

AD-A013 577

PROPELLANT IGNITION AND COMBUSTION IN THE 155MM HOWITZER

E. B. Fisher, et al

Calspan Corporation

Prepared for:

Picatinny Arsenal

January 1975

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

237060

Calspan

Technical Report

AD A013577

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
JUL 16 1975
B

Calspan Corporation
Buffalo, New York 14221

Calspan

PROPELLANT IGNITION AND COMBUSTION
IN THE 155MM HOWITZER

E. B. Fisher and K. W. Graves

Calspan Report No. VQ-5524-D-2

Prepared For:

U.S. ARMY PICATINNY ARSENAL
DOVER, NEW JERSEY

JANUARY 1975

CONTRACT NO. DAA A21-74-C-0401

AMCMS CODE NO. 4932.05.41861

FINAL REPORT

PREPARED BY:

E. B. Fisher

E. B. Fisher
Principal Engineer
Systems Research Dept.

K. W. Graves

K. W. Graves
Principal Engineer
Systems Research Dept.

APPROVED BY:

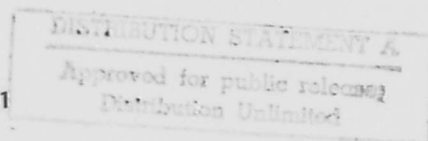
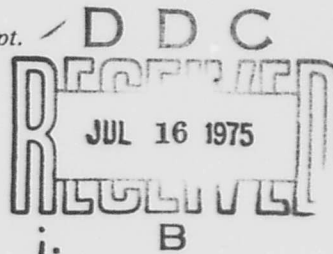
G. A. Sterbutzel

G. A. Sterbutzel, Head
Thermal Research Branch
Systems Research Dept.

E. D. O'Connor

A. D. O'Connor, Head
Systems Research Dept.

Calspan Corporation
Buffalo, New York 14221



FOREWORD

The work of this technical report was performed under Contract No. DAAA21-74-C-0401, by the Thermal Research Branch, Systems Research Department of Calspan Corporation for the Quality Assurance Directorate, Picatinny Arsenal, United States Army. The work was monitored by Messrs. F.J. Fitzsimmons and P. Serao of Picatinny Arsenal. The work described in this report was conducted during the period from May to December 1974.

This report contains only those modifications that were made to the Calspan 175mm gun model. It is recommended that References 1 and 2 be reviewed in order to obtain a better understanding of the details of the complete model.

ABSTRACT

A mathematical model has been formulated and programmed in FORTRAN IV, for use in the propellant charge design and investigation of the performance anomalies for the 155mm howitzer. The model solves the unsteady gas dynamic equations for conservation of mass, momentum, and energy by finite differencing simultaneously with auxiliary expressions for such important features as gas generation, bed friction, barrel boundary layer development, and projectile acceleration, until the projectile leaves the muzzle. This report details the mathematical concepts and experimental results that were incorporated into modifications of an existing mathematical model (for the 175mm gun) from which the 155mm howitzer model is derived.

The major modifications to the basic model include a better description of propellant movement, propellant bed compression, and more detailed representation of primer jet effects upon ignition sequences such as igniter base pad penetration, and heating of black powder. Flame spread in the base pad was also incorporated as well as center core loading tie failure and its subsequent compression and jamming in the igniter tube. Phenomena observed from movies of ignition sequence experiments were delay time for the important events and the effectiveness of ignition of the center core loading without benefit of the base pad igniter. Experiments were also conducted to measure propellant bed friction factor and to investigate the validity of the expression for bed friction.

Sensitivity analyses were performed using the computer program. They revealed the exceptional effectiveness of center core ignition for promoting uniform chamber pressure distribution during firing. The wave dynamics detected from some of the firing tests using the XM123E2 high-performance propellant charge were reproduced by the computer program only for charge configurations involving abnormal ignition such as a flawed igniter tube. One common cause of waves is the interference in the action of the igniter tube to relieve longitudinal pressure differences. The extreme case of the lack of this relief is represented by the end ignition configuration which was shown to be very sensitive to the initial conditions of standoff, loading, density, and chamber length. The first two conditions were of little significance to performance with a properly functioning center core igniter.

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
	FOREWORD	i
	ABSTRACT	ii
1.0	INTRODUCTION	1
2.0	SUMMARY OF ASSUMPTIONS	2
3.0	MATHEMATICAL MODEL CONFIGURATION	4
3.1	General Configuration of 155mm Howitzer	4
3.2	Chamber Routine	8
3.3	Barrel Routine	11
4.0	EXPERIMENTAL SUPPORT	14
4.1	Black Powder and Propellant Grain Ignition Tests	14
4.2	Center Core Igniter System Tests	17
4.3	Strength Test of Ribbons for Securing Igniter	19
4.4	Compression Test of Center Core Loading	19
4.5	Measurement of Propellant Bed Resistance to Flow	21
4.6	Conclusions and Summary of Experimental Effort	23
5.0	DETAILS OF THE MATHEMATICAL MODEL	26
5.1	List of Subroutines	26
5.2	Igniter Base Pad Penetration	30
5.3	Flame Spread in the Base Pad	31
5.4	Propellant and Propellant Bed Movement	32
5.5	Black Powder Movement in the Igniter Tube	34
5.6	Compressible Propellant Bed	35
5.7	Projectile Resistance to Movement	36
5.8	New Drag Expression	38
6.0	SENSITIVITY STUDIES CONDUCTED WITH THE COMPUTER PROGRAM	39
7.0	CONCLUSIONS AND RECOMMENDATIONS	54
8.0	REFERENCES	57
Appendix A	Program Flow Chart	58
Appendix B	FORTRAN IV Machine Listing	65
Appendix C	Required Input and Output	196
Appendix D	Glossary	210

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
1	BREAKDOWN OF 155MM HOWITZER PROPULSION SYSTEM	5
2	SCHEMATIC DIAGRAM OF THE 155MM HOWITZER SYSTEM ILLUSTRATING THE DOMAINS OF THE TWO COMPUTER ROUTINES	6
3	SCHEMATIC DIAGRAM OF THE 155MM HOWITZER PRIOR TO FIRING	7
4	MULTIPLE ONE-DIMENSIONAL CHAMBER GRID NETWORK FOR THE 155MM HOWITZER	10
5	GRID MATRIX FOR THE BARREL OF THE 155MM HOWITZER	12
6	HOT PLATE IGNITION TESTS FOR M30 PROPELLANT	15
7	HOT PLATE IGNITION TESTS FOR BLACK POWDER	16
8	TEST ARRANGEMENT FOR CENTER CORE IGNITION STUDIES	18
9	PROPELLANT-BED PRESSURE-DROP TEST	22
10	BARREL RESISTANCE FUNCTIONS BUILT INTO THE 155MM HOWITZER CODE	37
11	HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-2	42
12	HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ FIRING DATA	43
13	CHAMBER PRESSURE HISTORIES ~ RUN R-5	44
14	HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-6	46
15	CHAMBER PRESSURE HISTORIES ~ RUN R-6	47
16	HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-10	48

LIST OF FIGURES (CONT'D.)

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
17	CHAMBER PRESSURE HISTORIES ~ RUN R-11	49
18	COMPARISON OF CHAMBER PRESSURE WAVES FOR END IGNITION AND ABNORMAL CENTER CORE IGNITION	51

LIST OF TABLES

<u>Table No.</u>	<u>Title</u>	<u>Page</u>
I	COMPRESSION CHARACTERISTICS OF THE CENTER CORE LOADING IN THE IGNITER TUBE OF THE 155MM GUN	20
II	PRESSURE DROP TEST RESULTS IN INERT M30 PROPELANT BED	24
III	INPUT CONDITIONS AND RESULTS OF COMPUTER EXERCISES	40

1.0 INTRODUCTION

The 155mm howitzer is in the process of being upgraded to higher performance levels. The high performance propellant charge is designated XM123 and contains M30A1 propellant. The charge design incorporates a black powder base pad and center core igniter tube with a black powder charge.

Chamber pressure measurements made during firing tests with this propellant charge have exhibited dynamic wave phenomena under certain conditions. The waves have been strong enough in several instances to result in breech-blow gun failures. Pressures as high as 90,000 psi have been detected at the breech during one such failure. Subsequent investigation into the causes of dynamic phenomena have identified at least three sensitive parameters. These are charge standoff (the distance from the primer flash hole to the base of the charge), primer strength, and propellant loading density. For example, strong waves were observed with a strong primer, zero standoff, and high loading density.

Calspan has been actively engaged in formulating mathematical simulations of mortar and artillery systems. The mathematical model of the 175mm gun (References 1 and 2) was recently completed under Contract No. DAAA21-72-C-0577. This model provided representation of center core ignition, flame spread, and interior ballistics of the gun. Using this model it is possible to assess the sensitivity of propellant ignition and combustion characteristics on muzzle velocity as well as to aid in the understanding of phenomena associated with center core ignition.

The ultimate object of this project was to use Calspan's dynamic simulation of propelling charge ignition phenomena as an analytical tool to reveal details of the mechanisms that lead to wave generation, identify any additional sensitive parameters, and pose potential solutions to the problem.

The basic 175mm gun code was modified during this program to represent a variety of gun features and postulated events that could conceivably occur during a gun firing. The code underwent an exhaustive debugging and validation period and finally was used to conduct a parameter sensitivity analysis. The details of the code modification and improvements, and the results of the sensitivity analysis are presented in this report.

2.0 SUMMARY OF ASSUMPTIONS

The 155mm howitzer gun is a rather complex system. Performance depends on the action and interaction of numerous components and physical and chemical phenomena. Exact representation of all these would be impractical both in the implementation of the model and in its execution. Certain simplifying assumptions have helped to make the problem tractable while at the same time, the essential features have been retained. The important assumptions of the model are listed below:

- (1) The igniter tube is allowed to fail, grid by grid, either by tensile or compressive loading. In addition, combustion can cause the tube to disappear and allow propellant to shift into the region formerly occupied by the tube.
- (2) Propellant grains are assumed to have the thermal properties of good insulators and to be effectively semi-infinite in thickness for surface temperature calculations.
- (3) Ignition is assumed to occur when the surface temperature of a propellant or black powder grain reaches a critical value. This is allowed to occur in two stages: one due to the convective heating of gas flow through the propellant bed, and one due to direct impingement of gas flow through igniter tube holes. The latter stage affects only a small fraction of the total amount of propellant in a grid.
- (4) The gas state is assumed to be adequately represented by the ideal gas equation of state with secant values of gas properties defined by the ideal gas equation; that is, $H = c_p T$ and $R = P/\rho T$.
- (5) The propellant in each finite difference grid is represented by the dimensions of a typical grain, which are subject to change due to burning and an averaging process associated with propellant movement.
- (6) Main charge bag strength is assumed to be negligible considering the high pressures generated by the gas.

- (7) If a fraction of a grid contains a portion of the main charge igniter, that portion is assumed to be evenly distributed throughout the grid.
- (8) Propellant in the chamber is free to move in either axial direction but not radially.
- (9) Propellant in the barrel is free to move only in the direction of the projectile.
- (10) The rotating band of the projectile forms a perfect seal with the barrel.
- (11) Combustion of black powder and M30 propellant is assumed to be pressure dependent only. Erosive burning effects are not included.
- (12) The bands and grooves of the rifling are assumed to have no influence upon the character and growth of the boundary layer.

3.0 MATHEMATICAL MODEL CONFIGURATION

The mathematical model of the 155mm howitzer as modified from the existing 175mm gun model, is intended to serve as an analytical tool to aid in determining the causes of dynamic wave phenomena that have been observed in the XM123 propelling charge. The identification of these causes permits use of the mathematical simulation to evaluate potential engineering solutions to the problem. The model represents all elements of ignition, combustion, and their relationship to the wave problem.

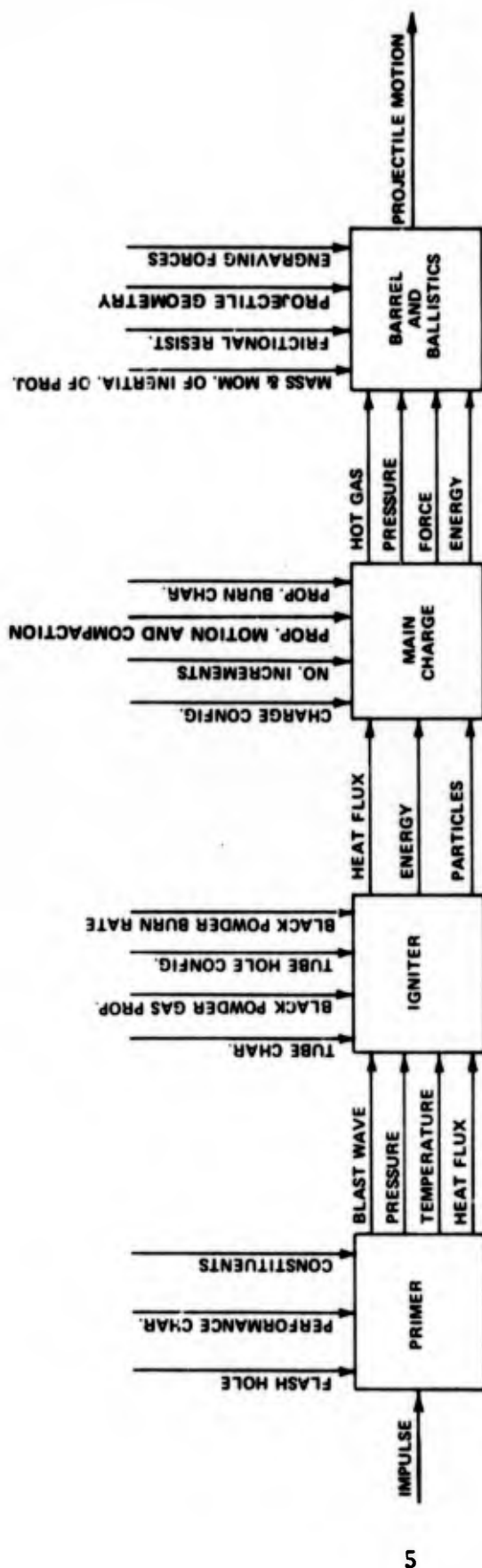
The problem of interest involves the ignition, flame propagation, propellant deflagration, and unsteady gas and propellant flow that occur during the firing of the 155mm gun. The interactions of the various components and phenomena are illustrated in Figure 1. It is important to include all these phenomena in the mathematical model so that the individual effects of each can be ascertained and applied to the general functional representation of gun dynamics.

The mathematical model consists of two major routines, the Chamber and Barrel Routines with domains as illustrated by Figure 2. The Chamber Routine calculates propellant ignition and deflagration in the gun chamber. The Barrel Routine handles the unsteady gas flow, boundary layer, and projectile motion through the barrel. The components of the 155mm howitzer that are considered in the model are defined and discussed in Section 3.1, while the actual structures of the two major routines are described in the remainder of the chapter.

3.1 GENERAL CONFIGURATION OF 155MM HOWITZER

The general configuration of the 155mm howitzer is shown in Figure 3. The shell provides the major portion of the cross-sectional area upon which the pressure acts. The rotating band performs a sealing function as well as the means for producing rotational acceleration. Any leaks past the band tend to reduce system efficiency. However, because this band undergoes an interference fit as it enters the barrel, the seal is assumed to be tight, allowing negligible loss of gas.

The ignition system and main propelling charge occupy the gun chamber. The maximum propelling charge consists of eight zones of M30A1 multiperf propellant. The maximum charge contains a nominal 26 lbs. of propellant. The full charge length is no more than 30.5 in., which is slightly less than the distance from the breech to the base of the projectile. The breech spindle of the XM198 155mm howitzer has positioning pads to assure a charge standoff of at least one inch. With this, the charge nearly touches the projectile base. The outer diameter of the charge is somewhat less than the chamber diameter in order to facilitate loading. This gap can allow gas to flow around in addition to through the bed of propellant and through a center core igniter tube.



PRIMER - PERCUSSION CHARGE AND DUCT
 CHAMBER { IGNITER - IGNITER TUBE AND BLACK POWDER IGNITER CHARGE
 MAIN CHARGE: UP TO EIGHT INCREMENTS OF M30A1 PROPELLANT
 BARREL BALLISTICS: BARREL FLOW AND PROJECTILE DYNAMICS

