------------
```

```
      C THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY
      C MULTIPLYING THE Y ARRAY BY THE PASCAL TRIANGLE MATRIX.  THEN
      C COSET IS CALLED TO OBTAIN EL, THE VECTOR OF COEFFICIENTS OF
      C LENGTH NO + 1.  RC IS THE RATIO OF NEW TO OLD VALUES OF THE
      C COEFFICIENT H/EL(2).  WHEN RC DIFFERS FROM 1 BY MORE THAN
      C DELPC, NEWJ IS SET TO MITER TO FORCE THE PARTIAL DERIVATIVES
      C TO BE UPDATED, IF USED.  DELRC IS 0.3.  IN ANY CASE, THE PARTIAL
      C DERIVATIVES ARE UPDATED AT LEAST EVERY 20-TH STEP.
      C-----------------------------------------------------------------
  200 T = T + H
      DO 210 J1 = 1,NO
        DO 210 J2 = J1,NO
          J = (NO + J1) - J2
          DO 210 I = 1,N
  210       Y(I,J) = Y(I,J) + Y(I,J+1)
      CALL COSET
      BND = FLOTN*(TO(4)*EPS)**2
      RL1 = ONE/EL(2)
      RC = RC*(RL1/PRL1)
      PRL1 = RL1
      IF (NSTEP .GE. NSTEPJ+ISTEPJ) NEWJ = MITER
      DRC=ABS(RC-ONE)
      IF (DRC .LE. DELPC) GO TO 215
      NEWJ = MITER
      CRATE = ONE
      RC = ONE
      GO TO 220
  215 IF ((MITER .NE. 0) .AND. (DRC .NE. ZERO)) CRATE = ONE
      C-----------------------------------------------------------------
      C UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  A CONVERGENCE TEST IS MADE
      C ON THE ROOT MEAN SQUARE NORM OF EACH CORRECTION, USING BND, WHICH
      C IS DEPENDENT ON EPS.  THE SUM OF THE CORRECTIONS IS ACCUMULATED IN
      C THE VECTOR ERROR.  THE Y ARRAY IS NOT ALTERED IN THE CORRECTOR
      C LOOP.  THE UPDATED Y VECTOR IS STORED TEMPORARILY IN SAVE1.
      C-----------------------------------------------------------------
  220 DO 230 I = 1,N
  230   ERROR(I) = ZERO
      M = 0
      CALL DIFFUN(N,T,Y,SAVE2)
      NFE = NFE + 1
      IF (NEWJ .LE. 0) GO TO 290
      C-----------------------------------------------------------------
      C IF INDICATED, THE MATRIX P = I - H*RL1*J IS REEVALUATED BEFORE
      C STARTING THE CORRECTOR ITERATION.  NEWJ IS SET TO 0 AS AN
      C INDICATOR THAT THIS HAS BEEN DONE.  IF MITER = 1 OR 2, P IS
      C COMPUTED AND PROCESSED IN PSET.  IF MITER = 3, THE MATRIX  IS
      C P = I - H*RL1*D, WHERE D IS A DIAGONAL MATRIX.  RL1 IS 1/EL(2).
      C-----------------------------------------------------------------
      NEWJ = 0
      RC = ONE
      NJE = NJE + 1
      NSTEPJ = NSTEP
      GO TO (250, 240, 260), MITER
  240 NFE = NFE + N
  250 CON = -H*RL1
      CALL PSET(Y,NO,CON,MITER,IER,NSO,YMAX,SAVE1,SAVE2,SAVE3,
     * PW,IPIV)
```

```
      IF (IER .NE. 0) GO TO 420
      GO TO 350
  260 R = RL1*SHORT
      DO 270 I = 1,N
  270   PW(I) = Y(I,1) + R*(H*SAVE2(I) - Y(I,2))
      CALL DIFFUN(N,T,PW,SAVE1)
      NFE = NFE + 1
      HRL1 = H*RL1
      DO 280 I = 1,N
        R0 = H*SAVE2(I) - Y(I,2)
        PW(I) = ONE
        D = SHORT*R0 - H*(SAVE1(I) - SAVE2(I))
        SAVE1(I) = ZERO
      IF(ABS(R0).LT.UROUND*YMAX(I))GO TO 280
      IF(ABS(D).EQ.ZERO)GO TO 420
        PW(I) = SHORT*R0/D
        SAVE1(I) = PW(I)*RL1*R0
  280   CONTINUE
      GO TO 370
  290 GO TO (295, 350, 350, 310), MITER1
C-----------------------------------------------------------------
C IN THE CASE OF FUNCTIONAL ITERATION, Y IS UPDATED DIRECTLY FROM
C THE RESULT OF THE LAST DIFFUN CALL.
C-----------------------------------------------------------------
  295 D = ZERO
      DO 300 I = 1,N
        R = RL1*(H*SAVE2(I) - Y(I,2))
        D = D + ((R - ERROR(I))/YMAX(I))**2
        SAVE1(I) = Y(I,1) + R
  300   ERROR(I) = R
      GO TO 400
C-----------------------------------------------------------------
C IN THE CASE OF A CHORD METHOD, THE RESIDUAL -G(Y SUB N(M))
C IS COMPUTED AND THE LINEAR SYSTEM WITH THAT AS RIGHT-HAND SIDE
C AND P AS COEFFICIENT MATRIX IS SOLVED, USING THE LU DECOMPOSITION
C OF P IF MITER = 1 OR 2.  IF MITER = 3 THE SCALAR H*RL1 IS UPDATED.
C-----------------------------------------------------------------
  310 PHRL1 = HRL1
      HRL1 = H*RL1
      IF (HRL1 .EQ. PHRL1) GO TO 330
      R = HRL1/PHRL1
      DO 320 I = 1,N
        D = ONE - R*(ONE - ONE/PW(I))
      IF(ABS(D).EQ.ZERO)GO TO 440
  320   PW(I) = ONE/D
  330 DO 340 I = 1,N
  340   SAVE1(I) = PW(I)*(RL1*H*SAVE2(I) - (PL1*Y(I,2) + ERROR(I)))
      GO TO 370
  350 DO 360 I = 1,N
  360 SAVE3(I)=RL1*H*SAVE2(I)-(RL1*Y(I,2)+ERROR(I))
      CALL SOLVE(N,NO,PW,IPIV,SAVE3,SAVE1)
  370 D = ZERO
      DO 380 I = 1,N
        ERROR(I) = ERROR(I) + SAVE1(I)
        D = D + (SAVE1(I)/YMAX(I))**2
  380   SAVE1(I) = Y(I,1) + ERROR(I)
C-----------------------------------------------------------------
```

```
C TEST FOR CONVERGENCE.  IF M .GT. 0. AN ESTIMATE OF THE SQUARE OF
C THE CONVERGENCE RATE CONSTANT IS STORED IN CRATE, AND THIS IS USED
C IN THE TEST.
C-----------------------------------------------------------------
400   IF(M.NE.0)CRATE=AMAX1(CRDOWN*CRATE,D/D1)
      IF(D*AMIN1(ONE,CRATE).LE.RND)GO TO 450
      D1 = D
      M = M + 1
      IF(D.GE.1000.0*RND)M=MAXCOR
      IF (M .EQ. MAXCOR) GO TO 410
      CALL DIFFUN(N,T,SAVE1,SAVE2)
      GO TO (295, 350, 350, 310), MITER1
C-----------------------------------------------------------------
C THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES. IF PARTIAL
C DERIVATIVES ARE INVOLVED BUT ARE NOT UP TO DATE, THEY ARE
C REEVALUATED FOR THE NEXT TRY.  OTHERWISE THE Y ARRAY IS RESTORED
C TO ITS VALUES BEFORE PREDICTION, AND H IS REDUCED,
C IF POSSIBLE.  IF NOT, A NO-CONVERGENCE EXIT IS TAKEN.
C-----------------------------------------------------------------
 410  NFE = NFE + MAXCOR - 1
      IF (NEWJ .EQ. -1) GO TO 440
 420  T = TOLD
      ETAMAX = ONE
      DO 430 J1 = 1,NQ
        DO 430 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 430 I = 1,N
 430        Y(I,J) = Y(I,J) - Y(I,J+1)
      IF(ABS(H).LE.HMIN*ONEPSM)GO TO 680
      ETA = ETACF
      IREDO = 1
      GO TO 180
 440  NEWJ = MITER
      GO TO 220
C-----------------------------------------------------------------
C THE CORRECTOR HAS CONVERGED.  NEWJ IS SET TO -1 IF PARTIAL
C DERIVATIVES WERE USED, TO SIGNAL THAT THEY MAY NEED UPDATING ON
C SUBSEQUENT STEPS.  THE ERROR TEST IS MADE AND CONTROL PASSES TO
C STATEMENT 500 IF IT FAILS.
C-----------------------------------------------------------------
 450  IF (MITER .NE. 0) NEWJ = -1
      NFE = NFE + M
      D = ZERO
      DO 460 I = 1,N
 460    D = D + (ERROR(I)/YMAX(I))**2
      E = FLOTN*(TQ(2)*EPS)**2
C  IF V1 IS NEGATIVE, WE HAVE GONE TOO FAR.
C  CUT BACK THE TIME STEP.
      IF(V1.LE.0.0)GO TO 500
      IF (D .GT. E) GO TO 500
C-----------------------------------------------------------------
C AFTER A SUCCESSFUL STEP, THE Y ARRAY, TAU, NSTEP, AND NQINDX ARE
C UPDATED, AND A NEW VALUE OF H AT ORDER NQ IS COMPUTED.
C THE VECTOR TAU CONTAINS THE NQ + 1 MOST RECENT VALUES OF H.
C A CHANGE IN NQ UP OR DOWN BY 1 IS CONSIDERED IF NQINDX = 0.
C IF NQINDX = 1 AND NQ .LT. MAXDER, THEN ERROP IS SAVED
C FOR USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.
```

```
C A CHANGE IN H OR NQ IS MADE ONLY OF THE INCREASE IN H
C IS BY A FACTOR OF AT LEAST 1.3.
C IF NOT, NQINDX IS SET TO 2 TO PREVENT TESTING FOR THAT MANY
C STEPS.  IF NQ IS CHANGED, NQINDX IS SET TO NQ + 1 (NEW VALUE).
C-----------------------------------------------------------------
      KFLAG = 0
      IREDO = 0
      NSTEP = NSTEP + 1
      HUSED = H
      NQUSED = NQ
      DO 470 IBACK = 1,NQ
        I = L - IBACK
 470    TAU(I+1) = TAU(I)
      TAU(1) = H
      DO 480 J = 1,L
        DO 480 I = 1,N
 480      Y(I,J) = Y(I,J) + ERROR(I)*EL(J)
      NQINDX = NQINDX - 1
      IF ((L .EQ. LMAX) .OR. (NQINDX .NE. 1)) GO TO 495
      DO 490 I = 1,N
 490    Y(I,LMAX) = ERROR(I)
      CONP = TQ(5)
 495  IF (ETAMAX .NE. ONE) GO TO 520
      IF (NQINDX .LT. 2) NQINDX = 2
      GO TO 690
C-----------------------------------------------------------------
C THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.
C T AND THE Y ARRAY ARE RESTORED TO THEIR PREVIOUS VALUES.  A NEW
C H FOR A RETRY OF THE STEP IS COMPUTED.  THE ORDER IS KEPT FIXED.
C-----------------------------------------------------------------
 500  KFLAG = KFLAG - 1
      T = TOLD
      DO 510 J1 = 1,NQ
        DO 510 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 510 I = 1,N
 510        Y(I,J) = Y(I,J) - Y(I,J+1)
      NEWJ = MITER
      ETAMAX = ONE
      IF(ABS(H).LE.HMIN*ONEPSM)GO TO 660
      IF (KFLAG .LE. KFC) GO TO 630
      IREDO = 2
 520  FLOTL=FLOAT(L)
      ETAQ = ONE/((BIAS2*D/E)**(PT5/FLOTL) + ADDON)
      IF ((NQINDX .NE. 0) .OR. (KFLAG .NE. 0)) GO TO 580
      ETAQM1 = ZERO
      IF (NQ .EQ. 1) GO TO 540
C COMPUTE RATIO OF NEW H TO CURRENT H AT THE CURRENT ORDER LESS ONE. -
      D = ZERO
      DO 530 I = 1,N
 530    D = D + (Y(I,L)/YMAX(I))**2
      EDN = FLOTN*(TQ(1)*EPS)**2
      ETAQM1 = ONE/((BIAS1*D/EDN)**(PT5/(FLOTL - ONE)) + ADDON)
 540  ETAQP1 = ZERO
      IF (L .EQ. LMAX) GO TO 560
C COMPUTE RATIO OF NEW H TO CURRENT H AT CURRENT ORDER PLUS ONE. -----
      CNQUOT = (TQ(5)/CONP)*(H/TAU(2))**L
```

```
      D = ZERO
      DO 550 I = 1,N
  550   D = D + ((ERROR(I) - CNQUOT*Y(I,LMAX))/YMAX(I))**2
      EUP = FLOTN*(TQ(3)*EPS)**2
      ETAQP1 = ONE/((BIAS3*D/EUP)**(PT5/(FLOTL + ONE)) + ADDON)
  560 NQINDX = 2
      IF (ETAQ .GE. ETAQP1) GO TO 570
      IF (ETAQP1 .GT. ETAQM1) GO TO 600
      GO TO 590
  570 IF (ETAQ .LT. ETAQM1) GO TO 590
  580 IF ((ETAQ .LT. THRESH) .AND. (KFLAG .EQ. 0)) GO TO 690
      ETA = ETAQ
      IF ((KFLAG .LE. -2) .AND. (ETA .GT. ETAMXF)) ETA = ETAMXF
      GO TO 180
  590 IF (ETAQM1 .LT. THRESH) GO TO 690
      CALL ADJUST (Y, NQ)
      L = NQ
      NQ = NQ - 1
      ETA = ETAQM1
      NQINDX = L
      GO TO 180
  600 IF (ETAQP1 .LT. THRESH) GO TO 690
      NQ = L
      ETA = ETAQP1
      L = L + 1
      DO 610 I = 1,N
  610   Y(I,L) = ZERO
      NQINDX = L
      GO TO 180
C-----------------------------------------------------------------------
C CONTROL REACHES THIS SECTION IF 3 OR MORE CONSECUTIVE FAILURES
C HAVE OCCURRED.  IT IS ASSUMED THAT THE ELEMENTS OF THE Y ARRAY
C HAVE ACCUMULATED ERRORS OF THE WRONG ORDER.  THE ORDER IS REDUCED
C BY ONE, IF POSSIBLE.  THEN H IS REDUCED BY A FACTOR OF 0.1 AND
C THE STEP IS RETRIED.  AFTER A TOTAL OF 7 CONSECUTIVE FAILURES,
C AN EXIT IS TAKEN WITH KFLAG = -2.
C-----------------------------------------------------------------------
  630 IF (KFLAG .EQ. KFH) GO TO 670
      IF (NQ .EQ. 1) GO TO 640
      ETA = ETAMIN
      CALL ADJUST (Y, NQ)
      L = NQ
      NQ = NQ - 1
      NQINDX = L
      GO TO 180
  640 ETA=AMAX1(ETAMIN,HMIN/ABS(H))
      H = H*ETA
      CALL DIFFUN(N,T,Y,SAVE1)
      NFE = NFE + 1
      DO 650 I = 1,N
  650   Y(I,2) = H*SAVE1(I)
      NQINDX = 10
      GO TO 200
C-----------------------------------------------------------------------
C ALL RETURNS ARE MADE THROUGH THIS SECTION.  H IS SAVED IN HOLD
C TO ALLOW THE CALLER TO CHANGE H ON THE NEXT STEP.
C-----------------------------------------------------------------------
```

```
  660  KFLAG = -1                                                    028480
       GO TO 700                                                     028490
  670  KFLAG = -2                                                    028500
       GO TO 700                                                     028510
  680  KFLAG = -3                                                    028520
       GO TO 700                                                     028530
  690  ETAMAX = ETAMX3                                               028540
       IF (NSTEP .LE. 10) ETAMAX = ETAMX2                            028550
  700  HOLD = H                                                      028560
       JSTART = NQ                                                   028570
C  IF V1 IS SMALL ENOUGH, COLLAPSE REGION 1.                         028580
       IF(V1.LT.VPER*V1I)MOD1=0                                      028590
C  IF THE PROJECTILE TRAVEL IS LARGE ENOUGH, OPEN REGION 4.          028600
       IF((OFFSET+SPJ).GE.0.01)MOD4=1                                028610
       IF(MVENT.NE.1)GO TO 800                                       028620
       IF(PR.LT.PJRS)GO TO 800                                       028630
C  ADJUST THE VENT OPENING.                                          028640
C  FIND THE DESIRED BASE PRESSURE.                                   028650
       PBASE=PJWT*0.098146*ACS/A4 - PJRS                             028660
       RATIO=(PBASE-PR)/PBASE                                        028670
       AVENT=AVENT*(1.0+AK*RATIO)                                    028680
       IF(AVENT.LT.AVMIN)AVENT=AVMIN                                 028690
       IF(AVENT.GT.AVMAX)AVENT=AVMAX                                 028700
  800  CONTINUE                                                      028710
       IF(KWRITE.NE.1)RETURN                                         028720
       WRITE(6,1000)NSTEP,NQ,T,H,MOD1,MOD4                           028730
 1000  FORMAT(/2X,2I4,1P2E12.4,6X,2I4)                               028740
       TMS=T*1000.                                                   028750
       CALL DIFFUN(N,T,Y,SAVE2)                                      028760
       WRITE(6,1010)TMS,P1,P3,PL,P4,PR,SPS,VPS,SPJ,VPJ,APJ           028770
       WRITE(6,1010)TMS,RH1,RHL3,RHG3,RH3,RHL4,RHG4,RH4              028780
       WRITE(6,1010)TMS,V1,VL3,VG3,V3,VL4,VG4,V4,EPS3,EPS4,AVENT     028790
       WRITE(6,1010)TMS,RM1,RML3,RMG3,RM3,RML4,RMG4,RM4,RMSUM        028800
       WRITE(6,1010)TMS,RMD13,RMD3,RMD34,RMD4,HL1,HL3,H3,HL4,H4      028810
       WRITE(6,1010)TMS,CL1,CL3,CG3,C3,CL4,CG4,C4,TEMP3,TEMP4        028820
       WRITE(6,1010)TMS,EL1,EL3,EG3,EL4,EG4,EKPS,EKPJ,EKL4,EKG4,ESUM 028830
       WRITE(6,1010)TMS,(PML3(I),I=1,10)                             028840
       WRITE(6,1010)TMS,(PMD3(I),I=1,10)                             028850
       WRITE(6,1010)TMS,(PML4(I),I=1,10)                             028860
       WRITE(6,1010)TMS,(PMD4(I),I=1,10)                             028870
 1010   FORMAT(1P11E11.4)                                            028880
       WRITE(6,1020)(Y(I,1),I=1,N)                                   028890
       WRITE(6,1020)(SAVE2(I),I=1,N)                                 028900
 1020  FORMAT(1P10E12.4)                                             028910
       RETURN                                                        028920
C---------------------- END OF SUBROUTINE TSTEP ------------------- 028930
       END                                                           028940
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
224	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE COSET
C-----------------------------------------------------------------------
C COSET IS CALLED BY TSTEP AND SETS COEFFICIENTS FOR USE THERE.
C
C FOR EACH ORDER NO, THE COEFFICIENTS IN EL ARE CALCULATED BY USE OF
C  THE GENERATING POLYNOMIAL LAMBDA(X), WITH COEFFICIENTS EL(I)
C      LAMBDA(X) = EL(1) + EL(2)*X + ... + EL(NO+1)*(X**NO).
C FOR THE BACKWARD DIFFERENTIATION FORMULAS.
C                          NO
C      LAMBDA(X) = PRODUCT (1 + X/XI(I) ) .
C                        I = 1
C FOR THE ADAMS FORMULAS,
C                                 NO-1
C      (D/DX) LAMBDA(X) = C * PRODUCT (1 + X/XI(I) ) ,
C                                I = 1
C      LAMBDA(-1) = 0,    LAMBDA(0) = 1,
C WHERE C IS A NORMALIZATION CONSTANT.
C IN BOTH CASES, XI(I) IS DEFINED BY
C      H*XI(I) = T SUB N  -  T SUB (N-I)
C              = H + TAU(1) + TAU(2) + ... TAU(I-1).
C
C COSET ALSO SETS MAXDER, THE MAXIMUM ORDER OF THE FORMULAS
C AVAILABLE. CURRENTLY THIS IS 5 FOR THE BACKWARD DIFFERENTIATION
C FORMULAS, AND 12 FOR THE ADAMS FORMULAS.  TO USE DIFFERENT
C VALUES (.LE. 13),  CHANGE THE NUMBERS IN STATEMENTS 1 AND 2 BELOW.
C
C IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
C WITH COSET USES THE FOLLOWING..
C   TAU    = A VECTOR OF LENGTH 13 CONTAINING THE PAST NQ VALUES
C            OF H.
C   EL     = A VECTOR OF LENGTH 13 IN WHICH COSET STORES THE
C            COEFFICIENTS FOR THE CORRECTOR FORMULA.
C   TQ     = A VECTOR OF LENGTH 5 IN WHICH COSET STORES CONSTANTS
C            USED FOR THE CONVERGENCE TEST, THE ERROR TEST, AND
C            SELECTION OF H AT A NEW ORDER.
C   LMAX   = MAXDER + 1, WHERE MAXDER IS THE MAXIMUM ORDER
C            AVAILABLE.  LMAX IS THE MAXIMUM NUMBER OF COLUMNS
C            OF THE Y ARRAY TO BE USED.
C   METH   = THE BASIC METHOD INDICATOR.
C   NQ     = THE CURRENT ORDER.
C   L      = NQ + 1, THE LENGTH OF THE VECTOR STORED IN EL, AND
C            THE NUMBER OF COLUMNS OF THE Y ARRAY BEING USED.
C   NQINDX = A COUNTER CONTROLLING THE FREQUENCY OF ORDER CHANGES.
C            AN ORDER CHANGE IS ABOUT TO BE CONSIDERED IF
C            NQINDX = 1.
C-----------------------------------------------------------------------
C CAUTION   NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
C-----------------------------------------------------------------------
      DIMENSION EM(13)
C
      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      COMMON /EPCM10/ TAU(13),EL(13),TQ(5),LMAX,METH,NQ,L,NQINDX
      DATA CORTES/0.1E0/
      DATA ONE/1.0E0/,SIX/6.0E0/,TWO/2.0E0/,ZERO/0.0E0/
      AHOSS=ONE
      IF(SS.NE.ZERO)AHOSS=ABS(H)/SS
      FLOTL=FLOAT(L)
```