Figure 1 BREAKDOWN OF 155 mm HOWITZER PROPULSION SYSTEM

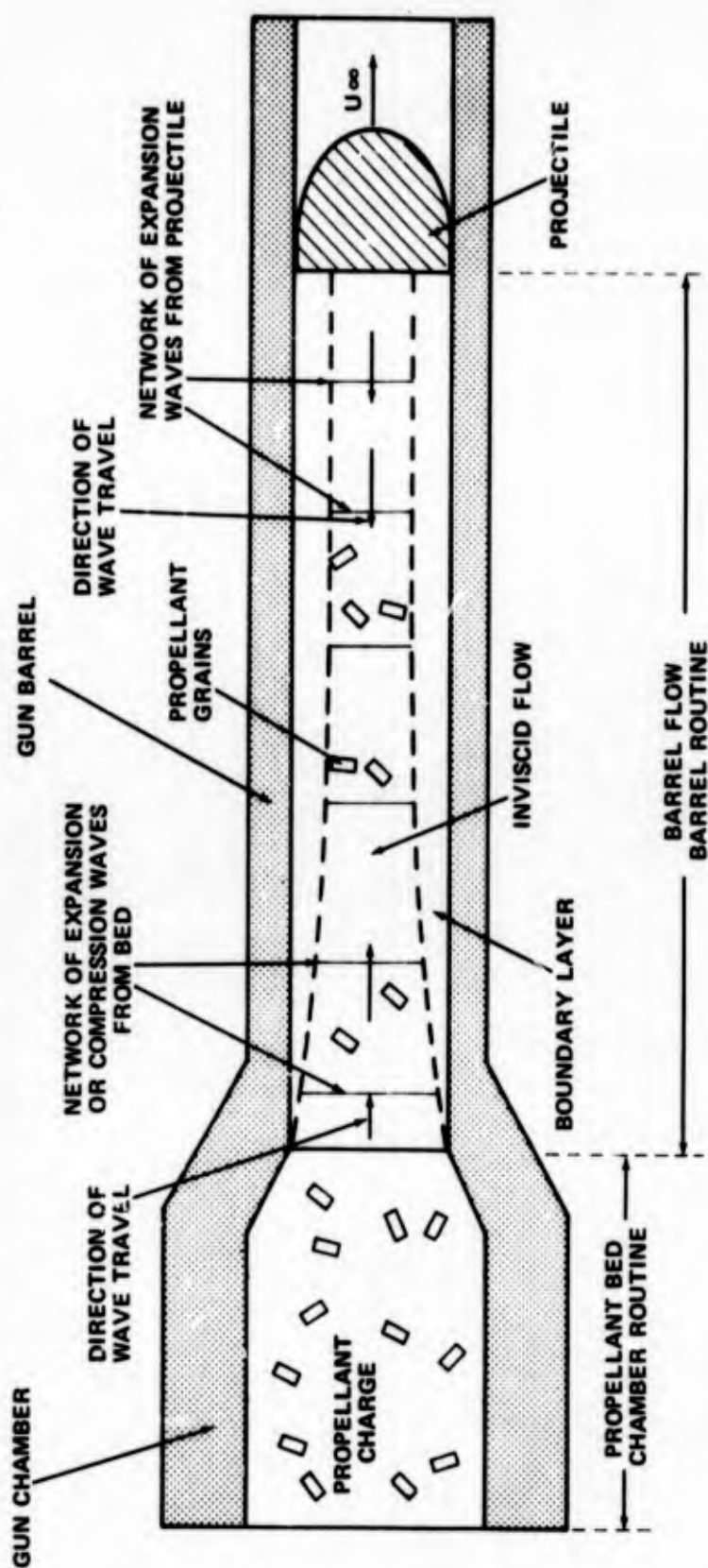


Figure 2 SCHEMATIC DIAGRAM OF THE 155 mm HOWITZER SYSTEM ILLUSTRATING THE DOMAINS OF THE TWO COMPUTER ROUTINES

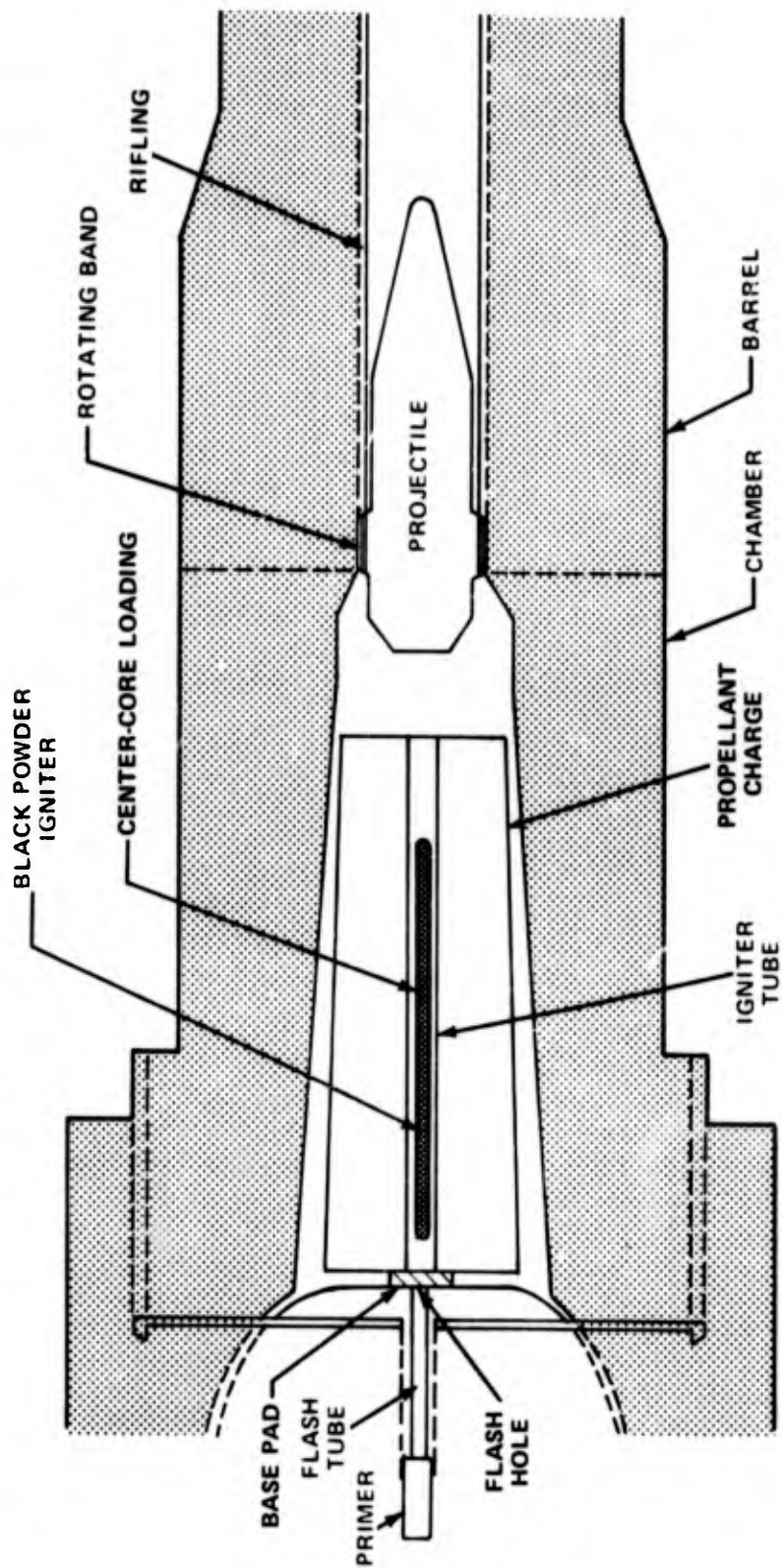


Figure 3 SCHEMATIC DIAGRAM OF THE 155 mm HOWITZER PRIOR TO FIRING

The XM123-Zone 8 propellant charge contains an igniter consisting of a nitrocellulose center-core tube enclosing a bag of black powder, which is tied to the tube at each end, known as the center core loading or snake. A base pad of black powder is located at the entrance to the center igniter tube. A gap of three inches separates the base pad from the beginning of the snake. The igniter tube contains a few holes near either end to anchor the center core loading. The tube is capped at the projectile end.

The actual gun system firing sequence is initiated when the primer is fired and primer gas flows from the flash tube. This causes a sequence of events resulting in black powder ignition. Hot gas and burning particles generated by the burning black powder flow through the igniter tube and into the end of the propellant bed. The grains of the main propellant charge and the combustible igniter tube are heated by this flow and eventually become ignited. After ignition, the propellant burns at a rate governed by local conditions. Gas flow through the propellant creates forces that result in movement of the bed. As pressure increases, loads on the burning igniter tube increase and eventually cause the tube to collapse. At this point, the space between the propellant bed and the chamber walls probably becomes loaded with propellant so that the chamber system may be simply represented as having a homogenous cross section of burning propellant.

As the pressure builds up in the system, the force created by pressure acting upon the projectile base overcomes the force restraining the projectile. This restraining force is a result of the material extrusion/shearing phenomena that occur while the rotating band is engraved. When this "shot start" force is reached, the projectile begins significant acceleration. As the projectile travels through the barrel, it is accelerated in a rotational direction at a rate proportional to the axial acceleration. These inertia forces, along with friction and engraving forces, constitute the projectile retarding forces.

Gas and propellant flow into the barrel behind the moving projectile. The gas loses energy and momentum through the boundary layer while it does work in overcoming retarding forces. The sequence of events of interest in this model terminates when the projectile leaves the muzzle.

3.2 CHAMBER ROUTINE

The Chamber Routine calculates all phenomena concerned with ignition, gas generation, and flow inside the chamber of the 155mm howitzer. This routine is basically the same as the corresponding routine for the 175mm gun code. The grid formulation consists of parallel one-dimensional networks, one to describe the center core igniter tube, one for the main charge, and one for the gap between the main charge and

chamber as shown in Figure 4. This system has many advantages, such as flexibility in defining the radial dimension of each grid network as a function of axial position, and arbitrary selection of grid size. Gas is allowed to flow between grid networks through simulated or pseudo holes in the boundary thereby achieving a semblance of radial mass and energy transport.

The basic equations of fluid motion with terms to take the porous, variable area bed into account are used to calculate flow propagation through the bed. These equations are the well-known, universal relationships that express conservation of mass, momentum, and energy. These equations contain terms to include gas generation by burning propellant and other source or sink terms such as heat transfer losses and mass flow through igniter tube and pseudo holes. In addition, equations expressing conservation of mass and momentum are included in order to express movement and compaction of the propellant bed.

The primer output is represented in terms of jet pressure, mass, and energy flux. The loads on the base pad and train of powder in the igniter tube are computed to determine (1) if a hole is punched in the base pad and (2) if the igniter charge of black powder is torn loose and driven down the tube. While ignition can occur naturally, instantaneous black powder and/or propellant ignition has been represented at time zero in one or more grids for sensitivity study purposes, in order to accelerate and accentuate certain events. In any case, remaining black powder becomes ignited by convective heating from the primer and previously ignited portions of the charge.

The treatment of flow through igniter tube and pseudo holes has been simplified but still retains the essential features. Gas flows sonically or subsonically through the holes, according to the existing pressure ratio, across the hole. This radial outflow creates radial gradients in the bed, but the model assumes that an exhausting elemental volume loses gas uniformly.

At some point, the igniter tube fails due to either tension or compression, it is or simply consumed by combustion. The failure is allowed to occur on a grid by grid basis. When the interior pressure exceeds the outer pressure by a given amount, the tube is assumed to split and a pseudo hole is added to the grid where a solid boundary had previously existed. If failure occurs in compression, the tube is assumed to collapse, a pseudo hole is added, and, in addition, propellant from the main charge is mixed with the black powder on the axis. When the black powder is completely consumed, the multiple one-dimensional grid matrix is converted into a single one-dimensional network. This is achieved through a weighted average of all grids at each axial location throughout the chamber.

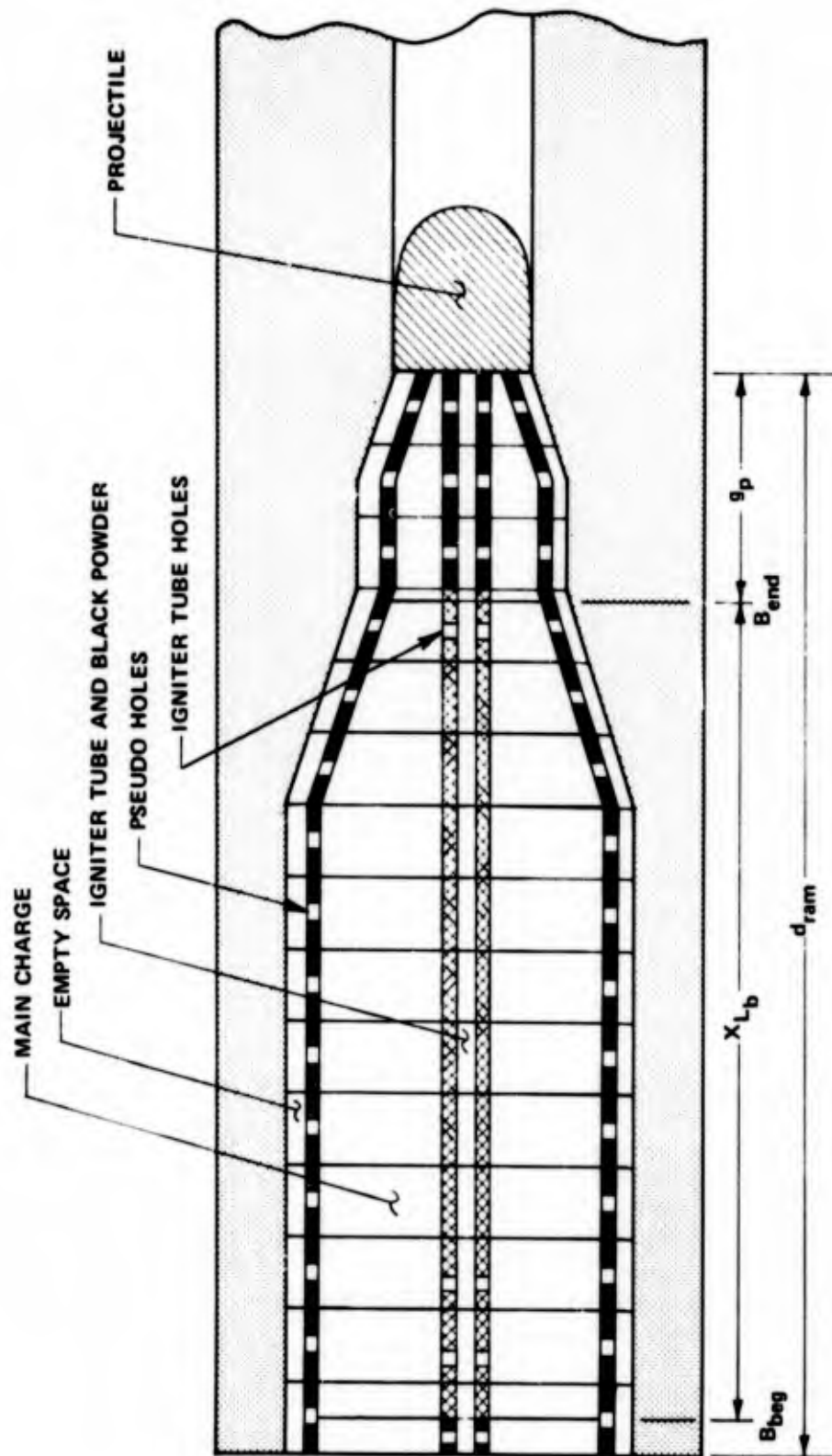


Figure 4 MULTIPLE ONE-DIMENSIONAL CHAMBER GRID NETWORK FOR THE 155 mm HOWITZER

The breech end of the chamber is assumed to be reflective; that is, waves are reflected with no losses. The multiple one-dimensional formulation requires no specification of wall boundary conditions. The downstream end of the chamber is non-reflective and allows a smooth flow of gas into the barrel after the projectile has moved at least one grid length. Prior to this, the projectile base conditions are computed by the chamber routine and the base is assumed to be a reflective surface.

Basic inputs for the Chamber Routine include the chamber and propelling charge geometry pertinent to propellant ignition, gas generation and flow, and propellant geometry and burning characteristics. Essentially all elements of the igniter system that could conceivably influence gun performance were included in the mathematical model. Virtually none of these elements is built into the program but, rather each is an input that can be varied independently from the others.

3.3 BARREL ROUTINE

The Barrel Routine accepts the flow of gas and burning propellant from the chamber and performs the unsteady gas flow and projectile motion calculations until the projectile eventually passes from the barrel. These calculations are performed in a one-dimensional framework which assumes that all two-dimensional effects can be assigned to boundary layer -type calculations. The grid network used to represent the barrel is shown in Figure 5.

The one-dimensional equations of fluid motion, modified to take the presence of solid propellant grains into account, are used to calculate the gas flow. These equations express conservation of mass, momentum, and energy for each grid and include losses of momentum and energy as well as the mass flow area constriction due to viscous effects of the boundary layer in the barrel and heat transfer to the barrel wall. Propellant movement is calculated from pressure gradients and drag forces exerted by gas flow. This is simplified by allowing propellant to move in one direction, away from the breech.

The individual items that influence projectile motion have been accounted for separately rather than being lumped into an effective projectile mass or resistance function. The main propelling force is that due to pressure acting on the projectile base. Retarding forces are considered individually and consist of the force required to engrave the rotating band, the component of the accelerating force producing rotational acceleration, and frictional resistance. The engraving force is a result of the extrusion process and subsequent slip fit/galling condition encountered by the projectile rotating band as it begins motion through the barrel. Rotational acceleration depends upon the axial moment of

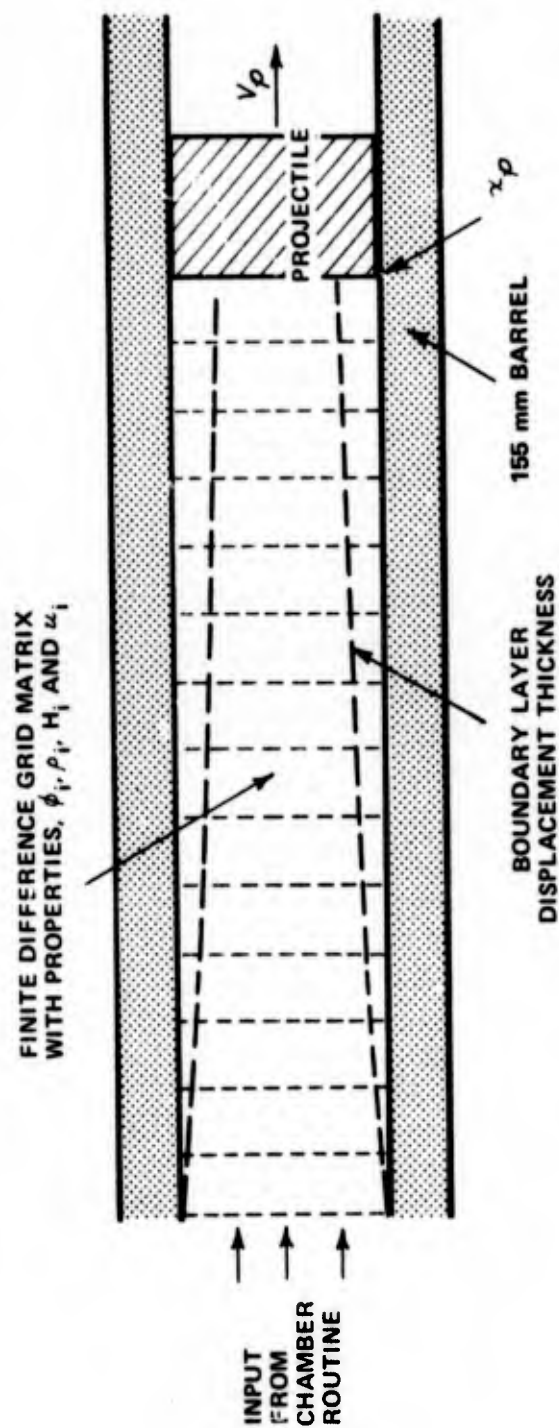


Figure 5 GRID MATRIX FOR THE BARREL OF THE 155 mm HOWITZER

inertia and the twist of rifling. It is a component of the total acceleration that detracts from the effect of the pressure force, and in this sense, it acts as a retarding mechanism. The frictional force is determined as a result of rotational acceleration. The torque required for rotational acceleration is supplied by a resultant force normal to the rifling. The retarding force occurs as a result of the coefficient of friction between the rotating band and the rifling and this force. Another resistance force that has been included is the pressure head that is developed ahead of the projectile.

Barrel Routine calculations are initiated with the projectile at rest and located at the first or second grid of the barrel network, whichever is specified. When the pressure force exceeds the assumed initial resistance force the projectile starts to move. As the projectile travels through the barrel, grids are added to the network. Initially, a relatively small grid size is required in order to supply the required computational accuracy. As the projectile moves through the barrel, the number of grids in the entire system is cut in half from time to time, greatly accelerating the calculation while providing acceptable accuracy.

The one-dimensional barrel calculations require no specification of radial boundary conditions. The initial grid of the barrel network is common with the last row of chamber grids and is loaded with weighted averages of parameters from these chamber grids. Therefore, no specific boundary conditions are applied to the barrel entrance. The barrel grid network is terminated at the projectile base, which is a reflective boundary moving at the projectile velocity.