```
      NQM1 = NQ - 1
      GO TO (1, 2), METH
    1 MAXDER = 12
      GO TO 100
C
    2 MAXDER = 5
      GO TO 200
C
  100 IF (NQ .NE. 1) GO TO 110
      EL(1) = ONE
      EL(2) = ONE
      TQ(1) = ONE
      TQ(2) = TWO*AHDSS
      TQ(3) = SIX*TQ(2)
      TQ(5) = ONE
      GO TO 300
  110 HSUM = H
      EM(1) = ONE
      FLOTNQ = FLOTL - ONE
      DO 115 I = 2,L
  115   EM(I) = ZERO
      DO 150 J = 1,NQM1
        IF ((J .NE. NQM1) .OR. (NQINDX .NE. 1)) GO TO 130
        S = ONE
        CSUM = ZERO
        DO 120 I = 1,NQM1
      CSUM=CSUM+S*EM(I)/FLOAT(I+1)
  120     S = -S
        TQ(1) = AHDSS*EM(NQM1)/(FLOTNQ*CSUM)
  130   RXI = H/HSUM
        DO 140 IBACK = 1,J
          I = (J + 2) - IBACK
  140     EM(I) = EM(I) + EM(I-1)*RXI
  150   HSUM = HSUM + TAU(J)
C COMPUTE INTEGRAL FROM -1 TO 0 OF POLYNOMIAL AND OF X TIMES IT. -----
      S = ONE
      EM0 = ZERO
      CSUM = ZERO
      DO 160 I = 1,NQ
      FLOTI=FLOAT(I)
        EM0 = EM0 + S*EM(I)/FLOTI
        CSUM = CSUM + S*EM(I)/(FLOTI+1)
  160   S = -S
C IN EL, FORM COEFFICIENTS OF NORMALIZED INTEGRATED POLYNOMIAL. ------
      S = ONE/EM0
      EL(1) = ONE
      DO 170 I = 1,NQ
  170 EL(I+1)=S*EM(I)/FLOAT(I)
      XI = HSUM/H
      TQ(2) = AHDSS*XI*EM0/CSUM
      TQ(5) = XI/EL(L)
      IF (NQINDX .NE. 1) GO TO 300
C FOR HIGHER ORDER CONTROL CONSTANT, MULTIPLY POLYNOMIAL BY 1+X/XI(Q).
      RXI = ONE/XI
      DO 180 IBACK = 1,NQ
        I = (L + 1) - IBACK
  180   EM(I) = EM(I) + EM(I-1)*RXI
```

```
      C COMPUTE INTEGRAL OF POLYNOMIAL. ------------------------------------
            S = ONE
            CSUM = ZERO
            DO 190 I = 1,L
            CSUM=CSUM+S*EM(I)/FLOAT(I+1)
       190    S = -S
            TQ(3) = AHDSS*FLOTL*EM0/CSUM
            GO TO 300
      C
       200  DO 210 I = 3,L
       210    EL(I) = ZERO
            EL(1) = ONE
            EL(2) = ONE
            HSUM = H
            HSUM1 = ZERO
            PROD = ONE
            RXI = ONE
            IF (NQ .EQ. 1) GO TO 240
            DO 230 J = 1,NQM1
      C IN EL, CONSTRUCT COEFFICIENTS OF (1+X/XI(1))*...*(1+X/XI(J+1)). ----
              HSUM = HSUM + TAU(J)
              HSUM1 = HSUM1 + TAU(J)
              PROD = PROD*(HSUM/HSUM1)
              RXI = H/HSUM
              JP1 = J + 1
              DO 220 IBACK=1,JP1
              I=  (J + 3) - IBACK
        220   EL(I)=EL(I) + EL(I-1)*RXI
       230    CONTINUE
       240  TQ(2) = AHDSS*EL(2)*(ONE + PROD)
            TQ(5) = (ONE + PROD)/EL(L)
            IF (NQINDX .NE. 1) GO TO 300
            CNQM1 = RXI/EL(L)
            ELP = EL(2) - RXI
            TQ(1) = AHDSS*ELP/CNQM1
            HSUM = HSUM + TAU(NQ)
            RXI = H/HSUM
            ELP = EL(2) + RXI
            TQ(3) = AHDSS*ELP*RXI*(ONE + PROD)*(FLOTL + ONE)
       300  TQ(4) = CORTES*TQ(2)
            LMAX = MAXDER + 1
            RETURN
      C----------------------  END OF SUBROUTINE COSET  -------------------
            END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
59	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE ADJUST (Y, NO)
C-----------------------------------------------------------------------
C THIS SUBROUTINE ADJUSTS THE Y ARRAY ON REDUCTION OF ORDER.
C SEE REFERENCE 1 FOR DETAILS.
C-----------------------------------------------------------------------
C CAUTION   NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
C-----------------------------------------------------------------------
      DIMENSION Y(NO,13)
C
      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,CS,UROUND,N,MF,KFLAG,JSTART
      COMMON /EPCM10/ TAU(13),EL(13),TQ(5),LMAX,METH,NQ,L,NQINDX
      DATA ONE/1.0E0/,ZERO/0.0E0/
      IF (NQ .EQ. 2) RETURN
      NQM1 = NQ - 1
      NQM2 = NQ - 2
      GO TO (100, 200), METH
C
 100  DO 110 J = 1,LMAX
 110    EL(J) = ZERO
      EL(2) = ONE
      HSUM = ZERO
      DO 130 J = 1,NQM2
C CONSTRUCT COEFFICIENTS OF X*(X+XI(1))*...*(X+XI(J)). ---------------
        HSUM = HSUM + TAU(J)
        XI = HSUM/H
        JP1 = J + 1
        DO 120 IBACK = 1,JP1
          I = (J + 3) - IBACK
 120      EL(I) = EL(I)*XI + EL(I-1)
 130    CONTINUE
C CONSTRUCT COEFFICIENTS OF INTEGRATED POLYNOMIAL. -------------------
      DO 140 J = 2,NQM1
140   EL(J+1)=FLOAT(NQ)*EL(J)/FLOAT(J)
      GO TO 300
C
 200  DO 210 J = 1,LMAX
 210    EL(J) = ZERO
      EL(3) = ONE
      HSUM = ZERO
      DO 230 J = 1,NQM2
C CONSTRUCT COEFFICIENTS OF X*X*(X+XI(1))*...*(X+XI(J)). -------------
        HSUM = HSUM + TAU(J)
        XI = HSUM/H
        JP1 = J + 1
        DO 220 IBACK = 1,JP1
          I = (J + 4) - IBACK
 220      EL(I) = EL(I)*XI + EL(I-1)
 230    CONTINUE
C
C SUBTRACT CORRECTION TERMS FROM Y ARRAY. ---------------------------
 300  DO 320 J = 3,NQ
        DO 310 I = 1,N
 310      Y(I,J) = Y(I,J) - Y(I,L)*EL(J)
 320    CONTINUE
      RETURN
C----------------------  END OF SUBROUTINE ADJUST  ------------------
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
16	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```
      SUBROUTINE PSET(Y,N0,CON,MITER,IER,NSQ,YMAX,SAVE1,SAVE2,
     * SAVE3,PW,IPIV)
C-----------------------------------------------------------------------
C PSET IS CALLED BY TSTEP TO COMPUTE AND TO PROCESS THE MATRIX
C P = I - (H/EL(2))*J, WHERE J IS AN APPROXIMATION TO THE
C JACOBIAN.  J IS COMPUTED BY EITHER THE USER SUPPLIED
C SUBROUTINE PEDERV, WHEN MITER = 1, OR BY FINITE DIFFERENCES,
C WHEN MITER = 2.  J IS STORED IN PW AND REPLACED BY P, USING
C CON = -H/EL(2).  THEN P IS SUBJECTED TO AN LU DECOMPOSITION
C FOR LATER SOLUTION OF LINEAR ALGEBRAIC SYSTEMS WITH P AS THE
C COEFFICIENT MATRIX.
C
C IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
C WITH PSET USES THE FOLLOWING..
C   EPSJ = SQRT(UROUND), USED IN THE NUMERICAL JACOBIAN INCREMENTS.
C   NSQ  = N0**2.
C-----------------------------------------------------------------------
C CAUTION   NOT ALL EPCOM1 VARIABLES ARE USED INTHIS SUBROUTINE.
C-----------------------------------------------------------------------
      DIMENSION Y(N0,13)
      DIMENSION YMAX(N0),SAVE1(N0),SAVE2(N0),PW(NSQ),IPIV(N0)
      DIMENSION SAVE3(N0)
C
      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      COMMON/EPCOM8/EPSJ,NSQR
      COMMON/EPCOM9/HUSED,NQUSED,NSTEP,NFE,NJE,NMI
      DATA ONE/1.0E0/,PEP/1.0E-3/,ZERO/0.0E0/
      IF (MITER .EQ. 2) GO TO 20
C IF MITER = 1, CALL PEDERV AND MULTIPLY BY A SCALAR.  ---------------
      CALL PEDERV (N, T, Y, PW, N0)
      DO 10 I = 1,NSQ
   10   PW(I) = PW(I)*CON
      GO TO 60
C IF MITER = 2, MAKE N CALLS TO DIFFUN TO APPROXIMATE J. -------------
   20 D = ZERO
      DO 30 I = 1,N
   30   D = D + SAVE2(I)**2
      R0=ABS(H)*SQRT(D)*UROUND/PEP
      J1 = 0
      DO 50 J = 1,N
        YJ = Y(J,1)
        P = EPSJ*YMAX(J)
        R=AMAX1(P,R0)
        Y(J,1) = Y(J,1) + R
        D = CON/R
      CALL DIFFUN(N0,T,Y,SAVE1)
        DO 40 I = 1,N
   40     PW(I+J1) = (SAVE1(I) - SAVE2(I))*D
        Y(J,1) = YJ
        J1 = J1 + N0
   50 CONTINUE
C ADD ON THE IDENTITY MATRIX.  ---------------------------------------
   60 J = 1
      DO 70 I = 1,N
        PW(J) = PW(J) + ONE
   70   J = J + (N0 + 1)
```

```
      C GET LU DECOMPOSITION OF P. ------------------------------------------ 031670
            CALL DECOMP(N,N0,PW,SAVE3,IPIV)                                   031680
60          NMI=NMI+1                                                         031690
            RETURN                                                            031700
      C----------------------- END OF SUBROUTINE PSET ---------------------- 031710
            END                                                               031720
```

```
      SUBROUTINE DECOMP(N,NDIM,UL,SCALES,IPS)                           031730
      DIMENSION UL(NDIM,NDIM),SCALES(NDIM),IPS(NDIM)                    031740
C  THIS IS A STANDARD ALGORITHM FOR SOLVING A SET OF LINEAR             031750
C  ALGEBRAIC EQUATIONS USING AN LU DECOMPOSITION AND                    031760
C  BACK SUBSTITUTION.                                                   031770
C  REFERENCE.                                                           031780
C  FORSYTHE AND MOLER.  COMPUTER SOLUTION OF LINEAR ALGEBRAIC SYSTEMS.  031790
      DO 5 I=1,N                                                        031800
      IPS(I)=I                                                          031810
      ROWNRM=0.0                                                        031820
      DO 2 J=1,N                                                        031830
      IF(ROWNRM-ABS(UL(I,J)))1,2,2                                      031840
1     ROWNRM=ABS(UL(I,J))                                               031850
2     CONTINUE                                                          031860
      IF(ROWNRM)3,4,3                                                   031870
3     SCALES(I)=1.0/ROWNRM                                              031880
      GO TO 5                                                           031890
4     CALL SING(1)                                                      031900
      SCALES(I)=0.0                                                     031910
5     CONTINUE                                                          031920
      NM1=N-1                                                           031930
      DO 17 K=1,NM1                                                     031940
      BIG=0.0                                                           031950
      DO 11 I=K,N                                                       031960
      IP=IPS(I)                                                         031970
      SIZE=ABS(UL(IP,K))*SCALES(IP)                                     031980
      IF(SIZE-BIG)11,11,10                                              031990
10    BIG=SIZE                                                          032000
      IDXPIV=I                                                          032010
11    CONTINUE                                                          032020
      IF(BIG)13,12,13                                                   032030
12    CALL SING(2)                                                      032040
      GO TO 17                                                          032050
13    IF(IDXPIV-K)14,15,14                                              032060
14    J=IPS(K)                                                          032070
      IPS(K)=IPS(IDXPIV)                                                032080
      IPS(IDXPIV)=J                                                     032090
15    KP=IPS(K)                                                         032100
      PIVOT=UL(KP,K)                                                    032110
      KP1=K+1                                                           032120
      DO 16 I=KP1,N                                                     032130
      IP=IPS(I)                                                         032140
      EM=-UL(IP,K)/PIVOT                                                032150
      UL(IP,K)=-EM                                                      032160
      DO 16 J=KP1,N                                                     032170
      UL(IP,J)=UL(IP,J)+EM*UL(KP,J)                                     032180
16    CONTINUE                                                          032190
17    CONTINUE                                                          032200
      KP=IPS(N)                                                         032210
      IF(UL(KP,N).NE.0.0)GO TO 19                                       032220
18    CALL SING(2)                                                      032230
19    RETURN                                                            032240
      END                                                               032250
```

```
      SUBROUTINE SOLVE(N,NDIM,UL,IPS,B,X)                                032260
      DIMENSION UL(NDIM,NDIM),IPS(NDIM),B(NDIM),X(NDIM)                  032270
      NP1=N+1                                                            032280
      IP=IPS(1)                                                          032290
      X(1)=B(IP)                                                         032300
      DO 2 I=2,N                                                         032310
      IP=IPS(I)                                                          032320
      IM1=I-1                                                            032330
      SUM=0.0                                                            032340
      DO 1 J=1,IM1                                                       032350
    1 SUM=SUM+UL(IP,J)*X(J)                                              032360
    2 X(I)=B(IP)-SUM                                                     032370
      IP=IPS(N)                                                          032380
      X(N)=X(N)/UL(IP,N)                                                 032390
      DO 4 IBACK=2,N                                                     032400
      I=NP1-IBACK                                                        032410
      IP=IPS(I)                                                          032420
      IP1=I +1                                                           032430
      SUM=0.0                                                            032440
      DO 3 J=IP1,N                                                       032450
    3 SUM=SUM+UL(IP,J)*X(J)                                              032460
    4 X(I)=(X(I)-SUM)/UL(IP,I)                                           032470
      RETURN                                                             032480
      END                                                                032490
```

```
1          SUBROUTINE SING(IWHY)                                          032500
           NOUT=6                                                         032510
           IF(IWHY.EQ.1)WRITE(NOUT,11)                                    032520
           IF(IWHY.EQ.2)WRITE(NOUT,12)                                    032530
5       11 FORMAT(1H0,'MATRIX WITH ZERO ROW IN DECOMPOSE.')               032540
        12 FORMAT(1H0,'SING MATRIX IN DECOMPOSE.  ZERO DIVIDE IN SOLVE.') 032550
           RETURN                                                         032560
           END                                                            032570
```

APPENDIX B

Below is a listing of the job stream for the above test problem (initial gas pressure = 10 MPa). Following is a listing of the output. Since there are many variables to keep track of, the output is on multiple files. Only the ones of interest for a particular problem need to be printed out. Additional files contain the values of interest for a problem with a droplet option.

```
TERRY,STMFZ,T[illegible],P5.
ACCOUNT,P002P9.
ATTACH,TC,RLGBARS,I[illegible]=TCOFFE[illegible].
MODE(0)
TC.
DISPOSE,TAPE12,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE11,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE10,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE9,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE8,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE7,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
DISPOSE,TAPE17,UN=RJE16[illegible],RP[illegible],ST=MEAI[illegible]I.
BEGIN,SAVMEAU,FILE,LE=TAPE[illegible],PF=25RC52,UN=TCOFFE[illegible].
 COFFEE-25MM-ROUND 6-OTTOII-INSTAN-P[illegible]=1[illegible]-[illegible]=.5[illegible]
*
    1.27E+00   2.1336E+02      OFFSET   TRAVEL
    2.50E+00     9.50E-01      04    DC34
   1.944E+02                   PROJ WEIGHT
 4.30820E+02                   PISTON WEIGHT
   0                           [illegible]
  9.6000E+01   3.5395E+01      V1    V
    1.00E-03                   VP [illegible] COLLAPSING [illegible] 1
   1.300E+01    1.545E+01      A1    A3
VENT1                          VENT AREA TABLE
   2
    0.00E+00     2.49E+[illegible]
    1.00E+00     2.49E+[illegible]
   2                          PISTON RESISTANCE
    0.00E+00     0.00E+00
    1.00E+00     0.00E+00
DIS1                          DIS COFFE VS. PISTON TRAVEL
   2
    0.00E+00     5.80E-01
    1.00E+00     5.80E-01
   3                          PROJECTILE RESISTANCE
    0.00E+00     3.62E+01
    1.00E-02     1.00E-01
  2.1336E+02     1.00E-01
   1.230E+00   2.4131E+03     [illegible]      RH1    K1    K2
  3.3309E+03    1.260E+00      ENERGY   GAMMA
    3.63E+01                   SURFACE TENSION
  1.9010E+01    1.257E+00      MOL WT GAS   COVOLUME
    9.00E+00     1.00E+01      P1    P3
DROP1                          INSTAN BURNING
PRIM1                          NO PRIMER
   1
    1.00E-04                   TINC
    1.00E-04     1.00E-06      SPS    SPEC
  22   0                     MF      KWRITE
SENINFO
25LC52
;

0
0
```

Reproduced from best available copy.

```
COFFEE-25MM-ROUND 6-OTTOII-INSTAN-P3=10-DC=.58

  OFFSET =  1.2700E+00      TRAVEL =  2.1336E+02

  TUBE DIAM =  2.5000E+00      TUBE AREA =  4.9087E+00

  TUBE ENTRANCE COEFF =  9.5000E-01

  PJWT =   1.9440E+02      PSWT =  4.8082E+02

  MBACK =   0      NO BACK FLOW

  V1 IN =  9.6000E+01      V3 IN =  3.5396E+01      VOL =  1.3140E+02

  VPER =  1.0000E-03

  AREA FUEL =  1.3000E+01      AREA  COMB =  1.5480E+01

  VENT1                          VENT AREA TABLE

  NVENT =   2

  PIS TRAV =  0.              VENT AREA =  2.4900E+00      HYDROLIC DIFF =  1.2360E+00
  PIS TRAV =  1.0000E+00      VENT AREA =  2.4900E+00      HYDROLIC DIFF =  1.2360E+00

  MAX PISTON TRAVEL =  7.3846E+00

  BORE =  4.4396E+00      BORE/STROKE =  6.0119E-01

  COMBUSTOR LENGTH =  3.4502E+01

  NPIS =   2

  FRAC PIS TRAV =  0.              PIS TRAV =  0.              PIS RESISTANCE =  0.
  FRAC PIS TRAV =  1.0000E+00      PIS TRAV =  7.3846E+00      PIS RESISTANCE =  0.

  DIS1                           DIS COEFF VS. PISTON TRAVEL

  NDC =   2

  FRAC PIS TRAV =  0.              PIS TRAV =  0.              DIS COEFF =  5.8000E-01
  FRAC PIS TRAV =  1.0000E+00      PIS TRAV =  7.3846E+00      DIS COEFF =  5.8000E-01

  NPROJ =   3

  TRAVEL =  0.              RESISTIVE PRESS =  3.6200E+01
  TRAVEL =  1.0000E-02      RESISTIVE PRESS =  1.0000E-01
  TRAVEL =  2.1336E+02      RESISTIVE PRESS =  1.0000E-01

  DENS LIQUID =  1.2300E+00      K1 =  2.4131E+03      K2 =  0.