Inputs to the Barrel Routine consist mainly of projectile characteristics, which include equivalent pressures to represent retarding forces, mass and moment of inertia, representative base radius, and friction coefficient. Barrel length and constants describing the twist of the rifling are also input parameters.

4.0 EXPERIMENTAL SUPPORT

A variety of experiments were conducted to investigate peculiarities of the action of some components and to measure certain quantities to aid in the description of associated phenomena. The use of properly designed experiments for these purposes is necessary to insure that the simulations are realistic in those cases where the available theory and experimental results are deficient. The experiments deemed necessary for adequate modeling of the 155mm gun included tests for measuring ignition temperatures of black powder and M30 propellant, experimental studies of center core ignition by high speed photography, compression characteristics of black powder, and propellant bed resistance to flow. Details of these experiments are described in the subsequent paragraphs.

4.1 BLACK POWDER AND PROPELLANT GRAIN IGNITION TESTS

Tests were conducted to measure the ignition temperature of both M30 propellant and black powder using the hot plate method. A steel plate, 1/2 inch thick, 4 inches by 6 inches, was fitted with an imbedded chromel alumel thermocouple by peening it into a shallow hole at the center of the plate. It was placed on a tripod for heating by a large Bunsen flame so that the thermocouple wires were not exposed to flame. Readout was done by means of a Brown potentiometer. The procedure was to adjust the flame to obtain a steady plate temperature, and then to drop the powder granule or grain onto the plate near the thermocouple while simultaneously starting a stopwatch. The stopwatch was stopped in the case of the propellant when vigorous degassing started (this was usually followed by flame in one or two seconds). In the case of black powder it was stopped when the black powder exploded into flame. Only the larger size black powder granules were used in the tests because it was found that small or thin granules ignited much sooner than the bulkier granules. The contacting surface of these larger granules was approximately 3/16 x 3/16 in.

Results are plotted in Figures 6 and 7. Although this method produces heat transfer to the grain in a different manner than it experiences in a gun chamber, and the actual temperature measured is that of the plate that heats the grain instead of the surface of the grain, the method is used universally and provides a relative value for ignition temperature that is useful. It is especially easy to interpret if, as plate temperature is slowly decreased, a point is reached where delay begins to increase rapidly. This is definitely the case with M30 propellant. The curve for black powder does not show this point because test temperatures were limited to 860°F but the trend is obvious. The inputs to the computer program require a single value of ignition temperature of each component. It can be seen from the figure that 850°F for black powder and 450°F for M30 would be reasonable values to use for the program.

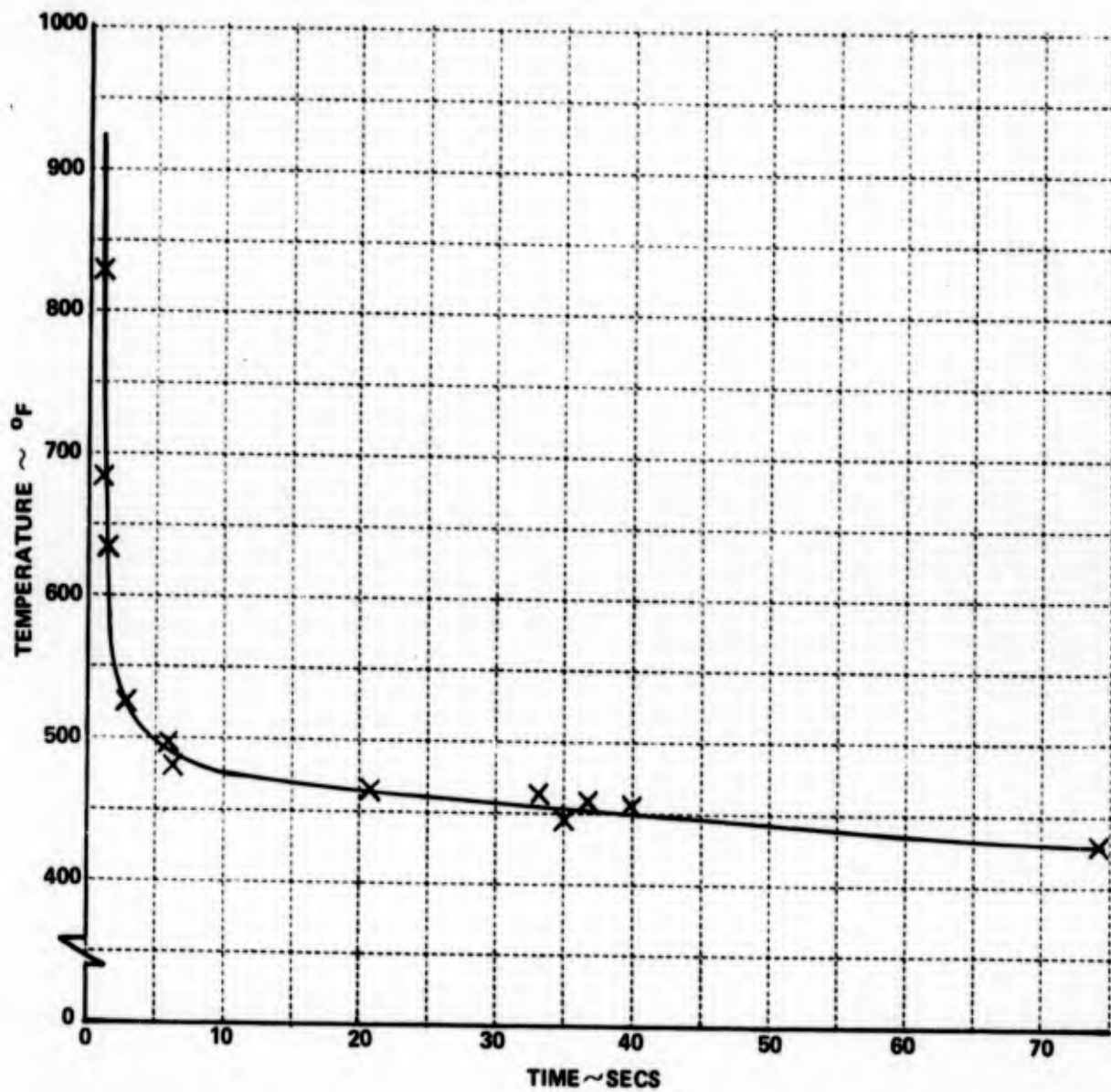


Figure 6 HOT PLATE IGNITION TESTS FOR M30 PROPELLANT

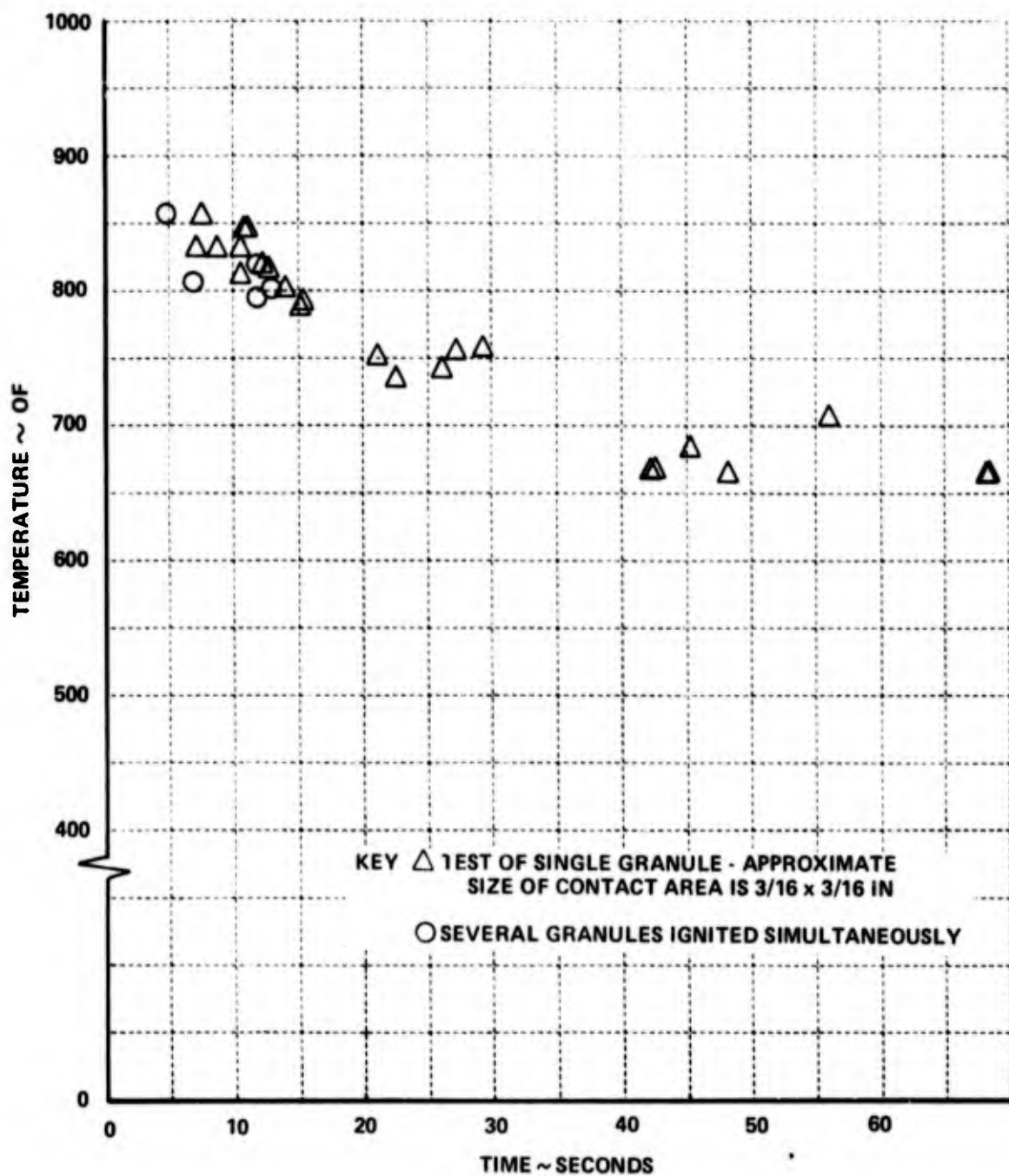


Figure 7 HOT PLATE IGNITION TESTS FOR BLACK POWDER

4.2 CENTER CORE IGNITER SYSTEM TESTS

High speed motion pictures were taken of the ignition sequences and burning with the center core loading installed in a transparent tube. The test setup was the same as previously used for similar studies of the 175mm igniter system, Ref. 1. It was arranged so that the M32 primer fired through a 1/4 inch inside-diameter flash tube, 5 1/2 inches long. The jet emerging from this tube was directed into a transparent tygon tube of 1 inch inside diameter across a gap that could be varied in distance to represent standoff and to accommodate the base igniter pad. The transparent tube was 28.5 inches long with holes in its walls for securing the center core loading as in the 155mm igniter tube. Two Fastax motion picture cameras were set up several feet away to photograph the burning sequences. A large shield was fitted over the transparent tube, located 1 inch from its primer end and secured in an orientation perpendicular to the tube axis to delay obscuration by smoke in the optical path of the cameras. The test setup is shown schematically in Figure 8.

Three firings were conducted. The first incorporated a full igniter charge complete with base pad and center core loading at a stand-off of 1/2 inch. The second used only a partial center core loading (1/2 oz. of black powder) sewn snugly into a shortened igniter bag and secured in the transparent tube with its primer end situated exactly as would the end of a full charge. The purpose of this test was to measure the ignition delay at a total gap of 1 inch without benefit of the base igniter pad.

The third test again used a full igniter charge but at 0 stand-off to determine if the center core loading would move down the tube in any manner during the early stages of its burning.

No perceptible movement of the igniter bag took place during the first test. After initial flow of the primer jet (ignition of the base pad could not be distinguished from this) the powder in the base pad burned vigorously. But its incandescence was considerably diminished before initial ignition occurred in the igniter tube. This delay was 335 milliseconds! The center core loading ignited several points along the tube and by 20 milliseconds later it had erupted into a single bright image over the full length of the tube.

The second test revealed that the center core loading ignited almost instantly 5 milliseconds after firing of the primer even though the nearest grains of black powder were 4 inches from the jet exit. Once again the ribbons securing the igniter bag were not torn away by the primer jet blast. These tests vividly demonstrate that the presence of the base pad seriously delays center core ignition.

The third test showed the same events even more clearly than did the first. Ignition spots on the center core loading were very

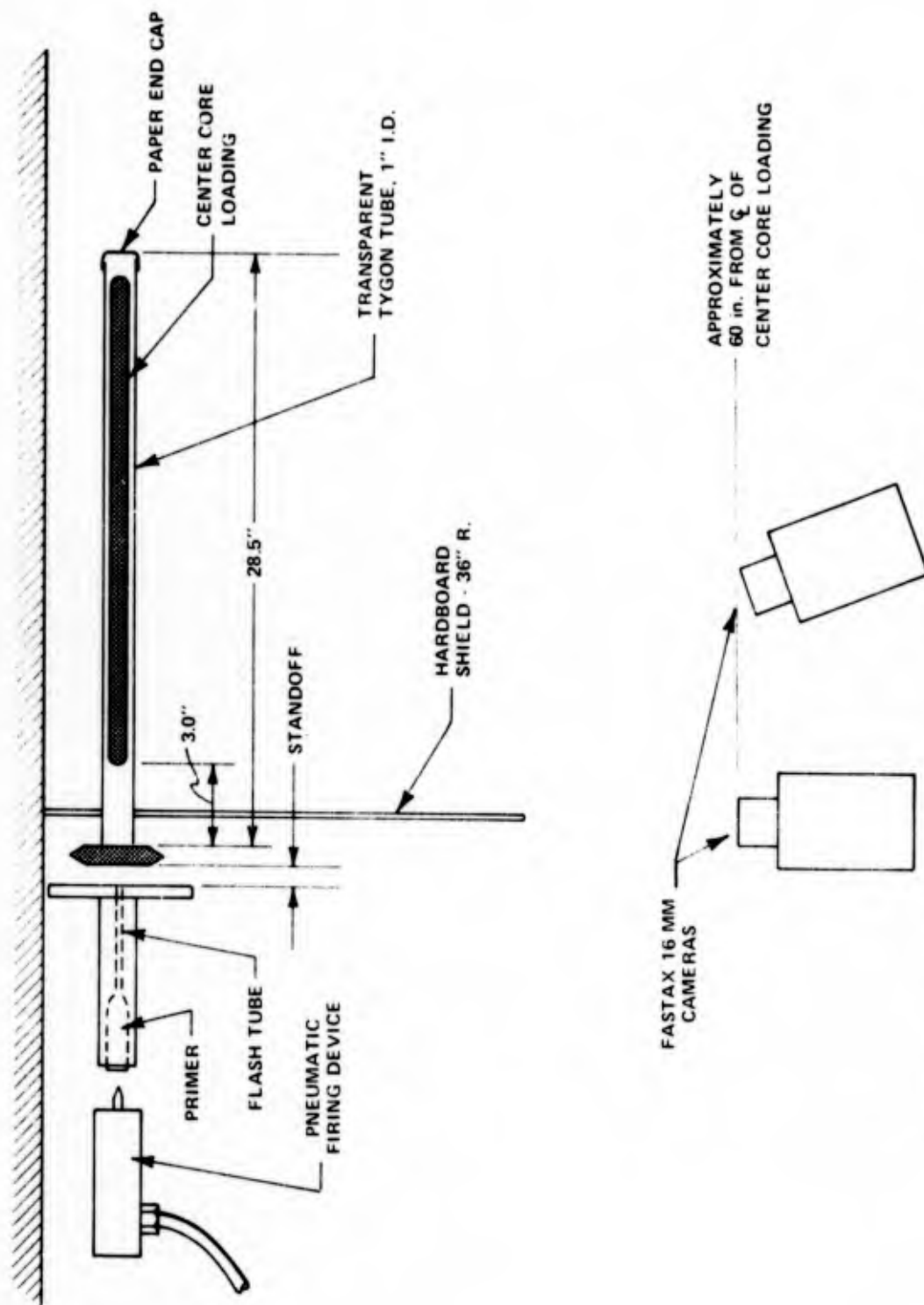


Figure 8 TEST ARRANGEMENT FOR CENTER CORE IGNITION STUDIES

distinct and after eruption gases could be seen to flow in both directions away from the central regions of the tube. Again the ribbons securing the igniter bag held the charge until all was flame in spite of the zero standoff, which demonstrates that as long as the M82 primer is used for the 155mm gun this igniter system can be considered to burn in its original loaded configuration. The delay between the appearance of flame from the burning base pad and ignition of the center core loading was 90 milliseconds.

4.3 STRENGTH TEST OF RIBBONS FOR SECURING IGNITER

A test was conducted to measure the strength of the pair of ribbons that are used to secure the center core loading in the igniter tube. The test fixture was arranged to load the ribbons as well as their attachment to the loading bag so that failure would reveal the weaker point, as well as a measurement of its strength. The bag was clamped in a vise so that the ribbons hung below it where they could be attached to a pan for applying the load by hanging weights. This was done in increments of one pound until a total weight of 24 pounds was reached whereupon the ribbons tore away at their point of attachment to the bag.

4.4 COMPRESSION TEST OF CENTER CORE LOADING

A test was devised to measure the compression characteristics of the center core charge in the igniter tube. A center core charge was cut to a four inch black powder bed length and the cut end was closed by stitching. It was placed in a six-inch long rigid tube having the same inside diameter as the igniter tube (1.17 in.) and this was placed upon a weighing scale so that it rested on one end. A cylindrical shaped ram was placed in the other end (its diameter was only slightly less than the tube I.D.) so that its lower end contacted the bag of black powder. The ram extended several inches above the upper end of the tube so that it could be pressed into the tube a sufficient distance to crush the loading completely.

The whole apparatus was placed in an arbor press for application of the compressive load. It was applied slowly and interrupted at intervals, while a steady load was maintained so that the travel of the ram could be measured by a vernier scale. The load was increased in steps in this manner until a load of 200 lbs. was reached.

Results are presented in Table I as compressive force vs. ram travel. These data were proportioned for a full length core loading of 24 in. (multiplied by 6) and presented in the third column as equivalent deflection of core loading.

TABLE I

Compression Characteristics of the Center Core
Loading in the Igniter Tube of the 155mm Gun

Force lb.	Ram Travel -inches	Equivalent Deflection of Core Loading - in
20	0.361	2.166
30	0.472	2.832
40	0.781	4.686
50	0.921	5.526
60	1.080	6.480
70	1.125	6.750
80	1.203	7.218
90	1.406	8.436
100	1.437	8.622
110	1.687	10.122
160	1.765	10.590
170	1.875	11.250
200	2.063	12.378

4.5 MEASUREMENT OF PROPELLANT BED RESISTANCE TO FLOW

The resistance to gas flow offered by a bed of gun propellant is a governing factor that determines the movement of the propellant. Therefore, flow resistance data for actual 155mm gun propellant is necessary for use in the mathematical model if proper movement of the propellant is to be simulated. Because specific data was nonexistent a special apparatus was fabricated for its measurement using steady air flow through a bed of actual or inert propellant. The test philosophy was based upon duplication of grain dimensions in a bed of representative size, and duplication of Reynolds number for modeling the flow conditions. The use of cold air for testing was made possible by the fact that the Reynolds number for hot propellant gas flow, at conditions of chamber pressure above 20,000 psi, can be duplicated in cold air flowing at comparable velocity but with pressures from 10 to 100 psi. The object was to obtain sufficient data to evaluate bed friction factor and to determine the validity of the relations used for computing bed friction, i.e., drag in the model.