  CHEM ENERGY =  3.3309E+03      GAM =  1.2500E+00

  SURFACE TENSION =  3.6300E+01

  MOL WT GAS =  1.9010E+01      COVOLUME =  1.2570E+00

  PRES LIQUID =  9.0000E+00      PRES GAS =  1.0000E+01
```

```
SPECIFIC GAS CONSTANT =  4.3756E-01
CV =  1.6829E+00      CP =  2.1205E+00

DROP1                          INSTAN BURNING

PRIM1                          NO PRIMER

NPRIM =   1

TIME =  1.0000E-04
MF =  22      EPS =  1.0000E-04      SPEC =  1.0000E-06
CHARGE =  1.1852E+02      PRIMER =  4.7382E-01      C/M =  6.1211E-01
LOADING DENSITY =  9.0562E-01
```

T (MS)	P1	P3	PL	P4	PR	S PS	V PS	S PJ	V PJ	ACC K-G
0.000	9.000	10.000	10.000	10.000	10.000	0.000	0.000	0.000	0.000	0.000
.100	10.061	10.056	10.055	10.055	10.055	.003	64.957	0.000	0.000	0.000
.200	11.587	11.569	11.567	11.567	11.567	.013	120.111	0.000	0.000	0.000
.300	14.104	14.062	14.058	14.058	14.058	.029	185.139	0.000	0.000	0.000
.400	17.818	17.732	17.725	17.725	17.725	.050	265.226	0.000	0.000	0.000
.500	23.031	22.864	22.854	22.854	22.854	.082	366.537	0.000	0.000	0.000
.600	30.157	29.847	29.834	29.834	29.834	.124	496.527	0.000	0.000	0.000
.700	39.754	39.188	39.171	39.170	39.169	.182	664.130	.000	11.356	.862
.800	52.490	51.450	51.418	51.405	51.378	.259	879.583	.020	618.044	13.193
.900	68.844	66.882	66.791	66.767	66.718	.360	1151.575	.152	2099.807	17.139
1.000	89.230	85.576	85.334	85.286	85.190	.491	1483.680	.454	4008.717	21.892
1.100	113.874	107.269	106.685	106.587	106.391	.658	1870.332	.971	6419.787	27.346
1.200	142.580	131.197	129.925	129.728	129.336	.866	2291.759	1.757	9390.959	33.250
1.300	174.500	156.107	153.622	153.250	152.504	1.117	2711.592	2.869	12947.889	39.210
1.400	207.989	180.446	176.078	175.417	174.096	1.407	3082.054	4.365	17073.857	44.765
1.500	240.595	202.609	195.642	194.556	192.383	1.730	3357.041	6.300	21707.026	49.470
1.600	269.353	221.159	211.009	209.354	206.045	2.074	3508.652	8.719	26746.263	52.985
1.700	291.449	234.992	221.384	219.038	214.346	2.428	3538.719	11.658	32063.738	55.121
1.800	305.121	243.489	226.541	223.428	217.201	2.779	3477.485	15.137	37521.750	55.955
1.900	310.189	246.637	226.812	222.916	215.125	3.122	3368.925	19.163	42987.796	55.321
2.000	308.012	245.070	222.986	218.352	209.083	3.453	3251.968	23.731	48348.681	53.767
2.100	300.742	239.875	216.110	210.825	200.254	3.773	3148.899	28.826	53518.995	51.495
2.200	290.497	232.286	207.327	201.497	189.837	4.083	3065.765	34.427	58443.631	48.815
2.300	278.859	223.419	197.770	191.494	178.942	4.386	2999.678	40.506	63096.886	46.012
2.400	266.929	214.178	188.311	181.665	168.373	4.684	2946.237	47.037	67477.579	43.293
2.500	255.456	205.167	179.133	172.190	158.303	4.976	2901.867	53.993	71597.777	40.702
2.600	244.627	196.442	169.950	162.801	148.504	5.264	2862.020	61.349	75466.806	38.181
2.700	234.225	188.080	161.556	154.241	139.612	5.548	2822.498	69.079	79100.247	35.893
2.800	224.537	180.288	153.780	146.341	131.465	5.829	2784.503	77.162	82518.623	33.797
2.900	215.578	173.063	146.627	139.097	124.038	6.105	2747.540	85.576	85740.410	31.886
3.000	207.296	166.392	140.063	132.468	117.277	6.378	2711.329	94.304	88783.075	30.147
3.100	199.647	160.241	134.046	126.405	111.124	6.648	2675.960	103.328	91662.732	28.564
3.200	192.629	154.576	128.528	120.859	105.520	6.913	2641.261	112.632	94394.026	27.122
3.300	186.096	149.347	123.463	115.778	100.407	7.176	2607.606	122.203	96990.121	25.807
3.400	0.000	137.858	118.649	110.973	95.620	7.380	0.000	132.026	99462.692	24.575
3.500	0.000	115.898	110.948	103.609	88.930	7.380	0.000	142.091	101793.875	22.854
3.600	0.000	104.673	102.248	95.404	81.716	7.380	0.000	152.379	103944.681	20.998
3.700	0.000	96.228	94.286	87.917	75.179	7.380	0.000	162.874	105921.085	19.316
3.800	0.000	88.944	87.226	81.286	69.406	7.380	0.000	173.558	107742.458	17.831
3.900	0.000	82.497	80.966	75.412	64.304	7.380	0.000	184.418	109426.765	16.518
4.000	0.000	76.757	75.395	70.188	59.776	7.380	0.000	195.440	110989.640	15.353
4.100	0.000	71.629	70.415	65.523	55.739	7.380	0.000	206.612	112444.407	14.315
4.150	0.000	68.813	67.685	62.959	53.509	7.380	0.000	213.384	113270.409	13.767

```
MUZZLE VEL (M/SEC)        1132.7

MAX P1 (MPA)               310.2

MAX P3 (MPA)               246.6

MAX PL (MPA)               226.8

MAX PR (MPA)               217.2

MAX ACC (K-G)               55.9

MAX MASS ERROR               .07

MAX ENERGY ERROR            2.20

RUN TIME =          2.5     NSTEP =     689
13.19.11.UCLP, FA, TM1602C,      0.501KLNS.
```

T (MS)	RH1	RHL3	RHG3	RH3	RHL4	RHG4	RH4
0.0000	1.2346	0.0000	0.0000	.0114	0.0000	0.0000	.0114
.1000	1.2351	0.0000	0.0000	.0115	0.0000	0.0000	.0114
.2000	1.2359	0.0000	0.0000	.0132	0.0000	0.0000	.0128
.3000	1.2372	0.0000	0.0000	.0161	0.0000	0.0000	.0150
.4000	1.2391	0.0000	0.0000	.0203	0.0000	0.0000	.0183
.5000	1.2418	0.0000	0.0000	.0261	0.0000	0.0000	.0228
.6000	1.2455	0.0000	0.0000	.0339	0.0000	0.0000	.0289
.7000	1.2504	0.0000	0.0000	.0441	0.0000	0.0000	.0369
.8000	1.2570	0.0000	0.0000	.0571	0.0000	0.0000	.0473
.9000	1.2656	0.0000	0.0000	.0731	0.0000	0.0000	.0609
1.0000	1.2763	0.0000	0.0000	.0920	0.0000	0.0000	.0780
1.1000	1.2894	0.0000	0.0000	.1134	0.0000	0.0000	.0978
1.2000	1.3049	0.0000	0.0000	.1364	0.0000	0.0000	.1191
1.3000	1.3222	0.0000	0.0000	.1595	0.0000	0.0000	.1404
1.4000	1.3407	0.0000	0.0000	.1816	0.0000	0.0000	.1601
1.5000	1.3590	0.0000	0.0000	.2012	0.0000	0.0000	.1773
1.6000	1.3752	0.0000	0.0000	.2176	0.0000	0.0000	.1911
1.7000	1.3879	0.0000	0.0000	.2301	0.0000	0.0000	.2013
1.8000	1.3958	0.0000	0.0000	.2386	0.0000	0.0000	.2077
1.9000	1.3987	0.0000	0.0000	.2432	0.0000	0.0000	.2107
2.0000	1.3975	0.0000	0.0000	.2445	0.0000	0.0000	.2108
2.1000	1.3933	0.0000	0.0000	.2429	0.0000	0.0000	.2085
2.2000	1.3874	0.0000	0.0000	.2394	0.0000	0.0000	.2045
2.3000	1.3807	0.0000	0.0000	.2345	0.0000	0.0000	.1996
2.4000	1.3739	0.0000	0.0000	.2289	0.0000	0.0000	.1943
2.5000	1.3674	0.0000	0.0000	.2229	0.0000	0.0000	.1887
2.6000	1.3612	0.0000	0.0000	.2168	0.0000	0.0000	.1828
2.7000	1.3554	0.0000	0.0000	.2107	0.0000	0.0000	.1771
2.8000	1.3499	0.0000	0.0000	.2046	0.0000	0.0000	.1715
2.9000	1.3449	0.0000	0.0000	.1988	0.0000	0.0000	.1662
3.0000	1.3403	0.0000	0.0000	.1932	0.0000	0.0000	.1612
3.1000	1.3361	0.0000	0.0000	.1879	0.0000	0.0000	.1563
3.2000	1.3322	0.0000	0.0000	.1828	0.0000	0.0000	.1518
3.3000	1.3286	0.0000	0.0000	.1780	0.0000	0.0000	.1475
3.4000	1.3260	0.0000	0.0000	.1687	0.0000	0.0000	.1433
3.5000	1.3260	0.0000	0.0000	.1511	0.0000	0.0000	.1369
3.6000	1.3260	0.0000	0.0000	.1415	0.0000	0.0000	.1297
3.7000	1.3260	0.0000	0.0000	.1339	0.0000	0.0000	.1228
3.8000	1.3260	0.0000	0.0000	.1271	0.0000	0.0000	.1165
3.9000	1.3260	0.0000	0.0000	.1208	0.0000	0.0000	.1107
4.0000	1.3260	0.0000	0.0000	.1151	0.0000	0.0000	.1054
4.1000	1.3260	0.0000	0.0000	.1098	0.0000	0.0000	.1005
4.1500	1.3260	0.0000	0.0000	.1068	0.0000	0.0000	.0978

13.14.54.UCLP. FA. TM1602C. 0.233KL IS.

T (MS)	V1	VL3	VG3	V3	VL4	VG4	V4	EPS3	EPS4	AVENT
0.000	96.000	0.000	0.000	35.396	0.000	0.000	6.234	0.000	0.000	2.490
.100	95.955	0.000	0.000	35.450	0.000	0.000	6.234	0.000	0.000	2.490
.200	95.836	0.000	0.000	35.592	0.000	0.000	6.234	0.000	0.000	2.490
.300	95.638	0.000	0.000	35.827	0.000	0.000	6.234	0.000	0.000	2.490
.400	95.348	0.000	0.000	36.173	0.000	0.000	6.234	0.000	0.000	2.490
.500	94.940	0.000	0.000	36.659	0.000	0.000	6.234	0.000	0.000	2.490
.600	94.382	0.000	0.000	37.323	0.000	0.000	6.234	0.000	0.000	2.490
.700	93.632	0.000	0.000	38.215	0.000	0.000	6.235	0.000	0.000	2.490
.800	92.635	0.000	0.000	39.403	0.000	0.000	6.330	0.000	0.000	2.490
.900	91.321	0.000	0.000	40.968	0.000	0.000	6.981	0.000	0.000	2.490
1.000	89.614	0.000	0.000	43.000	0.000	0.000	8.462	0.000	0.000	2.490
1.100	87.440	0.000	0.000	45.589	0.000	0.000	11.000	0.000	0.000	2.490
1.200	84.737	0.000	0.000	48.808	0.000	0.000	14.857	0.000	0.000	2.490
1.300	81.482	0.000	0.000	52.683	0.000	0.000	20.315	0.000	0.000	2.490
1.400	77.709	0.000	0.000	57.176	0.000	0.000	27.660	0.000	0.000	2.490
1.500	73.511	0.000	0.000	62.176	0.000	0.000	37.158	0.000	0.000	2.490
1.600	69.032	0.000	0.000	67.508	0.000	0.000	49.035	0.000	0.000	2.490
1.700	64.438	0.000	0.000	72.980	0.000	0.000	63.460	0.000	0.000	2.490
1.800	59.867	0.000	0.000	78.421	0.000	0.000	80.536	0.000	0.000	2.490
1.900	55.414	0.000	0.000	83.725	0.000	0.000	100.299	0.000	0.000	2.490
2.000	51.112	0.000	0.000	88.848	0.000	0.000	122.723	0.000	0.000	2.490
2.100	46.955	0.000	0.000	93.798	0.000	0.000	147.735	0.000	0.000	2.490
2.200	42.918	0.000	0.000	98.604	0.000	0.000	175.226	0.000	0.000	2.490
2.300	38.978	0.000	0.000	103.296	0.000	0.000	205.069	0.000	0.000	2.490
2.400	35.114	0.000	0.000	107.897	0.000	0.000	237.128	0.000	0.000	2.490
2.500	31.313	0.000	0.000	112.423	0.000	0.000	271.273	0.000	0.000	2.490
2.600	27.567	0.000	0.000	116.884	0.000	0.000	307.378	0.000	0.000	2.490
2.700	23.872	0.000	0.000	121.283	0.000	0.000	345.324	0.000	0.000	2.490
2.800	20.228	0.000	0.000	125.623	0.000	0.000	385.000	0.000	0.000	2.490
2.900	16.632	0.000	0.000	129.905	0.000	0.000	426.305	0.000	0.000	2.490
3.000	13.084	0.000	0.000	134.130	0.000	0.000	469.148	0.000	0.000	2.490
3.100	9.582	0.000	0.000	138.299	0.000	0.000	513.443	0.000	0.000	2.490
3.200	6.126	0.000	0.000	142.415	0.000	0.000	559.116	0.000	0.000	2.490
3.300	2.715	0.000	0.000	146.477	0.000	0.000	606.095	0.000	0.000	2.490
3.400	.055	0.000	0.000	149.644	0.000	0.000	654.317	0.000	0.000	2.490
3.500	.055	0.000	0.000	149.644	0.000	0.000	703.720	0.000	0.000	2.490
3.600	.055	0.000	0.000	149.644	0.000	0.000	754.224	0.000	0.000	2.490
3.700	.055	0.000	0.000	149.644	0.000	0.000	805.739	0.000	0.000	2.490
3.800	.055	0.000	0.000	149.644	0.000	0.000	858.186	0.000	0.000	2.490
3.900	.055	0.000	0.000	149.644	0.000	0.000	911.493	0.000	0.000	2.490
4.000	.055	0.000	0.000	149.644	0.000	0.000	965.596	0.000	0.000	2.490
4.100	.055	0.000	0.000	149.644	0.000	0.000	1020.440	0.000	0.000	2.490
4.140	.055	0.000	0.000	149.644	0.000	0.000	1053.681	0.000	0.000	2.490

13.15.25.UCLP, FA. TM1602C, 0.233KLNS.

T (MS)	RM1	RML3	RMG3	RM3	RML4	RMG4	RM4	RMSUM
0.000	118.521	0.000	0.000	.403	0.000	0.000	.071	118.995
.100	118.518	0.000	0.000	.406	0.000	0.000	.071	118.995
.200	118.445	0.000	0.000	.470	0.000	0.000	.080	118.995
.300	118.325	0.000	0.000	.577	0.000	0.000	.094	118.995
.400	118.147	0.000	0.000	.734	0.000	0.000	.114	118.995
.500	117.896	0.000	0.000	.957	0.000	0.000	.142	118.995
.600	117.550	0.000	0.000	1.265	0.000	0.000	.180	118.995
.700	117.081	0.000	0.000	1.684	0.000	0.000	.230	118.995
.800	116.446	0.000	0.000	2.249	0.000	0.000	.300	118.995
.900	115.575	0.000	0.000	2.994	0.000	0.000	.425	118.995
1.000	114.378	0.000	0.000	3.957	0.000	0.000	.660	118.995
1.100	112.748	0.000	0.000	5.171	0.000	0.000	1.076	118.994
1.200	110.570	0.000	0.000	6.655	0.000	0.000	1.770	118.995
1.300	107.739	0.000	0.000	8.405	0.000	0.000	2.851	118.996
1.400	104.186	0.000	0.000	10.382	0.000	0.000	4.429	118.997
1.500	99.898	0.000	0.000	12.511	0.000	0.000	6.588	118.997
1.600	94.937	0.000	0.000	14.687	0.000	0.000	9.372	118.996
1.700	89.433	0.000	0.000	16.791	0.000	0.000	12.772	118.995
1.800	83.562	0.000	0.000	18.710	0.000	0.000	16.726	118.998
1.900	77.509	0.000	0.000	20.364	0.000	0.000	21.132	119.005
2.000	71.427	0.000	0.000	21.719	0.000	0.000	25.866	119.011
2.100	65.420	0.000	0.000	22.787	0.000	0.000	30.801	119.008
2.200	59.543	0.000	0.000	23.607	0.000	0.000	35.840	118.990
2.300	53.816	0.000	0.000	24.226	0.000	0.000	40.933	118.974
2.400	48.242	0.000	0.000	24.697	0.000	0.000	46.068	119.006
2.500	42.817	0.000	0.000	25.064	0.000	0.000	51.192	119.073
2.600	37.525	0.000	0.000	25.344	0.000	0.000	56.187	119.056
2.700	32.356	0.000	0.000	25.549	0.000	0.000	61.152	119.057
2.800	27.307	0.000	0.000	25.705	0.000	0.000	66.045	119.056
2.900	22.369	0.000	0.000	25.823	0.000	0.000	70.863	119.055
3.000	17.537	0.000	0.000	25.913	0.000	0.000	75.604	119.054
3.100	12.803	0.000	0.000	25.981	0.000	0.000	80.269	119.053
3.200	8.162	0.000	0.000	26.035	0.000	0.000	84.856	119.053
3.300	3.607	0.000	0.000	26.075	0.000	0.000	89.370	119.052
3.400	.073	0.000	0.000	25.240	0.000	0.000	93.739	119.052
3.500	.073	0.000	0.000	22.610	0.000	0.000	96.369	119.053
3.600	.073	0.000	0.000	21.171	0.000	0.000	97.810	119.055
3.700	.073	0.000	0.000	20.037	0.000	0.000	98.948	119.057
3.800	.073	0.000	0.000	19.017	0.000	0.000	99.967	119.058
3.900	.073	0.000	0.000	18.082	0.000	0.000	100.902	119.058
4.000	.073	0.000	0.000	17.222	0.000	0.000	101.763	119.059
4.100	.073	0.000	0.000	16.429	0.000	0.000	102.557	119.059
4.160	.073	0.000	0.000	15.993	0.000	0.000	103.004	119.061

13.16.56.UCLP. F4. TM1602C. 0.233KLNS.

T (MS)	RMD13	RMD3	RMD34	RMD4	HL1	HL3	H3	HL4	H4
0.000	0.000	0.000	0.000	0.000	3338.190	0.000	4209.504	0.000	4209.504
.100	513.707	0.000	59.279	0.000	3339.046	0.000	4207.164	0.000	4214.330
.200	951.249	0.000	109.784	0.000	3340.275	0.000	4187.497	0.000	4330.796
.300	1470.662	0.000	168.843	0.000	3342.300	0.000	4165.684	0.000	4473.918
.400	2116.147	0.000	240.351	0.000	3345.290	0.000	4146.583	0.000	4617.923
.500	2940.291	0.000	328.263	0.000	3349.446	0.000	4132.393	0.000	4747.532
.600	4011.742	0.000	436.563	0.000	3355.113	0.000	4123.586	0.000	4858.119
.700	5431.907	0.000	570.812	0.000	3362.692	0.000	4120.000	0.000	4950.690
.800	7385.689	0.000	895.947	0.000	3372.657	0.000	4119.348	0.000	5013.826
.900	10178.117	0.000	1699.733	0.000	3385.297	0.000	4111.269	0.000	4986.929
1.000	13948.401	0.000	3109.937	0.000	3400.811	0.000	4092.874	0.000	4887.967
1.100	18848.774	0.000	5368.770	0.000	3419.213	0.000	4064.645	0.000	4766.108
1.200	24890.881	0.000	8688.087	0.000	3440.168	0.000	4028.492	0.000	4651.139
1.300	31850.690	0.000	13130.572	0.000	3462.873	0.000	3987.185	0.000	4550.359
1.400	39247.295	0.000	18573.790	0.000	3486.032	0.000	3943.762	0.000	4460.526
1.500	46404.063	0.000	24692.860	0.000	3507.944	0.000	3900.227	0.000	4376.986
1.600	52581.336	0.000	30989.896	0.000	3526.758	0.000	3857.079	0.000	4296.146