The apparatus confines a bed of propellant within a 4-inch-diameter pipe between heavy screens so that air may be blown upward through it over a wide range of pressure. To achieve this, the apparatus is connected to a compressed air supply so that pressure at the bed entrance may be varied up to 250 psi. A schematic diagram of the apparatus is shown by Figure 9. It is essentially a 4-inch I.D. pipe fixed in a vertical orientation, with a 1 foot long removable length that can be filled with propellant grains. These are restrained from falling through by a heavy screen of 3/16 inch diameter wire spaced 3 x 3 to the inch. In addition, an upper screen of the same material is provided to prevent lifting of the bed by the air flow. This screen can be positioned at various heights by the use of spacers. Pressure taps and thermocouple fittings are provided along the bed walls at 2-inch intervals. The pipe outlet is fitted with a precision orifice meter made and installed to satisfy requirements as a choked-flow meter according to Ref. 3.

Inert M30 propellant was tested over a wide range of air flow from that just sufficient to lift the bed off its lower screen, at a Reynolds number = 4500, based upon bulk velocity and effective particle size, to a flow rate 25 times greater where the particle Reynolds number was 163,000. Average grain length was 0.80 inch and outer diameter was 0.42 inch for an effective particle size, $D_p = 0.59$ inch (defined as the diameter of a sphere having the same volume as the particle). In addition, a shape factor of 0.82 was computed from the relation $\phi_s = 4d/(d+4\ell)$ where ℓ is grain length and d is its diameter, Ref. 4.

In preparation for the tests, 1326 grams of inert propellant were placed in the test section. When it was vibrated until completely settled, the bed depth was 7 1/2 inches. At this condition, the measured

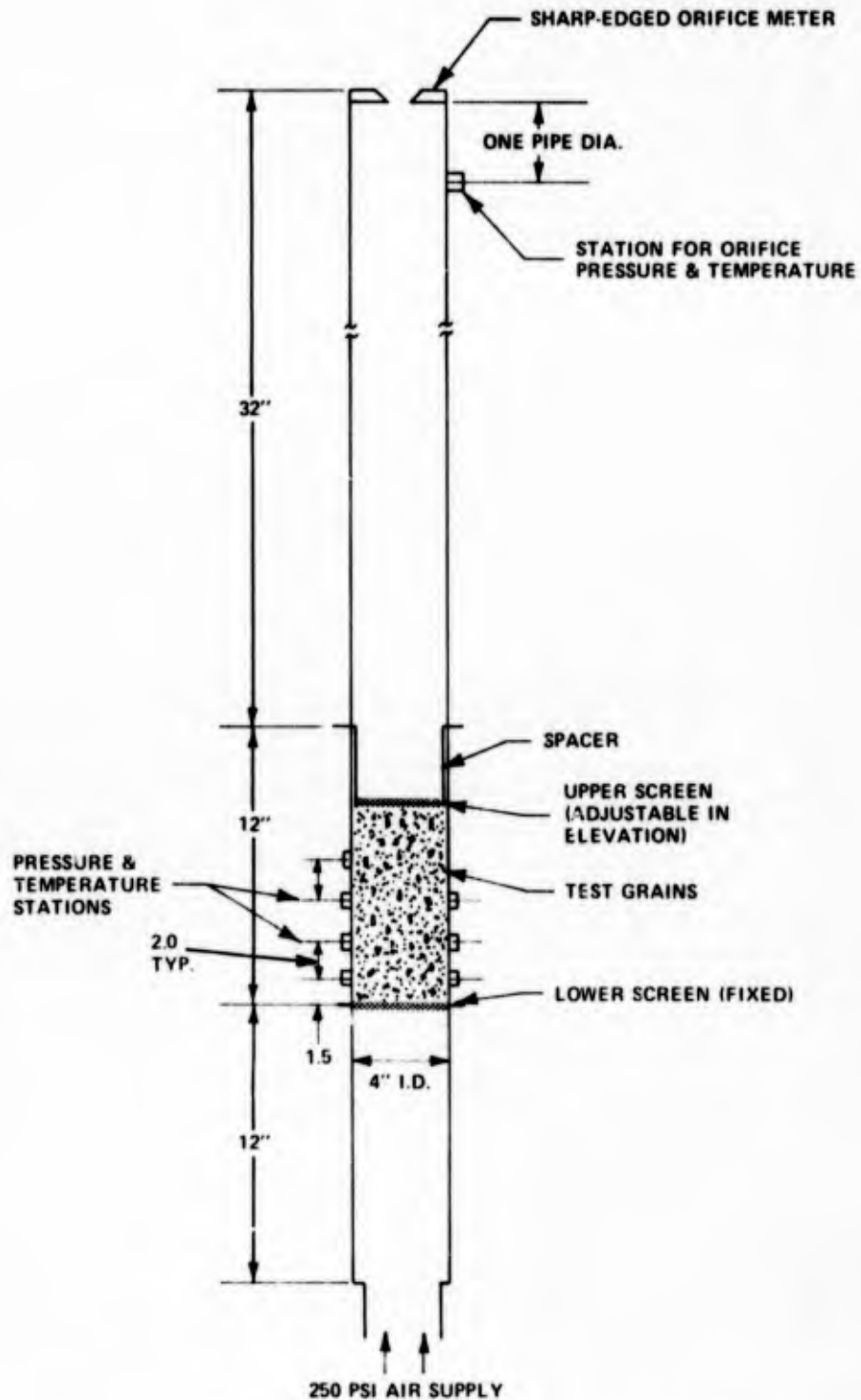


Figure 9 PROPELLANT-BED PRESSURE-DROP TESTER

voidage was 0.41, obtained by pouring a measured amount of water into a straight sided container holding similarly packed propellant to a known depth. Bed pressure drop was measured for the fluidized bed configuration, which was obtained with the upper screen located 2 inches above the top of the bed. Pressure drop at higher flow rates was measured with the bed restrained completely. Test data and results are presented in Table II. Friction factor was computed from the relation:

$$f = \frac{0.25 \phi_s \phi^3 \Delta p}{1 - \phi} \frac{1}{\rho} \frac{L}{D_p z} \frac{V^2}{g}$$

where	D_p	is the diameter of a sphere having the same volume as the particle
	L	is the bed length, ft.
	Δp	is the pressure drop, psf.
	V	is the superficial velocity, = ϕ times actual velocity, fps
	ρ	is the fluid density, lbs/ft ³
	ϕ	is the porosity
	ϕ_s	is the shape factor, equal to the area of a sphere having the same volume as the particle divided by the particle surface area

The results show that friction factor varies only slightly over the test range of Reynolds number. The amount of variation is approximately the same as the experimental error, except for the first and last run. Furthermore, the fact that f is almost invariant over the range of Reynolds number from 40,000 to 140,000 range lends credence to the validity of the expression for bed friction from which the foregoing relation for f is derived. This expression was, of course, originally obtained by a correlation of results from tests similar to those described in this report but using different shapes of granulated materials.

4.6 CONCLUSIONS AND SUMMARY OF EXPERIMENTAL EFFORT

Results obtained from experiments described in this report were very useful to the correct representation of the 155mm howitzer by the computer model. Specific phenomena were demonstrated and their natures defined in sufficient detail to permit mathematical description.

The ignition temperature as a function of delay was measured by the hot plate method for black powder and the M30 propellant grain. Although this method suffers obvious disadvantages as pointed out previously its results were easy to interpret in the case of black powder and M30 propellant, thereby yielding useful values for ignition temperature as required by the computer program. Specifically, these were 850°F for the black powder, and 450°F for the M30 propellant.

TABLE II

Pressure Drop Test Results in Inert M30 Propellant Bed

Run	Air flow rate lb/sec	Average Bed Air Density lb/ft ³	Bulk Air Velocity fps	ΔP across height, h~in.	h	Re _p	f
1*	0.09	.081	12.6	0.82	7.5	4530	1.102
2	0.927	.225	45.2	7.0	2.0	47200	0.916
3	1.030	.254	46.5	8.0		73300	0.955
4	1.204	.298	46.3	9.0		86000	0.922
5	1.528	.373	47.0	11.5		108700	0.916
6	1.652	.416	45.5	12.5		121300	0.950
7	1.780	.450	45.4	13.0		131200	0.920
8	1.938	.485	45.8	14.0		141400	0.901
9	2.260	.560	46.3	15.0		163300	0.820

* indicates the fluidized bed condition. The pressure drop of 0.82 psi was just sufficient to lift the total weight of 2178 gms. (4.78 lb.) constituted by the propellant bed and the upper screen.

High speed motion pictures of very good quality were taken of the ignition and burning of the center core loading both with and without the base igniter pad. This was made practical by the use of a thick tough transparent tube for simulating the igniter tube that houses the center core loading. Certain time delays were measured that could prove useful in assessing the effectiveness of the computer codes. Furthermore, specific phenomena were revealed such as the fact that the M82 primer jet does not dislocate the center core black powder charge, and that the charge ignites over its full length quite rapidly once it becomes ignited at some point. In addition, it was demonstrated that the presence of the base igniter pad absorbs some of the impact of the primer and drastically delays ignition of the center core charge.

The ribbons that secure the center core loading to the igniter tube were strength tested. Although the forces of the M82 primer jet will not cause ribbon failure, the data has been used in the program for studying the effects of firing with the XM119 primer, which may create forces sufficiently high to cause failure.

The center core loading was compression tested. These results mainly apply to the representation of the charge configuration with the stronger primer, because the need is contingent upon ribbon failure.

Propellant bed flow resistance was measured over a wide range of flow rate and particle Reynolds number. Results show that the revised relation for computing bed resistance to flow, as postulated in this report (See Section 5.8), is a valid formula for this purpose inasmuch as it provides a correlation of test data with a constant value for friction coefficient over the Reynolds number range of interest. Friction coefficient for M30 propellant was computed to be 0.95 from the data for the range of interest.

5.0 DETAILS OF THE MATHEMATICAL MODEL

The 155mm howitzer computer code consists of the 175mm gun model described in Ref. 1 and 2 with several additions and modifications. The majority of the forty subroutines employed in the model experienced no or only minor changes. Only those new significantly modified subroutines are described in this chapter. However, all subroutines are listed in Section 5.1 and given a one-line description. A program flow chart and complete FORTRAN IV machine listing are given in Appendices A and B. Program input and output are discussed in Appendix C.

5.1 LIST OF SUBROUTINES

The computerized mathematical model is characterized as consisting of two major routines: Chamber and Barrel. The subroutines contained in these major routines as well as those that perform a function not specifically associated with a routine are given a one-line description below. A detailed discussion of the subroutines in the original code is presented in Ref. 2, while those subroutines added or modified during this program are described in subsequent sections.

Chamber Routine

- | | |
|-----------|---|
| (1) AREAS | Calculates grid volumes and areas for use in finite difference calculations. |
| (2) AXIS | Performs finite difference calculations in the igniter tube, the axis of the propellant charge. |
| (3) AXIT | Performs two-dimensional finite difference calculations in the propellant bed adjacent to the igniter tube. |
| (4) AXIT2 | Performs one-dimensional finite difference calculations in the propellant bed. |
| (5) AXIT3 | Performs one-dimensional finite difference calculations in the void space between the propellant charge and chamber wall. |

- (6) BNDY Performs two-dimensional finite difference calculations at the outer boundary.
- (7) BPFIR Calculates heating and eventual ignition of black powder grains.
- (8) BPINIT Reads in and initializes black powder igniter charge.
- (9) BSURFA Performs two-dimensional finite difference calculations on the axis of an upstream solid boundary (breach).
- (10) BSURA2 Performs one-dimensional finite difference calculations in the igniter tube at the breach.
- (11) BSURFB Performs two-dimensional finite difference calculations at the outer radial boundary of an upstream, solid axial boundary.
- (12) BSURFI Performs two-dimensional finite difference calculations at an interior point of an upstream solid boundary.
- (13) BSURFT Performs two-dimensional finite difference calculations at the outside igniter tube surface and an upstream solid boundary.
- (14) BSURT2 Performs one-dimensional finite difference calculations for the propellant charge at the breach end.
- (15) BSURT3 Performs one-dimensional finite difference calculations for the void region between the propellant charge and the chamber wall at the breach end.
- (16) CHSET Reads in initial chamber data, initializes the chamber grid matrix and performs other necessary preliminary calculations before finite difference calculations begin.
- (17) DRAG Performs simultaneous solution of drag and updated gas velocity for finite difference subroutines.

- (18) FSURFA Performs two-dimensional finite difference calculations at the axis of a downstream solid boundary.
- (19) FSURFB Performs two-dimensional finite difference calculations at the outer radial boundary of a downstream solid boundary.
- (20) FSURFI Performs two-dimensional finite difference calculations at interior grid points at a downstream solid boundary.
- (21) FSURFT Performs two-dimensional finite difference calculations at the outer boundary of the igniter tube and a downstream solid boundary.
- (22) HOLES Calculates the igniter tube hole area that is assigned to each grid.
- (23) HOLSET Sets up the arrays of holes to include both igniter tube holes and "pseudo" holes that provide transport between adjacent grid networks.
- (24) INTER Performs two-dimensional finite difference calculations at interior points of the grid matrix.
- (25) MFLOW Calculates mass flow through igniter tube and pseudo holes.
- (26) ONEDIM Transforms the Chamber grid matrix from multiple one-dimensional or two-dimensional network into a single one-dimensional grid network.
- (27) PATHS Assigns values to an array of parameters that permits logical selection of the proper finite difference subroutine.
- (28) PRIMER Calculates burning of black powder in the 155mm howitzer.
- (29) PROPEL Calculates movement of propellant grains in the chamber.

- (30) PRPFIR Calculates two-stage heating of propellant grains and eventual ignition.
- (31) PRPVEL Calculates propellant grain velocity and black powder movement.
- (32) REGRES Calculates burning of main charge.
- (33) TUBFAL Calculates failure of igniter tube due to pressure loading and combustion.

Barrel Routine

- (1) BARSET Reads in barrel and projectile input data, initializes the barrel matrix, and performs other initializing functions prior to start of finite difference calculations.
- (2) BNDLYR Calculates the tube boundary layer with associated viscous and heat transfer losses.
- (3) DIMIN Calculates deflagration of propellant that has moved into the barrel.
- (4) MOTION Calculates projectile acceleration and movement.
- (5) PROPMO Calculates velocity and movement of propellant that has moved into the barrel.
- (6) RHOUH Performs all finite difference calculations in the barrel grid matrix.

Other Subroutines

- (1) CLEAR Zeroes specified arrays or blocks of common storage.
- (2) GSPROP Calculates propellant gas properties.
- (3) MAIN Performs main logic function for the 155mm howitzer mathematical model.
- (4) NEWDX Reduces the number of grids involved in the finite difference grid matrix.

- (5) UPDATE Updates the finite difference variables to the next time interval and provides for output of these variables.

The more significant additions and improvements provide for representation of such phenomena as: penetration of the igniter base pad by the force of the primer jet, radial flame spread in the igniter base pad after ignition at its center by the primer jet, movement of both propellant and black powder in large slugs packed to maximum density and encompassing many grid lengths, as well as in slugs of single grid length across grid boundaries, elastic response of the packed propellant when loaded in compression, and a more thorough expression for determining pressure drop through the propellant bed based upon actual experiments on the M30 propellant bed.

5.2 IGNITER BASE PAD PENETRATION

Base pad penetration can be brought about by either the process of burn-through after ignition or by shear failure as a result of primer jet blast. The former phenomenon was featured in the basic computer program as were some of the fundamental statements required to describe jet blast. Specifically, this included the statements for computing stagnation pressure in the primer jet in terms of time. However, it was necessary to add a factor to show dependence upon distance from the jet exit. Additional information on the jet width was needed which was obtained from a correlation (curve fit) of supersonic jet data from Ref. 5. This is:

$$\frac{y}{r_0} = 1 + \left(\frac{x}{r_0}\right)^{0.815}$$

for the radius to the point where stagnation pressure falls off. (r_0 is the radius of the jet exit, i.e., the primer tube). The expressions used for determining stagnation pressure are:

$$p_{p,x,t,M82} = 2060 \times 10^6 t e^{-0.635 \delta / r_0} \quad \text{for } t < 0.004 \text{ sec.}$$

and

$$p_{p,x,t,M82} = 1.33 \times 10^6 e^{-0.635 \delta / r_0 - 1200t} \quad \text{for } t > 0.004 \text{ sec.}$$

where $p_{p,x,t,M82}$ is the pressure in the primer jet as a function of distance from the flash hole and time for the M82 primer,

δ is standoff, the distance between the spindle and the base pad

and r_0 is the flash tube radius

Then the shear force per unit length of jet circumference is

$$F_t = P_{p,t,MBZ} y/2$$

This is resisted by the shear strength of the pad, i.e., allowable stress times thickness. Note that the resistance or reaction force of the pad is assumed to be the maximum shear force that can be exerted by it. This is realistic because the action of the jet is impulsive and the pad inertia provides the reaction necessary for development of the shear condition. Maximum shear will occur, then, around the edge of the jet, i.e., where its radial pressure gradient is a maximum, as long as the pad is unsupported in the region of jet impact, which is the case for a centrally located primer tube, because the pad bears against the end of the tube, and the inside diameter is greater than the diameter of the jet.

Inputs required by the pad penetration logic are flash tube diameter, called FTUBR, and pad allowable stress, called PDALST. A logical variable, PENET, is initialized as FALSE and when penetration occurs PENET is set to TRUE, and the words "base igniter pad penetrated," are printed. The black powder in the sheared out volume is assumed to be displaced to the first grid containing the center core loading, and porosity for each grid is adjusted accordingly. Of course, the base pad is heated by the primer jet over its whole exposed surface, and base pad penetration can occur as a result of burn through as well as by shear.

Subsequent to base pad penetration, the primer jet heating is applied to the near end of the center core loading in the igniter tube. This is accomplished in the logic for primer jet heating of black powder by using "flags" and the logical variable, PENET, as controls. The procedure insures that primer jet heating is applied only in one axial grid and when the black powder there is consumed a search is made to find the next grid containing black powder. This logic is located in subroutine BPFIR.