1.700	57171.732	0.000	36901.371	0.000	3540.893	0.000	3813.574	0.000	4215.415
1.800	59903.786	0.000	41935.720	0.000	3549.501	0.000	3768.680	0.000	4133.333
1.900	60893.478	0.000	45794.958	0.000	3552.665	0.000	3721.838	0.000	4049.557
2.000	60573.637	0.000	48455.109	0.000	3551.308	0.000	3673.575	0.000	3964.957
2.100	59477.172	0.000	50110.169	0.000	3546.756	0.000	3625.310	0.000	3881.129
2.200	58041.599	0.000	50978.294	0.000	3540.290	0.000	3578.960	0.000	3799.958
2.300	56507.129	0.000	51148.883	0.000	3532.873	0.000	3536.430	0.000	3723.435
2.400	54983.352	0.000	50745.310	0.000	3525.190	0.000	3499.212	0.000	3653.374
2.500	53557.822	0.000	50243.243	0.000	3517.725	0.000	3467.744	0.000	3589.466
2.600	52307.341	0.000	49983.994	0.000	3510.610	0.000	3440.707	0.000	3529.055
2.700	51078.006	0.000	49297.065	0.000	3503.712	0.000	3417.504	0.000	3475.289
2.800	49917.501	0.000	48570.424	0.000	3497.230	0.000	3398.276	0.000	3426.649
2.900	48839.252	0.000	47807.717	0.000	3491.188	0.000	3382.408	0.000	3382.747
3.000	47822.374	0.000	47034.921	0.000	3485.560	0.000	3369.434	0.000	3343.110
3.100	46864.705	0.000	46263.553	0.000	3480.326	0.000	3358.878	0.000	3307.293
3.200	45985.871	0.000	45508.192	0.000	3475.493	0.000	3350.377	0.000	3274.850
3.300	45129.986	0.000	44766.810	0.000	3470.968	0.000	3343.525	0.000	3245.422
3.400	0.000	0.000	37537.645	0.000	3330.900	0.000	3294.504	0.000	3217.366
3.500	0.000	0.000	18035.815	0.000	3330.900	0.000	3156.951	0.000	3165.638
3.600	0.000	0.000	12215.956	0.000	3330.900	0.000	3079.401	0.000	3103.924
3.700	0.000	0.000	10632.194	0.000	3330.900	0.000	3017.613	0.000	3044.410
3.800	0.000	0.000	9745.169	0.000	3330.900	0.000	2961.726	0.000	2988.718
3.900	0.000	0.000	8967.546	0.000	3330.900	0.000	2909.699	0.000	2936.755
4.000	0.000	0.000	8257.115	0.000	3330.900	0.000	2861.023	0.000	2888.183
4.100	0.000	0.000	7613.606	0.000	3330.900	0.000	2815.435	0.000	2842.676
4.160	0.000	0.000	7238.210	0.000	3330.900	0.000	2789.487	0.000	2816.755

13.15.55.UCLP. FA. TM1602C. 0.233KLNS.

T (MS)	CL1	CL3	CG3	C3	CL4	CG4	C4	TEMP3	TEMP4
0.000	139805.811	0.000	0.000	105976.943	0.000	0.000	105976.843	1979.239	1979.239
.100	139775.075	0.000	0.000	105955.772	0.000	0.000	106043.541	1978.102	1981.492
.200	139730.895	0.000	0.000	105920.961	0.000	0.000	107664.583	1967.930	2035.510
.300	139658.035	0.000	0.000	105994.607	0.000	0.000	109707.923	1956.165	2101.528
.400	139550.592	0.000	0.000	106264.745	0.000	0.000	111877.054	1944.982	2167.219
.500	139399.952	0.000	0.000	106799.032	0.000	0.000	114032.949	1935.268	2225.348
.600	139194.262	0.000	0.000	107658.104	0.000	0.000	116174.282	1926.955	2273.362
.700	138917.757	0.000	0.000	108911.217	0.000	0.000	118383.912	1919.727	2311.494
.800	138551.622	0.000	0.000	110611.467	0.000	0.000	120614.139	1912.151	2334.006
.900	138082.924	0.000	0.000	112678.577	0.000	0.000	122272.072	1899.193	2312.215
1.000	137500.883	0.000	0.000	115107.843	0.000	0.000	123605.529	1879.436	2254.567
1.100	136800.541	0.000	0.000	117889.329	0.000	0.000	125125.704	1853.264	2184.473
1.200	135989.296	0.000	0.000	120958.631	0.000	0.000	127043.681	1822.030	2116.536
1.300	135092.897	0.000	0.000	124183.744	0.000	0.000	129245.689	1787.784	2055.066
1.400	134158.821	0.000	0.000	127377.847	0.000	0.000	131456.590	1752.878	1999.561
1.500	133255.570	0.000	0.000	130317.907	0.000	0.000	133382.420	1719.209	1948.819
1.600	132463.903	0.000	0.000	132776.522	0.000	0.000	134780.847	1687.865	1901.923
1.700	131858.696	0.000	0.000	134556.080	0.000	0.000	135489.404	1659.148	1858.110
1.800	131485.482	0.000	0.000	135527.065	0.000	0.000	135440.125	1632.939	1816.799
1.900	131347.366	0.000	0.000	135659.382	0.000	0.000	134660.343	1608.983	1777.594
2.000	131406.679	0.000	0.000	135037.117	0.000	0.000	133260.313	1587.152	1740.403
2.100	131604.921	0.000	0.000	133829.922	0.000	0.000	131399.975	1567.470	1705.333
2.200	131884.679	0.000	0.000	132241.686	0.000	0.000	129261.521	1550.110	1672.582
2.300	132203.064	0.000	0.000	130464.196	0.000	0.000	127030.659	1535.310	1642.425
2.400	132530.219	0.000	0.000	128656.838	0.000	0.000	124856.696	1523.236	1615.211
2.500	132845.612	0.000	0.000	126924.264	0.000	0.000	122769.219	1513.738	1590.689
2.600	133144.005	0.000	0.000	125268.363	0.000	0.000	120704.758	1506.159	1567.765
2.700	133431.285	0.000	0.000	123698.864	0.000	0.000	118814.777	1500.174	1547.484
2.800	133699.403	0.000	0.000	122253.510	0.000	0.000	117063.109	1495.725	1529.229
2.900	133947.811	0.000	0.000	120928.486	0.000	0.000	115449.291	1492.525	1512.820
3.000	134177.883	0.000	0.000	119719.078	0.000	0.000	113965.858	1490.361	1498.057
3.100	134390.683	0.000	0.000	118616.290	0.000	0.000	112603.394	1489.029	1484.755
3.200	134586.287	0.000	0.000	117612.131	0.000	0.000	111351.948	1488.379	1472.747
3.300	134768.589	0.000	0.000	116695.066	0.000	0.000	110201.338	1488.247	1461.982
3.400	134902.034	0.000	0.000	114321.723	0.000	0.000	109104.143	1471.939	1451.499
3.500	134902.034	0.000	0.000	109228.628	0.000	0.000	107308.911	1420.089	1431.470
3.600	134902.034	0.000	0.000	106482.901	0.000	0.000	105236.580	1390.170	1407.230
3.700	134902.034	0.000	0.000	104344.945	0.000	0.000	103282.245	1366.038	1383.602
3.800	134902.034	0.000	0.000	102445.327	0.000	0.000	101492.375	1343.999	1361.259
3.900	134902.034	0.000	0.000	100711.465	0.000	0.000	99854.671	1323.286	1340.245
4.000	134902.034	0.000	0.000	99120.137	0.000	0.000	98351.195	1303.733	1320.435
4.100	134902.034	0.000	0.000	97655.510	0.000	0.000	96965.956	1285.274	1301.741
4.140	134902.034	0.000	0.000	96832.455	0.000	0.000	96186.412	1274.707	1291.036

13.14.26.UCLP. FA. TM1602C. 0.233KLNS.

T (MS)	EL1	EL3	EG3	EL4	EG4	EKPS	EKPJ	EKL4	EKG4	ESUM
0.000	394782.327	0.000	1341.908	0.000	236.343	0.000	0.000	0.000	0.000	396360.577
.100	394771.089	0.000	1351.345	0.000	237.635	.101	0.000	0.000	0.000	396360.171
.200	394528.838	0.000	1557.435	0.000	272.895	.347	0.000	0.000	0.000	396359.514
.300	394128.216	0.000	1898.474	0.000	330.718	.824	0.000	0.000	0.000	396358.232
.400	393534.969	0.000	2403.985	0.000	415.245	1.591	0.000	0.000	0.000	396355.890
.500	392698.326	0.000	3117.870	0.000	532.282	3.230	0.000	0.000	0.000	396351.707
.600	391547.039	0.000	4101.997	0.000	689.358	5.927	0.000	0.000	0.000	396344.321
.700	389984.188	0.000	5440.857	0.000	895.671	10.604	.001	0.000	.000	396331.321
.800	387870.686	0.000	7237.871	0.000	1177.054	18.600	3.713	0.000	.002	396307.926
.900	384969.821	0.000	9570.308	0.000	1655.458	31.981	42.857	0.000	.031	396270.357
1.000	380981.179	0.000	12515.754	0.000	2503.642	52.922	156.199	0.000	.177	396209.871
1.100	375551.547	0.000	16127.343	0.000	3955.065	84.099	400.597	0.000	.739	396119.389
1.200	368297.533	0.000	20407.393	0.000	6303.043	126.267	857.208	0.000	2.601	395994.044
1.300	358868.756	0.000	25288.139	0.000	9861.489	176.767	1629.537	0.000	7.967	395832.654
1.400	347031.901	0.000	30625.095	0.000	14905.287	228.347	2833.541	0.000	21.521	395645.712
1.500	332749.149	0.000	36196.693	0.000	21608.101	270.935	4580.015	0.000	51.741	395456.634
1.600	316224.005	0.000	41719.406	0.000	29997.793	295.960	6953.324	0.000	111.740	395302.228
1.700	297891.877	0.000	46883.908	0.000	39937.649	301.054	9992.970	0.000	218.939	395226.297
1.800	278337.900	0.000	51416.641	0.000	51140.407	290.725	13684.610	0.000	392.472	395262.755
1.900	258174.265	0.000	55140.219	0.000	63218.396	272.857	17962.080	0.000	650.859	395418.675
2.000	237914.997	0.000	58012.622	0.000	75759.436	254.241	22721.423	0.000	1007.724	395670.443
2.100	217906.910	0.000	60111.005	0.000	88398.256	238.380	27840.229	0.000	1470.401	395965.781
2.200	198330.896	0.000	61582.975	0.000	100884.173	225.959	33200.196	0.000	2040.306	396264.505
2.300	179254.799	0.000	62595.229	0.000	113140.876	216.323	38697.430	0.000	2716.034	396620.491
2.400	160689.735	0.000	63309.106	0.000	125223.967	208.683	44257.333	0.000	3495.928	397184.752
2.500	142617.911	0.000	63851.570	0.000	137039.903	202.445	49827.069	0.000	4373.670	397912.569
2.600	124993.008	0.000	64241.222	0.000	148244.160	196.924	55357.722	0.000	5333.273	398366.309
2.700	107774.704	0.000	64503.011	0.000	159257.470	191.523	60816.573	0.000	6376.973	398920.254
2.800	90955.717	0.000	64703.936	0.000	169970.815	186.401	66186.620	0.000	7495.335	399498.824
2.900	74510.370	0.000	64862.054	0.000	180414.078	181.485	71455.783	0.000	8682.398	400106.168
3.000	58414.263	0.000	64993.496	0.000	190607.117	176.733	76617.262	0.000	9932.452	400741.323
3.100	42645.722	0.000	65107.381	0.000	200568.960	172.152	81667.988	0.000	11240.350	401402.553
3.200	27185.454	0.000	65212.487	0.000	210318.113	167.716	86607.456	0.000	12601.514	402092.740
3.300	12013.561	0.000	65310.288	0.000	219869.851	163.469	91436.851	0.000	14011.795	402805.817
3.400	244.517	0.000	62572.538	0.000	228982.311	0.000	96158.280	0.000	15455.778	403363.425
3.500	244.517	0.000	54036.401	0.000	232157.184	0.000	100718.572	0.000	16642.880	403799.555
3.600	244.517	0.000	49531.101	0.000	231638.999	0.000	105019.708	0.000	17613.126	404047.451
3.700	244.517	0.000	46062.684	0.000	230398.402	0.000	109051.366	0.000	18501.995	404258.965
3.800	244.517	0.000	43014.394	0.000	229015.992	0.000	112834.009	0.000	19341.091	404450.005
3.900	244.517	0.000	40269.223	0.000	227588.005	0.000	116389.389	0.000	20137.130	404628.264
4.000	244.517	0.000	37786.790	0.000	226136.455	0.000	119737.767	0.000	20893.133	404798.664
4.100	244.517	0.000	35536.720	0.000	224673.521	0.000	122897.198	0.000	21611.674	404963.630
4.150	244.517	0.000	34287.953	0.000	223797.813	0.000	124709.403	0.000	22026.001	405065.686

13.17.30.UCLP. FA. TM1602C. 0.233KLNS.

T (MS)	U13	WE	OC
0.000	0.000	0.000	.580
.100	167.032	0.000	.580
.200	309.104	0.000	.580
.300	477.387	0.000	.580
.400	685.858	0.000	.580
.500	950.913	0.000	.580
.600	1293.603	0.000	.580
.700	1744.589	0.000	.580