5.3 FLAME SPREAD IN THE BASE PAD

Flame spread in a bed of propellant has been studied in the closed bomb and it was found to exhibit characteristics similar to the linear burning rate relation, $r = bp^n$, where p is the chamber, or bomb pressure, Ref. 6. The rate of increase of burning surface area was found to obey the relation, $s = cp^m$ for M1 propellant, for which $m = 2.2$ and $c = 1.45 \times 10^{-5}$ in 2/sec-psi³. For black powder a value for $c = 1.8 \times 10^{-5}$ was first attempted. This was incorporated into the logic for

black powder burning in subroutine PRIMER. But first, the mass generation logic of the primer jet was moved to MAIN so that it would be continuously credited. Then the logical variable BPLEFT was made TRUE in a different location of the sequence for detecting black powder ignition and depletion so that subroutine PRIMER would be called at all times until black powder is depleted, including the time previous to its ignition.

Flame spread is assumed to start with the ignition of black powder in the first axial grid that contains it. Then it proceeds radially outward from the periphery of that axial grid, and computes the change in porosity of the black powder in the $J = 2$ grid and the mass of gas generated. Entry to the base pad flame spread logic is permitted only when the loop (within which it is located) is executed for the axial grid of the base pad, inasmuch as it constitutes the source of flame spread. When the base pad is consumed as a result of the flame spread, a logical variable, BRNOUT, is set true and this ends entry to the flame spread logic. At the same point, porosity of the black powder is set at 1.0.

5.4 PROPELLANT AND PROPELLANT BED MOVEMENT

The basic mathematical model has been demonstrated to simulate gas motion in the gun very effectively. However, its representation of propellant movement is inadequate to treat the problem of wave formation and its propagation back and forth in the chamber of the 155mm gun. The large pressure gradients that have been recorded in this chamber are sufficient to cause vigorous movement of propellant and it is very likely that this movement in turn affects the wave formation and its shape, i.e., pressure distribution. This would especially be the case where bed porosity is low as when the propellant is packed to maximum density. This is exactly the condition that the basic math model was incapable of simulating and yet is so necessary to the study of gun behavior.

A new subroutine has been written and extensively debugged for description of propellant movement. In addition to describing the independent one-dimensional motion of propellant in each grid the new PRPVEL subroutine describes the movement that results when two or more adjacent grids become tightly packed with propellant so that it moves as a single slug. Furthermore, in its new form the subroutine can treat the situation arising when a slug separates into two slugs as a result of a pressure peak occurring within it.

Execution of this subroutine starts with computations for pressure gradient in each grid, $DPDX$. From these values, a total force on the propellant is computed grid by grid. It depends of course, upon propellant porosity, $PHIBG(I,J)$ and is obtained by summing drag force per unit area and the pressure gradient.

After this, a computation sequence is entered to evaluate the acceleration and velocity of propellant in each grid, if it is not packed, i.e., if the porosity is greater than the minimum allowed value. The acceleration is simply force divided by mass. Then, the change in velocity is computed for the computing interval.

If a packed grid is identified, the solution flows to a repetitive sequence where the force and mass of propellant in all adjacent packed grids are summed. This is preparatory to the computation of acceleration and velocity for the slug, which moves as a single mass through many grids, unless it splits into two separate masses because of a pressure peak. In the process of distinguishing this phenomenon, the end grids of each slug of packed grids must be identified. The first packed grid in the search is called $LF1$, and the last one of the first packed group is $LF2$. If the slug separates into two separate groups the first grid of the second group is assigned the name $MF1$, and its last grid is called $MF2$.

In order to determine whether a slug separates, the individual value for the velocity of each grid is compared as soon as it is computed with the average value of velocity for all packed grids that were computed previously for the slug during the looping. If the individual value exceeds the average value, separation is indicated and the computer is routed to another section of the program. This contains logic similar to that for grids $LF1$ to $LF2$, for the purpose of computing acceleration and velocity of grids $MF1$ to $MF2$.

It can be shown by momentum theory that the imposition of the group velocity upon an individual mass within a group of masses in intimate contact is valid unless the individual's velocity is greater and it is located at the forward boundary. This fact has been applied to the propellant bed movement logic by maintaining a running sum for grids packed with propellant, of mass and force on the group which is used to compute group acceleration and updated velocity. When a grid is encountered where propellant velocity is greater than that for the group, separation is indicated as described previously. If not, the logic continues to the last packed grid, updating velocity with each loop. When this is complete the final average velocity is assigned to the propellant in all packed grids.

The same logic is provided for packed grids in a second group if it is identified. Execution of the program using this logic has proven its validity by a printout of propellant velocity from subroutine PROPEL as well as PRPVEL. Subroutine PROPEL actually makes a check on propellant velocity using statements based upon momentum theory. In a variety of computations, the results obtained from PROPEL always have been the same as from PRPVEL, as indicated by the special printout.

5.5 BLACK POWDER MOVEMENT IN THE IGNITER TUBE

Experiments have shown that the igniter core loading may be driven far along the igniter tube by the force of the primer jet, prior to ignition of the black powder (Ref. 1). Of course, a prerequisite for this action is the penetration of the igniter base pad at the breech end of the charge, which is a demonstrated capability of the M82 primer jet. This phenomenon has been modeled as discussed in previous paragraphs. (The M82 primer has approximately 1/2 the energy of the XM119 primer that has been used in some firings of the XM123E2 charge in the 155mm gun). In addition, logic was formulated and incorporated to represent the effect of the primer jet forces upon the igniter core loading.

The stagnation pressure of the primer jet is modified by $P_{Px,t} = r P_{Px,t,M82}$ to permit the representation of primers other than the M82. The quantity r , which is given by the input quantity PRIMCOF, is the ratio of the strength of the primer used relative to that of the M82 primer. Presently, a factor of 2.15, (the weight ratio of black powder content) is being used in lieu of test results for the XM119 primer.

Resistance to movement of the igniter core loading is provided by its inertia but initially it is tied to the igniter tube near its breech end by two ribbons. A strength test of these ribbons revealed that they could be torn from their attachments by a total tensile force of 24 pounds. Therefore, this force was made an input to be used in logic contained in subroutine PRPVEL. In this subroutine, this logic is entered when the center core calculations are made ($J = I$) and the base pad has been penetrated (PENET is true). Then $P_{Px,t}$ is evaluated and converted to a force (BGFCE), which is summed along with the forces of fluid flow for several grids along the igniter core loading. If this sum of forces, called FORCE, at any time exceeds TYFCE a logical variable, BGJAM is set to TRUE, which permits entry to the movement logic for the igniter core loading.

The method for treating the actual movement of black powder in the igniter tube is one of considering the velocity, and therefore, the distance moved during a time interval, to be an average value for all the powder, i.e., it all moves simultaneously at the same rate, until

the forward end of the core loading encounters the base of the projectile. Subsequently, it becomes compressed by the prevailing forces that are induced by gas flow, according to compression characteristics that were determined by test.

After obtaining the sum of the forces caused by the gas flow at all grids, the weight of black powder is summed for all grids. Then the black powder is moved from grid to grid under the control of an indexing system. The intent here was eventually to make the indexing rate a function of the computing interval and an input quantity, to be determined as a result of actual tests of core loading movement in a transparent igniter tube, as done for the 175mm gun model (Ref. 1). Presently, the indexing rate corresponds to a powder velocity of about 200 fps.

After sufficient movement the igniter core loading is stopped by a limitation on NEND, an input variable, called INBP, that represents the grid where the loading is assumed to jam. Next, the variable BGSTP is set TRUE, permitting entry to the powder compression logic. Basically, this logic determines the powder volume, i.e. core loading length after compression, and the porosity in each packed center grid, using the previously determined powder weight. The expression for minimum compressed length is

$$l_{min} = \sum l (1 - 0.03145 (F - 10)^{0.545})$$

where $\sum l$ is the sum of the loading lengths and F is the force. One of the first steps in this logic is to identify the first grid containing powder, IBGNP. The next grid after IBGNP is, of course, the first packed grid, called LLI, because the breech end of the igniter bag is not always located at a grid boundary.

The total deformation by compression is not permitted to occur during only one computing interval after encounter with the limiting grid, INBP. Instead, IBGNP is advanced one grid for each computing interval. A test is made each time this occurs to detect the event when compressed length reaches the minimum compressed length.

5.6 COMPRESSIBLE PROPELLANT BED

The elasticity of a porous granular bed such as that typified by a propellant charge when loaded in compression may be represented as a proportionality between compressive force and the bed solidity, (1- ϕ) instead of bed deflection inasmuch as the two properties are directly related. This fact has been utilized in providing for bed elasticity effects in the force balance on the propellant grains within each grid.

Furthermore, the new logic in subroutine PRPVEL lends itself very well to the accommodation of bed compressibility in this manner because of the individual treatment given each grid.

It was necessary only to add a new pair of terms to the force balance on the propellant that account for the difference in the reactions in each direction caused by the elasticity of propellant in adjoining grids. In addition, the lower limit of bed porosity was reduced to a small fraction of the original porosity, ϕ_0 , i.e., bed porosity was permitted to be as low as $0.2\phi_0$ to represent compaction to very high solidity.

This logic should provide for a better representation of dynamic effects within the propellant bed as gases flow within and through at ever changing conditions. For example, the situation can be envisioned where wave formation in the gases is reinforced by the action of the propellant. Another example is the action that the free ends may exhibit when influenced by the arrival of an impact-induced shock wave.

5.7 PROJECTILE RESISTANCE TO MOVEMENT

The previous logic for describing the variation of projectile resistance to movement through the barrel was inconvenient to implement because considerable preparation of inputs was required. Consequently, it has been changed so that data that are conventionally used to describe the pressure during the projectile may be directly applied as inputs to the program. The conventional format is a plot of pressure vs. distance along the barrel. Four significant points on this plot may be selected that will adequately describe the most important part of the plot, i.e., for the first foot or two of travel. The remainder can be expressed by simple statements that are based upon isentropic compression of the air in the barrel that is ahead of the projectile. The coordinates of the four points of the driving pressure plot constitute the inputs and the logic provides all the rest, i.e., the conversion to force, FDPRIM, for all values of XP the distance traveled. Referring to Figure 10, these points are (PZO, 0) (PDMAX, WOB), (PINT, XINT), and (PLO, XLO). Then, for values of $XP > XLO$ the pressure is computed from the expression:

$$\Delta p = \gamma p \Delta u / (c - (\frac{\gamma+1}{4}) \Delta u)$$

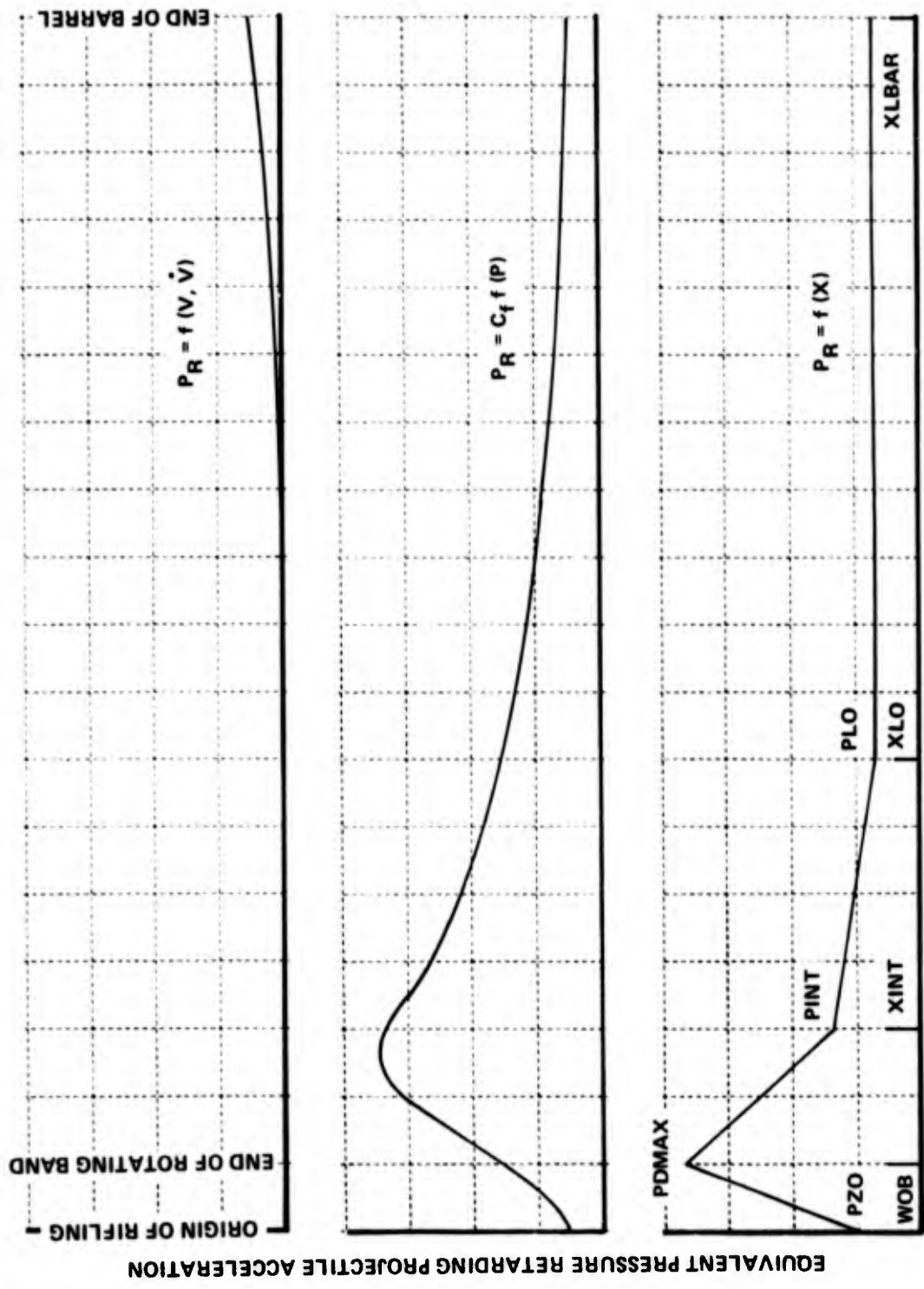


Figure 10 BARREL RESISTANCE FUNCTIONS BUILT INTO THE 155 mm HOWITZER CODE

The value of ΔU used, i.e., the change in projectile velocity, is obtained from the previous computation. For Air, $\gamma = 1.4$.

5.8 NEW DRAG EXPRESSION

A more nearly complete expression for drag has been incorporated that is useful because it is valid for any particle shape of the porous bed. The expression was obtained from Perry's Chemical Engineers' Handbook, pg. 5-53, Ref. 4, and has been verified for M30 propellant by experiment as previously discussed:

$$\frac{\Delta p}{\rho} = \left[\frac{4 f_m (1 - \phi)^{3-n}}{\phi_s^{3-n} \phi^3} \right] \left(\frac{L}{D_p} \right) \left(\frac{V^2}{2g} \right)$$

where Δp is the pressure drop, psf
 ρ is the fluid density, lbs/ft³
 f_m is the friction factor
 ϕ is the porosity
 ϕ_s is the shape factor, equal to the area of a sphere having the same volume as the particle divided by the particle surface area
 n is the function of the Reynolds number approximately equal to 2 for $Re > 3000$ based upon $Re = \rho V D_p / \mu$
 μ is fluid viscosity
 V is the superficial velocity, = ϕ times actual velocity, fps
 D_p is the diameter of a sphere having the same volume as the particle

6.0 SENSITIVITY STUDIES CONDUCTED WITH THE COMPUTER PROGRAM

After exercising the computer program thoroughly to eliminate "bugs" it was employed in a sensitivity analysis of a number of the variables associated with the XM123 propelling charge and its loading configuration. These variables include; the chamber length, which could be established by the distance the projectile is rammed into the barrel; the loading density or degree of initial compaction, which is defined in the program by chamber diameter and igniter tube length for a given charge weight; the standoff, i.e., the distance separating the breech from the end of the charge; the propellant burn rate at the low pressure end of the range for which existing data is of somewhat dubious value; the friction factor of the propellant bed which has been measured but whose significance has been little understood; the shot start value, i.e., the engraving force on the projectile; the primer strength; and the initial temperature of the propellant.

A large part of the study was devoted to the ammunition configuration in which ignition is accomplished wholly by an igniter base pad to identify the significance of each variable to the development of pressure waves in the chamber. A few computer runs were made with the igniter base pad located at the barrel end of the charge. Finally, the input combinations that produced the most interesting effects were submitted and the complete center core igniter logic was invoked. The center core igniter exercises provide a variety of situations for examining the consequences of using the strong primer, i.e., the XM119 primer, inasmuch as it has not yet been demonstrated that this primer is incapable of dislocating the center core loading before its burning is well under way. It is easy to speculate that if the XM119 primer does have such a capability, the manner of ignition of the propellant charge must be abnormal, to say the least.

A description of the input conditions and configuration of each computer exercise, or run, is given by data in Table III. Each is also identified by a run number for use in the following discussion. Charge weight was 25.5 lb. for all cases.

The first run, R-1, represented a loosely compacted charge in a moderate length chamber with breech end ignition and zero standoff. Propellant minimum burning rate was 0.36 in/sec up to 600 psi, chamber pressure. Pressure throughout the chamber rose uniformly, climbing steadily to a peak value of 61000 psi without any unusual effects.

For the next run, R-2, all conditions were similar to those of R-1 except that initial charge compaction was increased to a moderate level. The effect of this was to produce an unsteady rise in pressure such that at any given time the pressure at each point along the chamber

Table III
INPUT CONDITIONS AND RESULTS OF COMPUTER EXERCISES

RUN NO.	*IGNITER TUBE LENGTH - in.	CHAMBER LENGTH - in.	STANDOFF in.	BED FRICTION FACTOR	BURNING RATE OF PROPELLANT (SEE NOTES)	SHOT START PRESSURE - psi	PROPELLANT INITIAL TEMPERATURE - °F	PRIMER STRENGTH COEFFICIENT	PEAK CHAMBER PRESSURE - psi	MAX. POSITIVE DIFFERENCE IN PRESSURE AT PROJECTILE MINUS BREACH - psi	REMARKS (SEE NOTES)
R-1	27.5	32.5	0	3.0	HIGH ¹	10,000	70	1.0	61000	0	
R-2	25.0	32.5	0	3.0					61300	8000	
R-3	22.0	32.5	0	3.0					57300	2000	
R-4	25.0	34.0	0	3.0					47000	1100	
R-5		30.0	1.5	20.0	MEDIUM ²				80000	8000	
R-6		32.5	0	3.0	MEDIUM ²				80000*	8000	
R-7			0	3.0	LOW ³				72500	11000	
R-8			0	3.0	MEDIUM ²				64700	10000	
R-9			0	3.0	MEDIUM ²				58300	4700	
R-10			1.5	3.0					57700	5000	
R-11			1.5	3.0					53500	4300	
R-12			0	3.0		20,000			76700	4800	
R-13			1.5	3.0		2,700			53800	4600	
R-14			0	3.0		10,000	40				
R-15			0	0.2			340		58000	5000	
R-16			0	3.0			70		53000	1200	
R-17			0	3.0					58000	5500	
B-1			0	3.0	HIGH ¹				72100	17500	CS
C-1			0	3.0				2.15	7	4600	
C-2			1.5	3.0				1.0	57000	0	
C-3		30.0	1.5	3.0					62500	1000	
C-4		30.0	0	3.0					62400	2700	
C-5		34.0	1.5	3.0	MEDIUM ²				48000	1500	
C-6		30.0	0	3.0					62500	2400	FIT
C-7		30.0	0	3.0					60500	4300	DLI

NOTES:

¹ HIGH BURNING RATE INDICATES A MINIMUM RATE = 0.36 in./sec

² MEDIUM BURNING RATE INDICATES A RATE THAT CONFORMS VERY CLOSELY TO THE CLOSED BOMB RATE FOR M30 OVER THE WHOLE PRESSURE RANGE

³ LOW BURNING RATE MEANS THAT BGEN = .005

CS - CRUSHING AND BREAKUP OF PROPELLANT WHEN COMPRESSED

FIT - FLAWED IGNITER TUBE

DLI - DENSELY LOADED IGNITER

* IGNITER TUBE LENGTH IS AN INDICATION OF THE INITIAL COMPACTION OF PROPELLANT

rose at a different rate compared to others and this rate was variable with time. In fact, the pressure decreased briefly on a couple of occasions before peaking. The peak value indicated was 61300 psi at the breech end. This peculiar behavior of chamber pressure history is interpreted to indicate the existence of pressure waves, which is confirmed by the regular progress of the pressure variations along the chamber and their reflection at the ends. The character of these waves can be appreciated by examining the history of the difference between pressures at the projectile end and the breech end of the chamber. This history is plotted in Figure 11 for run R-2. It is similar to the plot obtained from firing test data (Figure 12), except that initially the pressure drops to a much greater negative value than for test. (The time scale cannot be compared on an absolute basis because no attempt was made to duplicate the early ignition mechanisms.) The compaction of the propellant bed apparently promotes local pressure buildup, which leads to the formation of waves.

Run R-3 represents a highly compacted bed, which was obtained at the expense of increased gap between the end of the propelling charge and the projectile. Pressure waves were obtained but they were greatly reduced in intensity. The first positive excursion in end pressure differences peaked at only 2900 psi compared to the 6800 of Figure 11 for Run R-2. A possible explanation for this is that the large gap strongly attenuates the waves.

When the conditions of R-2 were repeated except that a longer chamber was input making the gap even greater, run R-4, the waves disappeared completely (in the sense that the projectile base pressure never exceeded the breech pressure), and the peak pressure reached was 47000 psi, which tends to support the speculation that pressure waves are attenuated by a large gap. In addition, it demonstrates the effect of chamber volume relative to charge volume, which was too low for R-1, causing excessive peak pressure.

Run R-5 was similar to R-4 except that the chamber length was reduced to a minimum. Even so the gap was 4.5 inches (allowing for a 1/2 inch thick base igniter pad). Again, the result was very strong waves. Chamber pressure histories are plotted in Figure 13, which shows the interrupted pressure rise and how exaggerated it becomes at the projectile end.

All previous runs had been conducted for a zero standoff so the effect of increasing this variable was studied. For the first run (R-6), all inputs were at moderate values (standoff = 1.5) except that the .36 in/sec minimum burn rate was retained. This made the run the same as Run R-2 except for standoff. Waves that were almost as strong

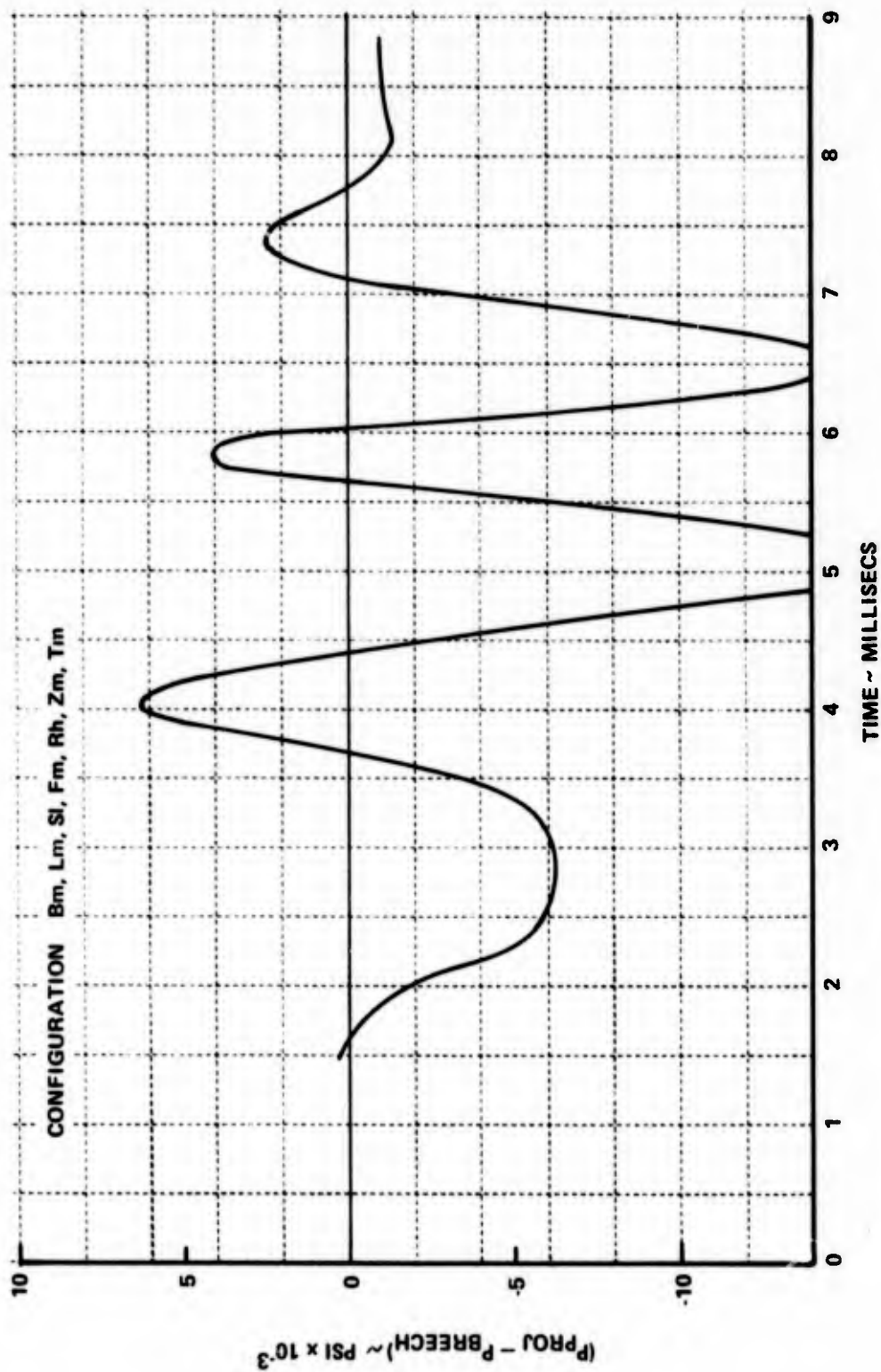


Figure 11 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-2

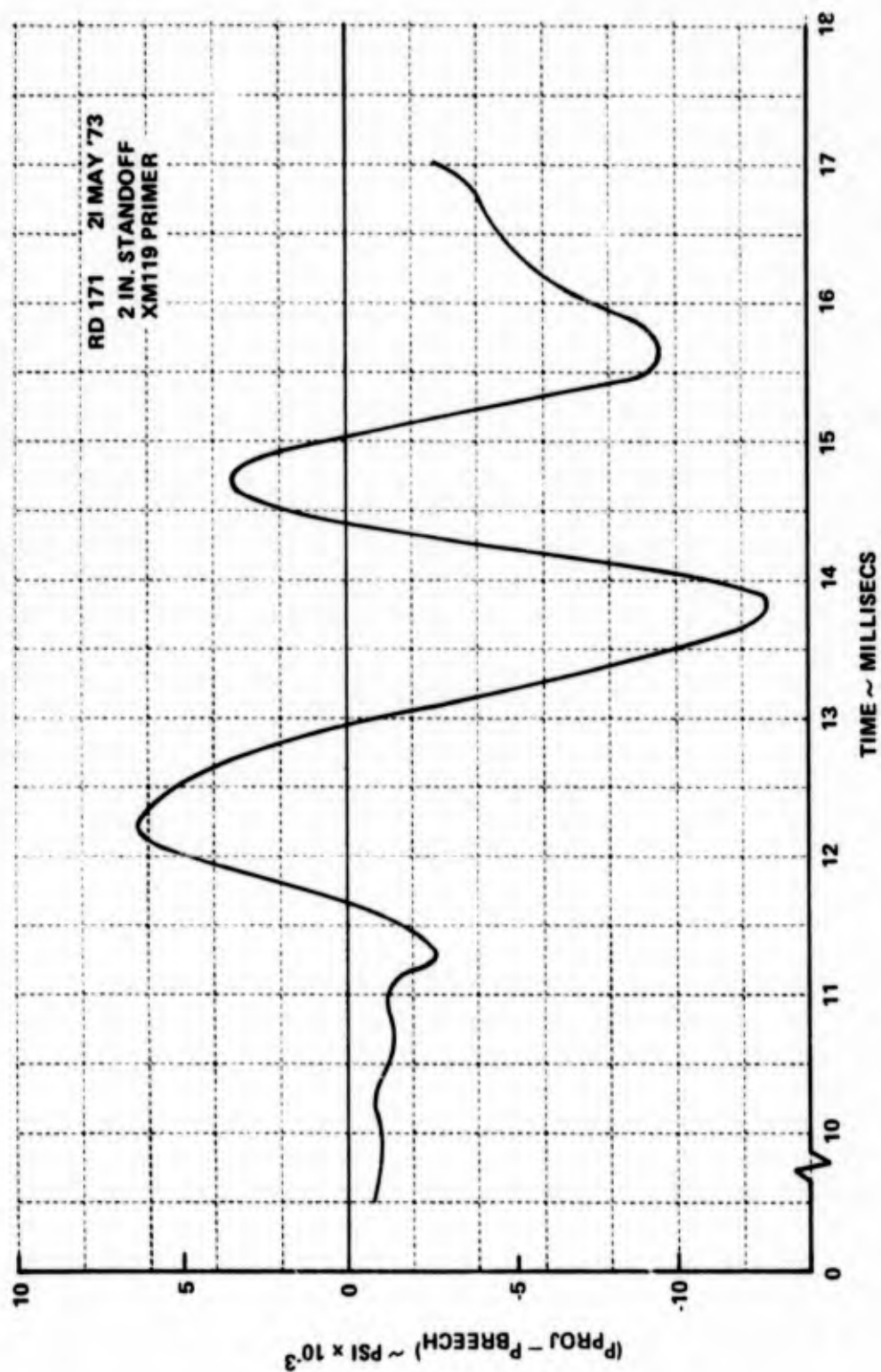


Figure 12 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ FIRING DATA

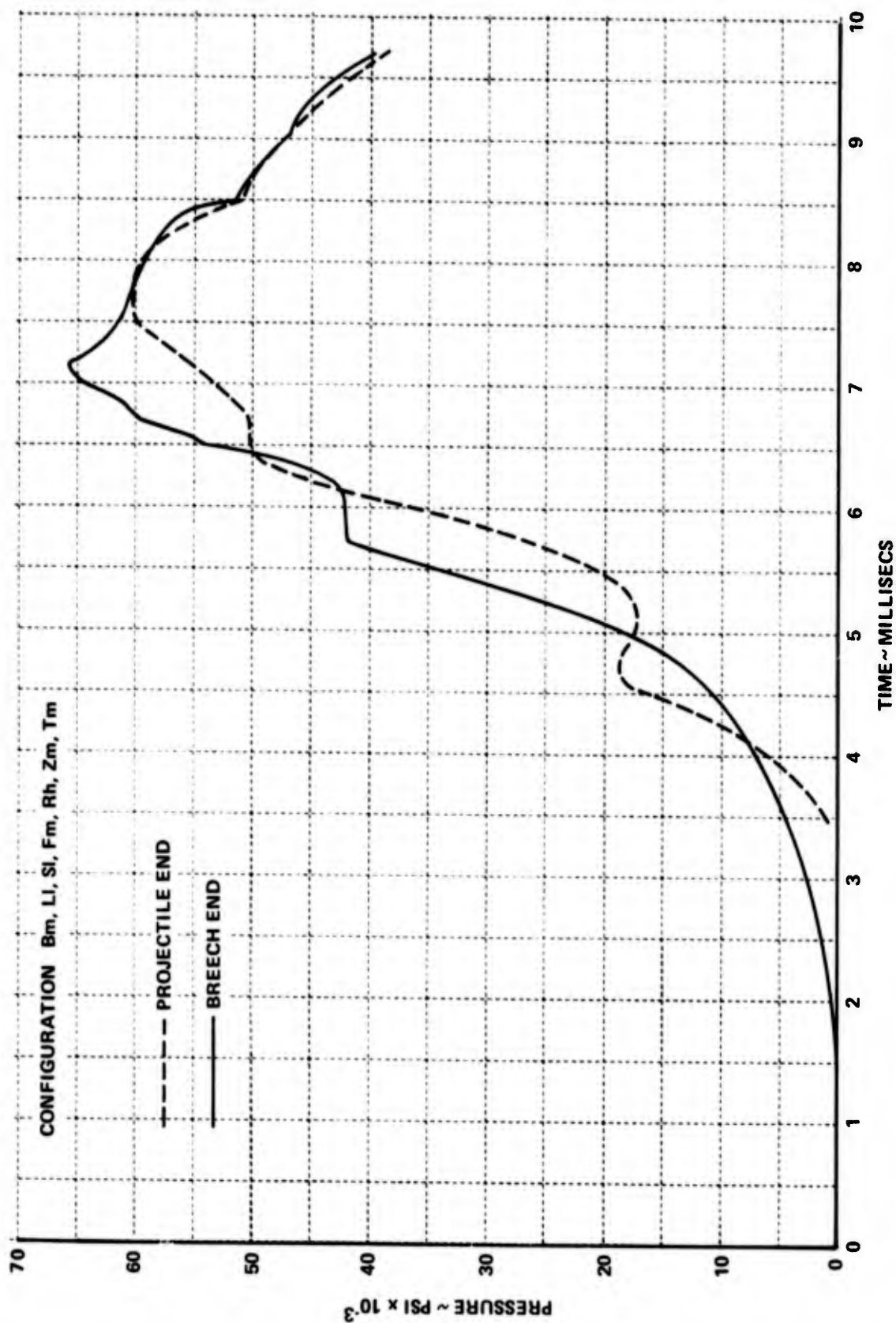


Figure 13 CHAMBER PRESSURE HISTORIES ~ RUN R-5

as for R-2 were obtained, see Fig. 14. Furthermore, the nature of this plot is very much like that obtained from the firing test data (Figure 11). Chamber pressure histories for each end are plotted in Figure 15.

A further increase in standoff placed the propellant charge much closer to the projectile than the breech for Run R-7. Gap was 1.0 inch while standoff was 6.0 inches. This caused even stronger wave action than previously with the projectile end pressure going above breech pressure when they were only 1540 psi. The first positive excursion peaked at a difference of 11800 psi when the projectile end pressure was 15000 psi. The highest peak chamber pressure was recorded also for this run - a value of 72500 psi. Our interpretation of this run is that if a strong ignition, which means that a large amount of propellant is ignited suddenly, occurs at the breech end of the propellant charge, waves will be present regardless of the standoff. However, increased standoff reduces the probability of a strong ignition.

A run was conducted for very high bed friction factor. Other conditions for run R-8 were the same as R-1, which produced no wave action although peak pressure was excessive because the charge was confined. Vigorous wave action resulted for R-8 demonstrating that friction of the bed and excessive compression are related conditions that promote wave formation. The effect of increasing initial bed compaction was nearly the same (run R-2) although this cannot be construed as an independent cause of wave action inasmuch as the compaction must be maintained for a period of time to be effective in promoting wave formation.

In order to determine whether the high initial value of burning rate contributes to the formation of waves, Run R-2 was repeated with a much lower value of burning rate as Run R-9. The result was that the wave action decreased slightly in amplitude, and it was very much like that of run R-6. Peak pressure was 59,300. The next run, R-10 was conducted with the same inputs as R-6 except for the moderate initial burning rate, (increased standoff compared to run R-9). Waves were even more vigorous than for either run R-9 or R-6. The history of chamber end pressure difference is plotted in Figure 16. Obviously, propellant burn rate at low pressure is not a critical factor. This was confirmed by Run R-11, which was the same as R-6 except that $BGEN = .005$ instead of $.0057$. Its effect was to delay everything slightly and to produce a flatter, lower peak, see Figure 17.

Run R-12 was then made with a high shot start pressure. Other inputs were the same as for run R-9. Wave action was very similar to that for run R-9 with end pressure histories crossing and recrossing at nearly the same pressure levels. However, peak chamber pressure went to 76700 psi as a direct result of increasing the projectile resistance to movement.

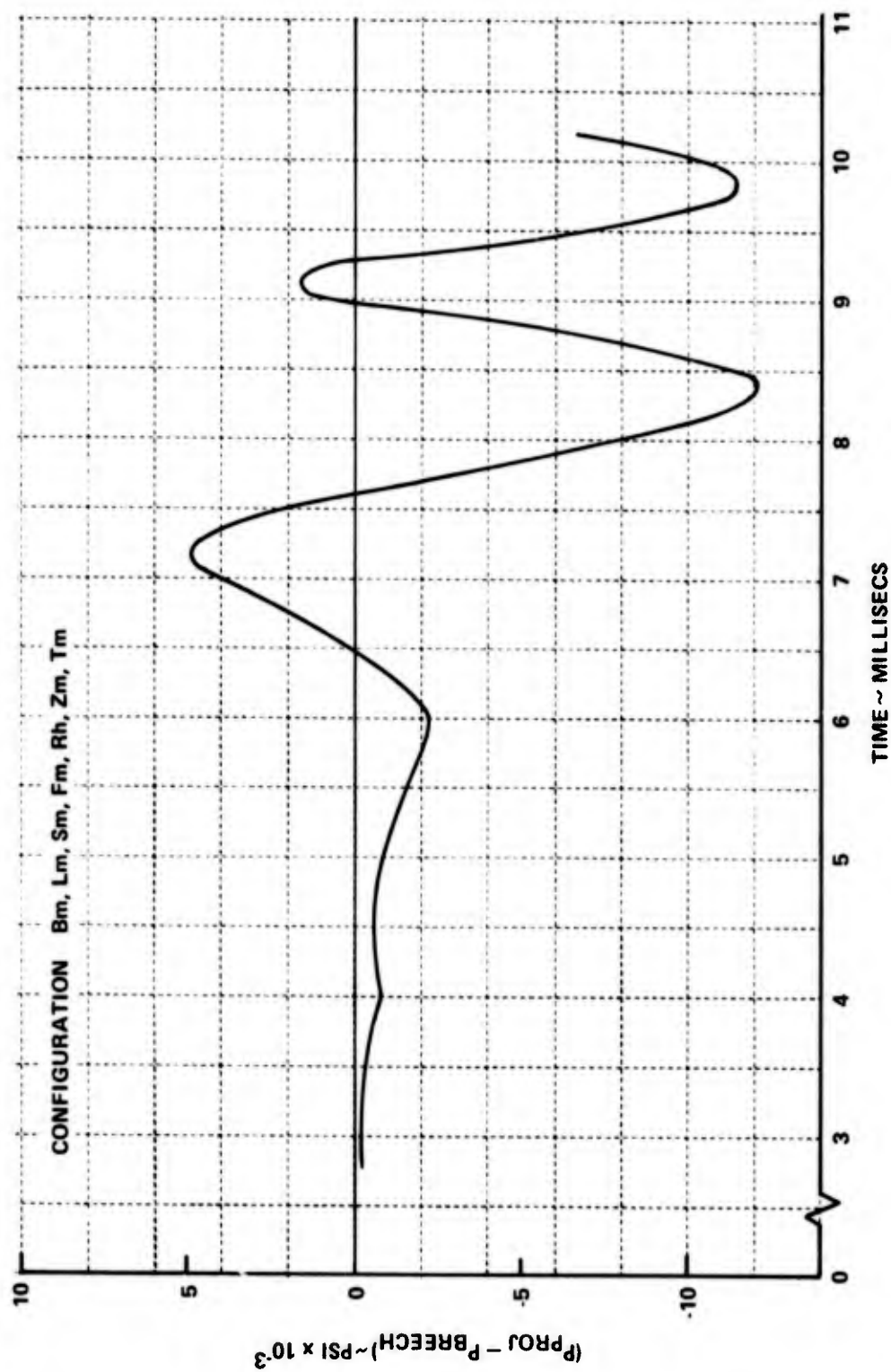


Figure 14 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-6

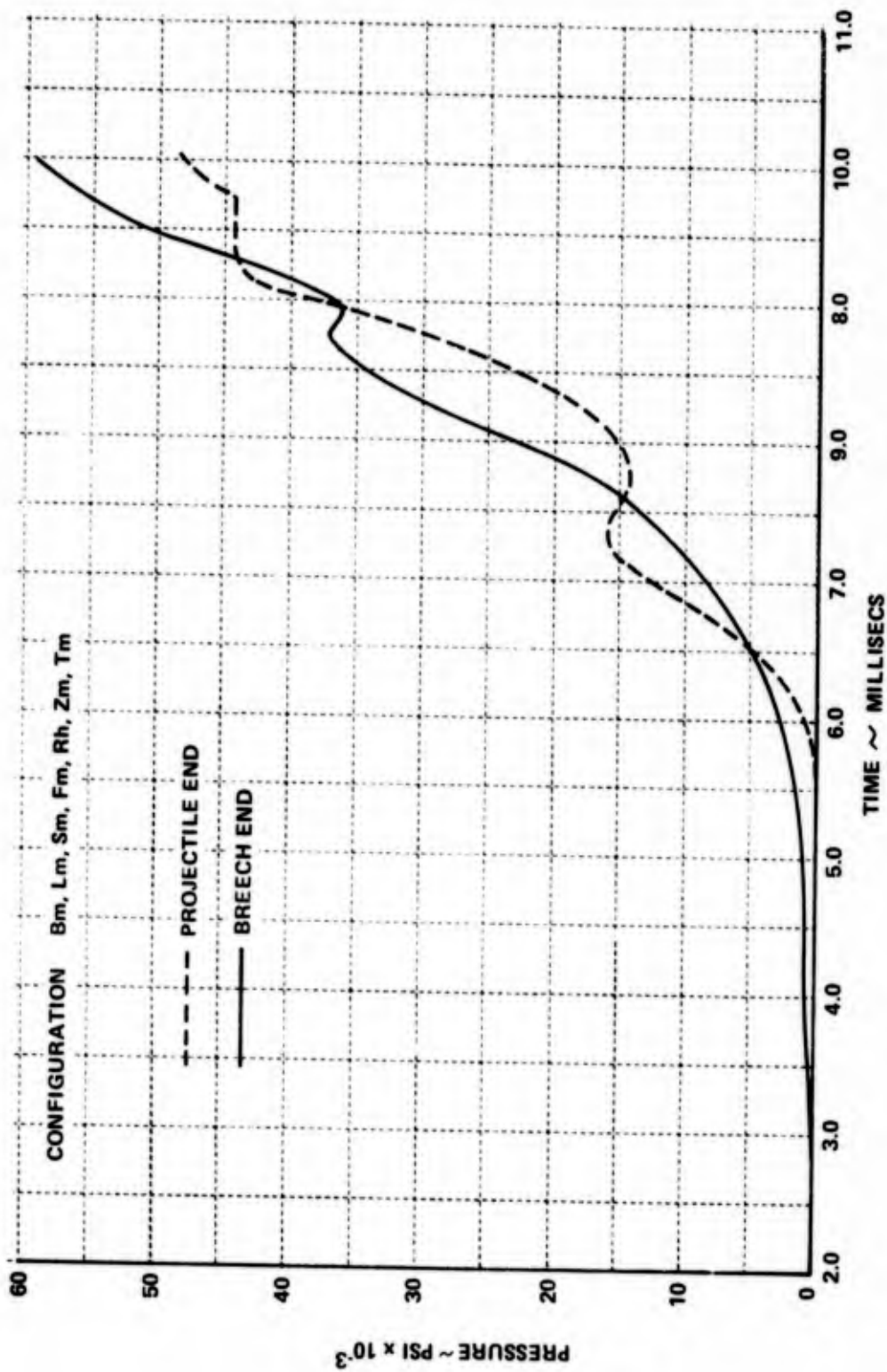


Figure 15 CHAMBER PRESSURE HISTORIES ~ RUN R-6

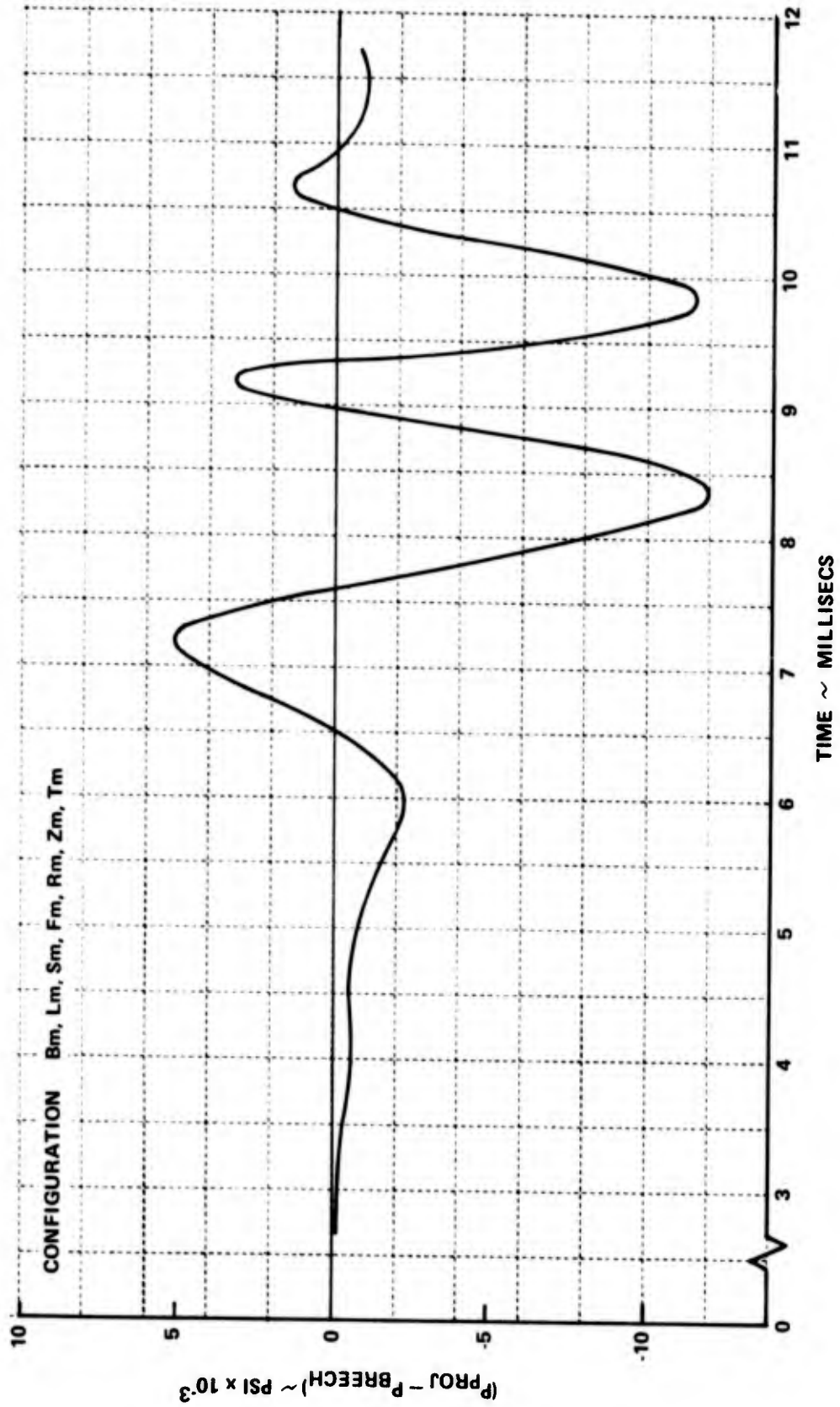


Figure 16 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-10

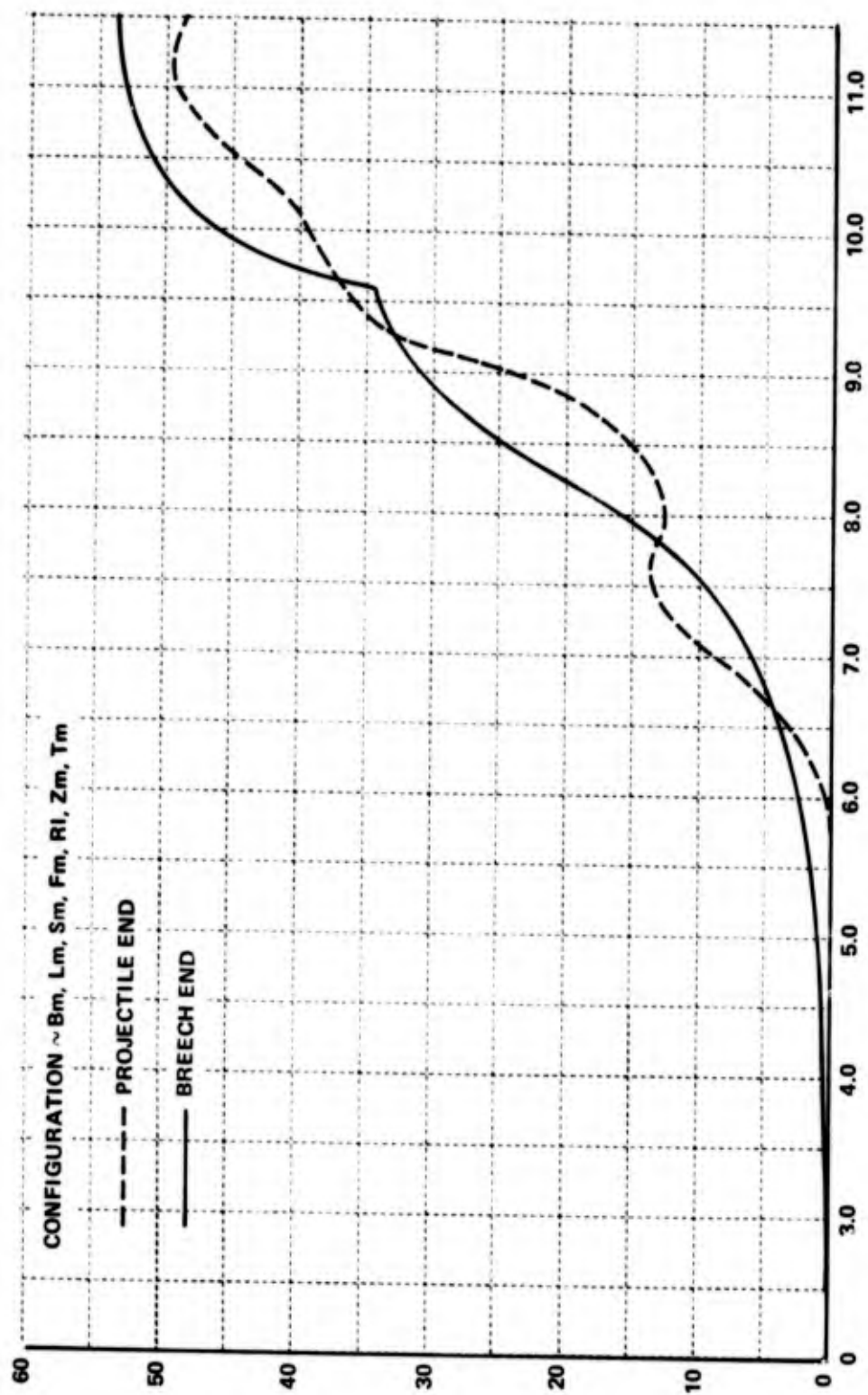


Figure 17 CHAMBER PRESSURE HISTORIES ~ RUN R-11

The effect of initial propellant temperature was studied by means of the next two runs, R-14 and R-15. Run R-14, at the low temperature level, was carried out to 12 millisecs and no propellant ignited so efforts were discontinued. Run R-15 exhibited an end pressure difference peak of 5000 psi when projectile pressure was at 17000 psi and at 3.86 millisecs but the second positive excursion of pressure difference occurred at a very high chamber pressure level. In fact it was near the peaking point for chamber pressure at 7.4 millisecs. Compared with run R-6 it can be appreciated that action was considerably accelerated by the high initial temperature although the development of the second wave was not sufficient in strength to raise it beyond the breech pressure.

Run R-16 was conducted to complete the series on the variation of shot start pressure, which was only 2700 psi for this case. Very moderate wave activity was manifested and chamber pressure peaked at 53000 psi. Wave action was similar to that of run R-9.

The final two runs of this series were made to investigate the effect of grain breakup due to impact at the ends of the chamber and to crushing by compression of the propellant bed. This was formulated temporarily into the logic in subroutine PROPEL by operating on XL(I,J). For the first it was stipulated that whenever velocity of propellant entering the end grids exceeded 20 fps, that entering was broken into 100 pieces from each grain. Results of this run, R-17, showed a slightly greater rate of pressure rise compared to run R-6, but otherwise results were similar. For the other run, XL(I,J) was diminished by 5 percent for each computing interval during the time and wherever the propellant bed was compacted to less than 0.9 PHIO. This produced an excessive rate of pressure rise that was unreal, leading to conditions that the program could not accommodate. Although the peak pressure was not obtained results served to demonstrate that crumbling of the propellant is very significant to performance and could be a critical factor in simulation.

The run that represents end ignition by a base pad at the barrel end of the charge is labeled B-1 in Table III. It was computed for near-zero standoff to study the effect of reflection of pressure waves at the breech. This extreme configuration was expected to yield extreme results, which were fully realized, in the form of very strong waves. The first peak of end pressure difference for Run B-1 (after propellant ignition) reached a positive value of 17,500 psi. see Figure 18. Projectile base pressure at this point showed a peak of 36100 psi and dropped thereafter to a low of 19,840 psi before climbing again to later exceed breech pressure by 14200 psi. Maximum breech pressure reached was 72100 psi.

One object of conducting run B-1 was to provide data for comparison with the computation for the charge with a full center-core

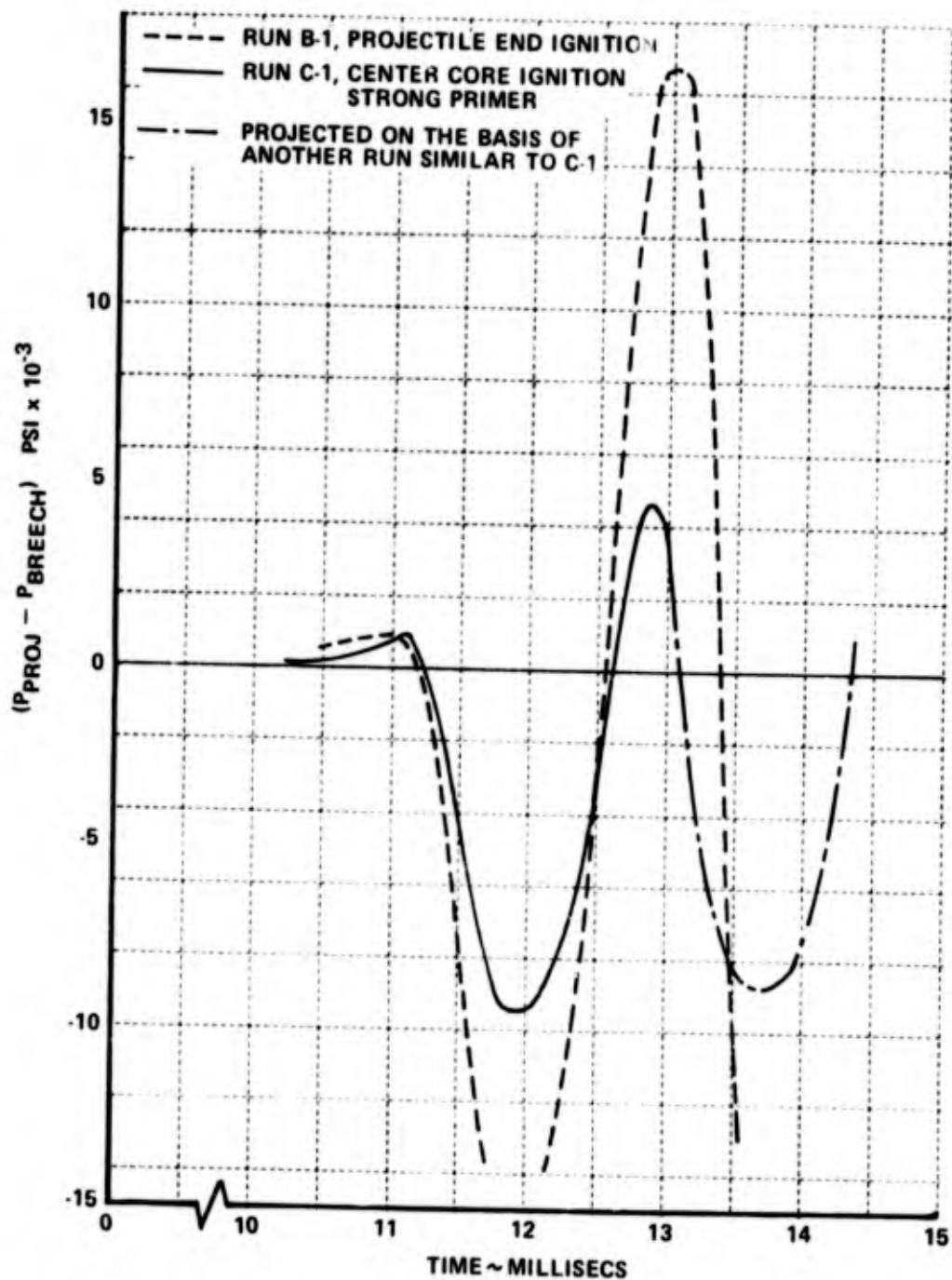


Figure 18 COMPARISON OF CHAMBER PRESSURE WAVES FOR END IGNITION AND ABNORMAL CENTER CORE IGNITION

igniter, in particular with the configuration using the strong primer where the bag ribbons fail and the center core loading is pushed down the tube and jammed at the end. This run is identified as C-1 and is the first of a series of center core igniter runs. It resulted in propellant ignition starting at the barrel end of the charge and progressing rapidly towards the breech so that ignition was complete by 1.3 milliseconds after it started. (The sequence lasted 3 milliseconds for run B-1.) Initially, the chamber pressure at the projectile was pushed up to a peak of 1430 psi, 840 psi in excess of breech pressure as a result of black powder burning. However, this situation was not sustained by burning of propellant. The pressure at the projectile end did not again exceed breech pressure until it rose above 15500 psi, whereupon it reached a pressure difference of 4500 psi. This run is also plotted in Figure 18 where it can be contrasted with the results of run B-1. Frequency of the excursions for both runs is roughly the same but the peak amplitudes are obviously far different. One possible explanation for this is that the igniter tube of the center core configuration acts as a relief device to prevent extreme longitudinal gradient in pressure from developing. Even though tube collapse interrupts the suppression of pressure wave development, this effect is manifested beyond the time of tube failure. It is obvious that suppression is no longer needed after the rate of chamber pressure rise reaches a maximum.

Run C-2 is a center core igniter run for a configuration that produced vigorous waves with end ignition (same configuration as for run R-2). Run C-2 used the fixed center core loading and weak primer. Propellant ignited in an orderly progression from the breech end to the projectile end. This sequence lasted only 1.5 milliseconds although it was 1.2 milliseconds for run R-2. Peak chamber pressure was 57000 psi but no waves developed.

Run C-3 was conducted with the same inputs as Run C-2 except that chamber length was reduced to 30.0. Ignition required about the same length of time as for run C-2. Wave action apparently started but never became substantial. Chamber pressure at projectile was in excess of breech pressure for a considerable length of time (centered at the 6000 psi level) but their difference never exceeded 1000 psi.

Inputs for Run C-4 were the same as for C-3, and a fixed center core igniter again, except that standoff was reduced to 0. Definite wave forms were initially exhibited ($\Delta P = 2700$ psi at 13000 psi chamber pressure) but the second pulse did not materialize.

Run C-5 might be considered a standard configuration-fixed center core igniter, and chamber length = 34.0 in. with everything else moderate. Peak pressure was 46000 psi.

Run C-6 represents a deliberate attempt to produce waves in the chamber by introducing a flawed igniter tube into the charge configuration. This was done by assigning large values of radius to pseudo holes, one in each of the first four whole grids at the breech end. The total grid length affected was 3 in. In addition, the other inputs were conducive to wave formation for the end ignition case (same as for Run R-5). Results showed that ignition was very rapid, requiring only 1 millisecond to progress from the breech end to the other. (In R-5 it required 1.7 milliseconds.) Wave formation was initiated, (the first Δp peak was 2400 psi) but it quickly damped out and chamber pressure distribution assumed the usual character with the breech pressure maintained at a higher level than at the other end. Run C-6 was repeated with low strength values assigned to the tube material in an effort to simulate overall early failure which can be expected to promote wave formation. (PHOOP = PCOMP = 100.) However, the failure did not occur sufficiently early to have any great effect. Of course, to carry this concept much further in reality would be to reach the end ignition situation which would practically constitute a repeat of run R-5, because of the considerable ignition delay observed with the center core loading, i.e., drastic tube failure starting near the breech end would cause restriction of flow through the center core region so that ignition would move progressively from the breech end through the propellant in the same manner as it does using only a base pad igniter.

Run C-7 represents another deliberate attempt to induce wave formation in the gun chamber. This simulates the folded igniter bag such that all the black powder is jammed into half the tube length at the breech end, thereby restricting pressure relief between chamber ends that the igniter tube normally provides. Results revealed that the chamber end pressure difference reached a peak at the point where projectile end pressure was 16600 psi and exceeded breech pressure by 4300 psi. The difference went to zero at the 17000 psi level and to a negative peak of 6100 psi as breech pressure rose to 40500 psi. Projectile end pressure equalled the breech pressure at 46300 but never again exceeded it. Pressure peaked at 60,500 psi.

7.0 CONCLUSIONS AND RECOMMENDATIONS

In addition to numerous conclusions derived from the computer results of the sensitivity analysis, some very interesting conclusions have been reached by observation during the formulation of the mathematical model and as a result of the various experiments that were conducted in support of this formulation.

1. Sensitivity analysis showed that the center core igniter of the 155mm XM123 propelling charge promotes a desirable, near-uniform chamber pressure distribution by providing pressure relief between the ends of the chamber. If the tube remains intact and severe flow restriction does not occur until chamber pressure rise rate approaches its maximum, this uniformity of pressure distribution is practically assured.

2. If the center core igniter tube remains intact during the early period of propellant burning, initial conditions such as standoff, loading density and propellant temperature will have little effect upon chamber pressure history and its distribution according to the results of the sensitivity analysis.

3. The mathematical model showed that if the center core igniter tube fails before chamber pressure exceeds approximately 2000 psi, drastic pressure disturbances will result depending upon initial conditions of standoff and loading density.

4. It was also demonstrated that if the center core loading becomes jammed in the igniter tube either by deliberately loading it that way or as a result of strong primer action, pressure maldistribution will result. Its severity depends upon the degree of tube flow restriction introduced and the initial conditions of the propelling charge.

5. Initial conditions that promote pressure maldistribution, given an igniter abnormality such as those already cited, are very large or very small standoff, reduced chamber length, high loading density, high shot start resistance and over-strong primer action. In other words these conditions can aggravate an unfavorable XM123 ammunition condition but each one, by itself, will not induce maldistribution of chamber pressure, given an intact igniter tube.

6. A propelling charge configuration using end ignition exhibits performance in the 155mm gun chamber that is very sensitive to the initial conditions of standoff, chamber length, and loading density, as well as to their interaction. Zero standoff does not cause maldistributions if the air space at the other end of the charge is over several inches in length. On the other hand, if this length

of space is divided by centering the charge in the chamber having the same length, pressure waves could be severe.

7. Computer results show that propellant burn rate affects chamber pressure level and wave activity somewhat; increased rate being an aggravation given other conditions conducive to the development of waves. Crushing of the propellant, such as might be caused by bed compression, can result in increased burning rate and excessive pressures.

8. Results obtained from computer exercises, which represent a variety of propellant charge configurations, generally indicate that the more severe chamber pressure waves of the firing test results were caused by abnormalities of the propelling charge.

9. Ignition sequence experiments revealed that the presence of the base igniter pad causes a manifold increase in ignition delay of the center core loading. Furthermore, the igniter pad was demonstrated to be unnecessary to center core ignition at room temperature given the proper alignment. Computer code results show that propellant ignition by the center core loading is at least as rapid or more so than ignition started at either end of the propellant charge. These results should be useful to any effort to redesign the center core igniter for improved effectiveness.

Additional work on the computer program could produce improved simulation and effectiveness in its use as a design aid. It is recommended that the computation be accelerated temporarily during the early period before center core ignition. This might be done by a combination of procedures including either the lengthening of computing interval or providing another interval for primer action and black powder heatup, combined with an intermittent use of the finite differencing subroutines. The present computer code requires the use of artifices to avoid long executing time such as initialization of black powder to ignition temperature in selected grids. The recommended improvements would permit more effective simulation of ignition delay in all its phases.

A better simulation would result from the representation of black powder movement. This could be implemented by modifying the propellant movement subroutines to accept the black powder variables. The effect of this improvement would be to provide for the spread of burning powder grains as their ignition propagates and pressure gradients develop.

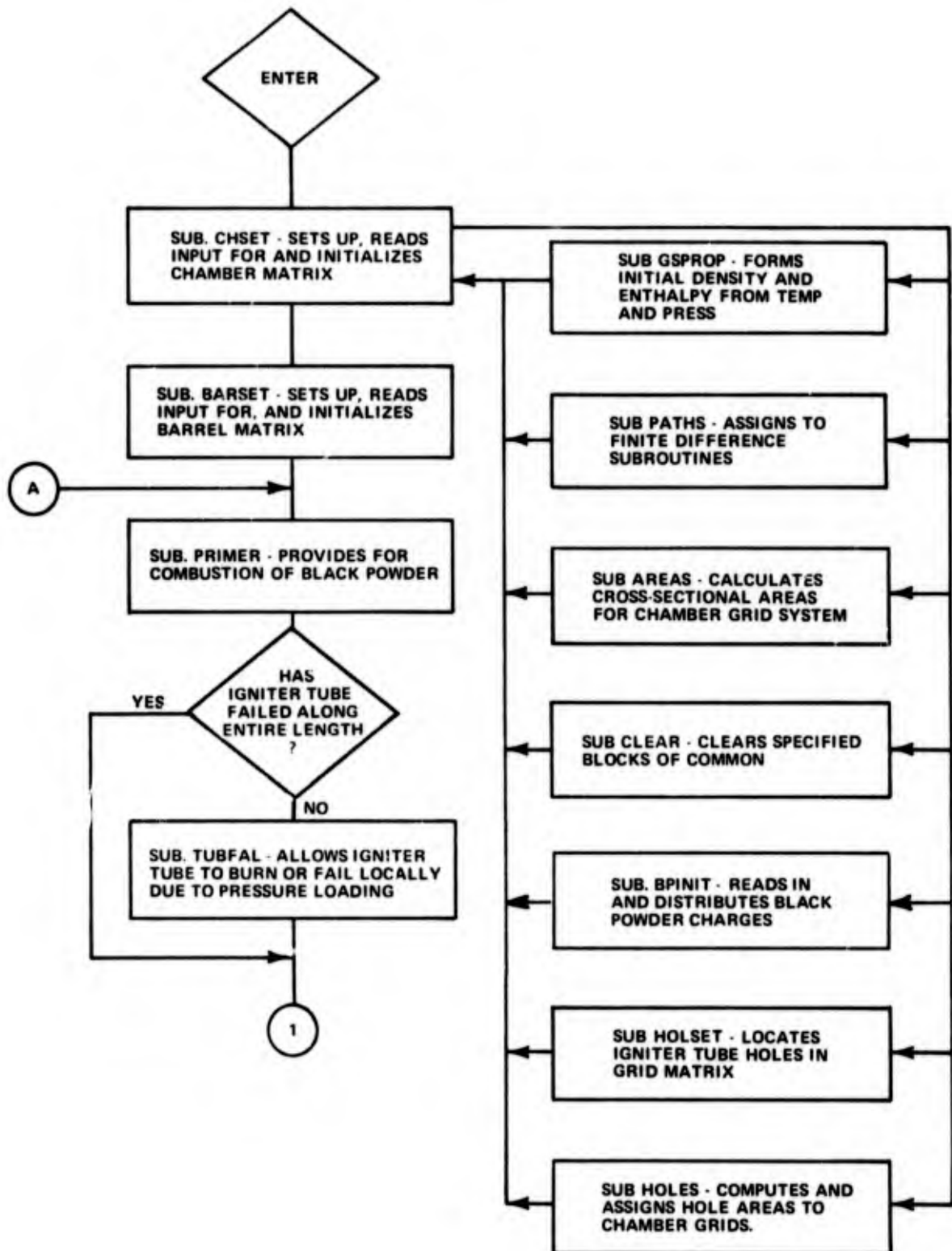
Propellant bed compression tests are recommended. These would provide data on the modulus of elasticity and the characteristics of the quasi-plastic flow regime, i.e., the state of no recovery, where actual crushing of propellant occurs. In addition, it would be useful to measure bed friction at this compacted, crushed condition. The required fixtures already exist, inasmuch as the test section of the propellant bed friction test apparatus was designed for such a contingency. It

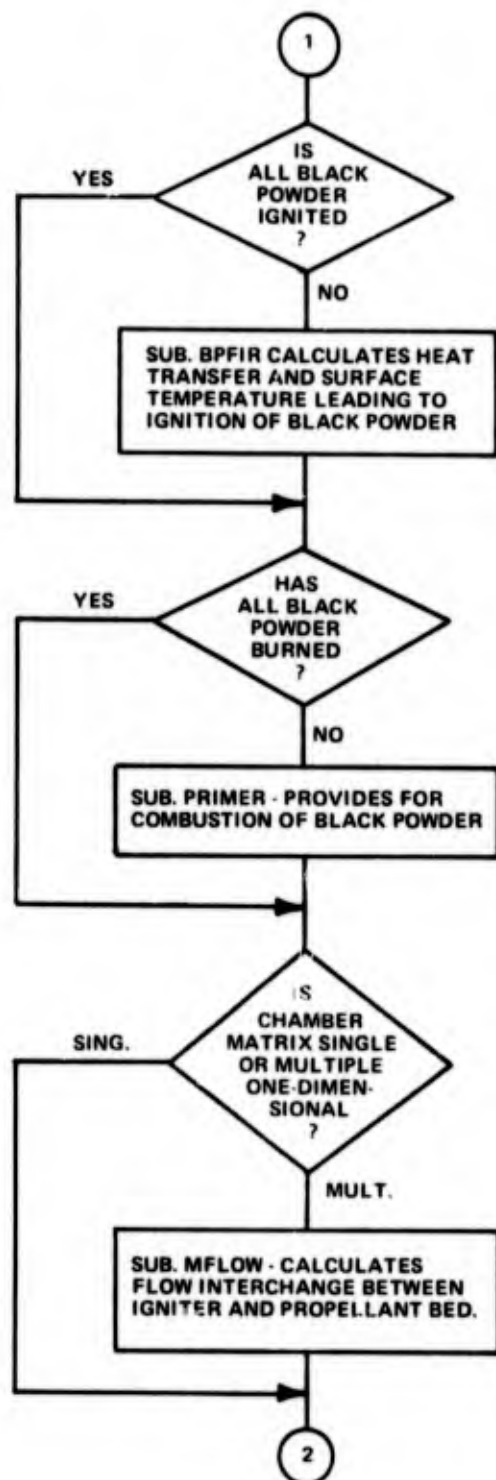
has sufficiently thick walls that it can contain the bed for compression tests and without disturbing the compacted bed it can be moved to the friction test apparatus for air flow tests.

8.0 REFERENCES

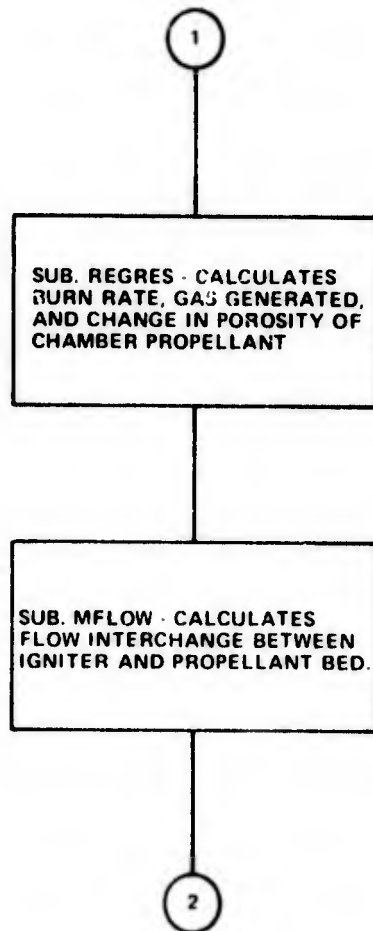
1. Fisher, E.B., and Trippe, A.P., "Mathematical Model of Center Core Ignition in the 175mm Gun," Calspan Report No. VQ-5163-D-2, Picatinny Arsenal, Contract No. DAAA21-72-C-0577, March 1974.
2. Fisher, E.B., and Trippe, A.P., "Development of a Basis for Acceptance of Continuously Produced Propellant," Calspan Report No. VQ-5163-D-1, Picatinny Arsenal, Contract No. DAAA21-72-C-0577, November 1973.
3. Cunningham, R.G., "Orifice Meters with Supercritical Compressible Flow," ASME Trans., July 1951.
4. Chemical Engineers' Handbook, Robert H. Perry, ed. Fifth Edition, McGraw-Hill Book Co., 1973.
5. Love, E.S., et. al., "Experimental and Theoretical Studies of Axisymmetric Free Jets," NASA TR R-6, 1959.
6. Jakus, Karl, "Feasibility of a Laboratory Ignition Test for Lot Acceptance of Gun Propellants," U.S. Army Ammunition Systems Reliability and Safety Division, Picatinny Arsenal Report No. ASRSD-QA-A-P-59-74, May 1974.

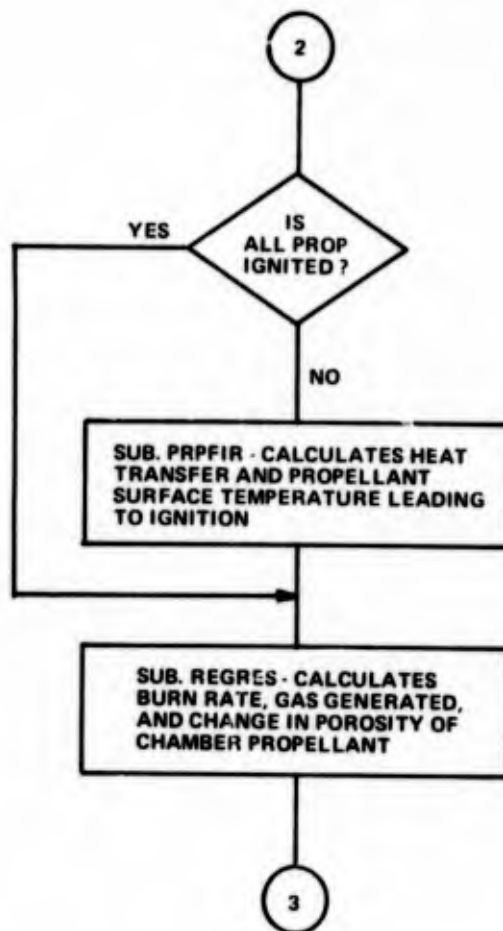
APPENDIX A PROGRAM FLOW CHART