.800	2359.606	0.000	.580
.900	3229.778	0.000	.580
1.000	4388.951	0.000	.580
1.100	5870.622	0.000	.580
1.200	7660.816	0.000	.580
1.300	9674.070	0.000	.580
1.400	11756.378	0.000	.580
1.500	13713.619	0.000	.580
1.600	15355.079	0.000	.580
1.700	16543.380	0.000	.580
1.800	17235.949	0.000	.580
1.900	17483.922	0.000	.580
2.000	17407.799	0.000	.580
2.100	17144.306	0.000	.580
2.200	16801.706	0.000	.580
2.300	16436.586	0.000	.580
2.400	16072.609	0.000	.580
2.500	15730.505	0.000	.580
2.600	15432.319	0.000	.580
2.700	15134.727	0.000	.580
2.800	14850.364	0.000	.580
2.900	14583.628	0.000	.580
3.000	14329.081	0.000	.580
3.100	14086.709	0.000	.580
3.200	13862.813	0.000	.580
3.300	13641.681	0.000	.580
3.400	0.000	0.000	.580
3.500	0.000	0.000	.580
3.600	0.000	0.000	.580
3.700	0.000	0.000	.580
3.800	0.000	0.000	.580
3.900	0.000	0.000	.580
4.000	0.000	0.000	.580
4.100	0.000	0.000	.580
4.160	0.000	0.000	.580

13.18.02.UCLP. FA. TM1602C. 0.233KLNS.

GLOSSARY

A_1 Cross section area of the liquid reservoir, cm^2.

A_3 Cross section area of the combustion chamber, cm^2.

A_4 Cross section area of the gun tube, cm^2.

A_v Vent area, cm^2.

b Covolume, cm^3/gm.

c_1 Speed of sound in the liquid in the reservoir, cm/s.

c_{L3} Speed of sound in the liquid in the combustion chamber, cm/s.

c_{L4} Speed of sound in the liquid in the gun tube, cm/s.

c_{G3} Speed of sound in the gas in the combustion chamber, cm/s.

c_{G4} Speed of sound in the gas in the gun tube, cm/s.

c_3 Average speed of sound in the combustion chamber, cm/s.

c_4 Average speed of sound in the gun tube, cm/s.

c_p Specific heat at constant pressure, joules/gm-K.

c_v Specific heat at constant volume, joules/gm-K.

C_D Discharge coefficient for the piston.

C_D' Discharge coefficient for the gun tube.

e_1 Chemical energy of the liquid, joules/gm.

e_3	Internal energy of the gas in the combustion chamber, joules/gm.
e_4	Internal energy of the gas in the gun tube, joules/gm.
E_1	Total internal energy of the liquid, joules.
E_3	Total internal energy in the gas in the chamber, joules.
E_4	Total internal energy in the gas in the gun tube, joules.
EK_{G4}	Kinetic energy of the gas in the gun tube, joules.
EK_{L4}	Kinetic energy of the liquid in the gun tube, joules.
EK_{pj}	Kinetic energy of the projectile, joules.
EK_{ps}	Kinetic energy of the piston, joules.
g_o	Conversion constant = 10^7 gm/s-cm-MPa.
h_1	Liquid enthalpy in the reservoir, joules/gm.
h_{L3}	Liquid enthalpy in the combustion chamber, joules/gm.
h_{G3}	Gas enthalpy in the combustion chamber, joules/gm.
h_{L4}	Liquid enthalpy in the gun tube, joules/gm.
h_{G4}	Gas enthalpy in the gun tube, joules/gm.
K	Bulk modulus, MPa.
K_1	Bulk modulus at zero pressure, MPa.
K_2	Derivative of the bulk modulus, MPa.

M_{G3} Gas mass in the combustion chamber, gm.

M_{G4} Gas mass in the gun tube, gm.

M_1 Liquid mass in the reservoir, gm.

M_{L3} Liquid mass in the combustion chamber, gm.

M_{L4} Liquid mass in the gun tube, gm.

M_T Total propellant mass in the system, gm.

M_{pj} Projectile mass, gm.

M_{ps} Piston mass, gm.

$\dot{m}_{13}$ Mass flux into the combustion chamber, gm/s.

$\dot{m}_{34}$ Mass flux into the gun tube, gm/s.

$\dot{m}_3$ Rate of gas formation in the combustion chamber, gm/s.

$\dot{m}_4$ Rate of gas formation in the gun tube, gm/s.

P_1 Pressure in the liquid reservoir, MPa.

P_3 Pressure in the combustion chamber, MPa.

P_4 Average pressure in the gun tube, MPa.

P_L Pressure at the gun throat, MPa.

P_R Pressure at the base of the projectile, MPa.

P_{RS} Projectile resistance pressure, MPa.

R	Universal gas constant = 8.318 joules/mole-K.
S_{pj}	Projectile travel, cm.
S_{ps}	Piston travel, cm.
T_3	Temperature in the combustion chamber, K.
T_4	Average temperature in the gun tube, K.
V_1	Volume of the liquid reservoir, cm^3.
V_3	Volume of the combustion chamber, cm^3.
V_4	Volume of the gun tube behind the projectile, cm^3.
V_{L3}	Liquid volume in the combustion chamber, cm^3.
V_{G3}	Gas volume in the combustion chamber, cm^3.
V_{L4}	Liquid volume in the gun tube, cm^3.
V_{G4}	Gas volume in the gun tube, cm^3.
V_{pj}	Projectile velocity, cm/s.
V_{ps}	Piston velocity, cm/s.
W	Molecular weight of the gas, gm.
ε_3	Porosity of the combustion chamber.
ε_4	Porosity of the gun tube.
γ	Ratio of specific heats.

ρ_1 Liquid density in the reservoir, gm/cm.

ρ_{L3} Liquid density in the combustion chamber, gm/cm.

ρ_{L4} Liquid density in the gun tube, gm/cm.

ρ_{G3} Gas density in the combustion chamber, gm/cm.

ρ_{G4} Gas density in the gun tube, gm/cm.

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Director Defense Advanced Research Projects Agency ATTN: H. Fair 1400 Wilson Boulevard Arlington, VA 22209
1	HQDA DAMA-ART-M Washington, DC 20310
1	Commander US Army Materiel Command ATTN: AMCDR-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
3	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TSS SMCAR-SCA, M. Devine SMCAR-SCA-T, R. Yalamanchili Dover, NJ 07801
5	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCA, D. Downs A. Beardell SMCAR-LCE, N. Slagg SMCAR-LCS, W. Quine SMCAR-TDC, D. Gyorog Dover, NJ 07801
2	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL P. Votis Watervliet, NY 12189
1	Commander US Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	Commander US Army Aviation Research and Development Command ATTN: AMSAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120
1	Director US Army Air Mobility Rsch. and Development Lab. Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander ERADCOM Technical Library ATTN: STET-L Ft. Monmouth, NJ 07703-5301
1	Commander US Army Harry Diamond Labs ATTN: DELHD-TA-L 2800 Powder Mill Rd Adelphi, MD 20783
1	Commander US Army Missile Command ATTN: AMSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-YDL Redstone Arsenal, AL 35898

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Belvoir R&D Ctr ATTN: STRBE-WC Tech Library (Vault) B-315 Fort Belvoir, VA 22060-5606
1	Commander US Army Tank Automotive Cmd ATTN: AMSTA-TSL Warren, MI 48090
1	Commander US Army Research Office ATTN: Tech Library P.O. Box 12211 Research Triangle Park,NC 27709-2211
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range NM 88002
1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commander US Army Development and Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	Commander Naval Surface Weapons Center ATTN: Code G33, J. East Dahlgren, VA 22448
2	Commander US Naval Surface Weapons Ctr. ATTN: O. Dengel K. Thorsted Silver Spring, MD 20910
1	Commander Naval Weapons Center China Lake, CA 93555
1	Commander Naval Ordnance Station ATTN: C. Dale Code 5251 Indian Head, MD 20640
1	Superintendent Naval Postgraduate School Dept of Mechanical Eng. ATTN: Code 1424, Library Monterey, CA 93943
1	AFWL/SUL Kirtland AFB, NW 87117
1	Air Force Armament Lab ATTN: AFATL/DLODL Eglin, AFB, FL 32542-5000
1	AFOSR/NA (L. Caveny) Bldg. 410 Bolling AFB, DC 20332
1	US Bureau of Mines ATTN: R.A. Watson 4800 Forbes Street Pittsburgh, PA 15213
1	Director Los Alamos Scientific Lab ATTN: D. Butler P.O. Box 1663 Los Alamos, NM 87545

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Jet Propulsion Lab ATTN: Tech Lib 4800 Oak Grove Drive Pasadena, CA 91109
2	Director National Aeronautics and Space Administration ATTN: MS-603, Tech Lib MS-86, Dr. Povinelli 21000 Brookpark Road Lewis Research Center Cleveland, OH 44135
1	Director National Aeronautics and Space Administration Manned Spacecraft Center Houston, TX 77058
1	The BDM Corporation ATTN: Dr. T.P. Goddard P.O. Box 2019 2600 Cearden Rd, North Bldg Monterey, CA 93940
1	Calspan Corporation ATTN: Tech Library P.O. Box 400 Buffalo, NY 14225
1	Food & Machinery Corporation Northern Ordnance Division ATTN: J. Oberg Columbia Heights Post Office Minneapolis, MN 55421
3	General Electric Ord. Sys Dpt ATTN: J. Mandzy, OP43-220 R.E. Mayer H. West 100 Plastics Avenue Pittsfield, MA 01201-3698
2	General Electric Company Armanent Systems Department ATTN: E. Ashley M. Bulman Burlington, VT 05401
1	Science Applications, Inc. ATTN: R. Edelman 23146 Cumorah Crest Woodland Hills, CA 91364
2	TRW Electronics & Defense ATTN: D.R. Ausherman M.P. Bronstein One Space Park, 134/9048 Redondo Beach, CA 90278
1	Veritay Technology, Inc. ATTN: E. B. Fisher 4845 Millersport Highway, P.O. Box 305 East Amherst, NY 14051-0305
1	Director Applied Physics Laboratory The Johns Hopkins Univ. Johns Hopkins Road Laurel, Md 20707
2	Director Chemical Propulsion Info Agency The Johns Hopkins Univ. ATTN: T. Christian Tech Lib Johns Hopkins Road Laurel, MD 20707
1	Pennsylvania State University Dept. of Mechnical Eng ATTN: K. Kuo University Park, PA 16802
1	University of Michigan Gas Dynamics Laboratory Aerospace Engineering Bldg. ATTN: G. M. Faeth Ann Arbor, MI 48109-2140

DISTRIBUTION LIST

No. of Copies	Organization
2	Princeton Combustion Rsch Laboratories, Inc. ATTN: N.A. Messina M. Summerfield 475 US Highway One North Monmouth Junction, NJ 08852
1	University of Mississippi Mechanical Engineering Department ATTN: C.R. Wimberly University, MS 38677
1	University of Arkansas Department of Chemical Engineering 227 Engineering Building Fayetteville, AR 72701

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP, H. Cohen

Cdr, USATECOM
ATTN: AMSTE-TO-F

CDR, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number_____ Date of Report_____

2. Date Report Received_____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)_____

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.)_____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate._____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC
POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE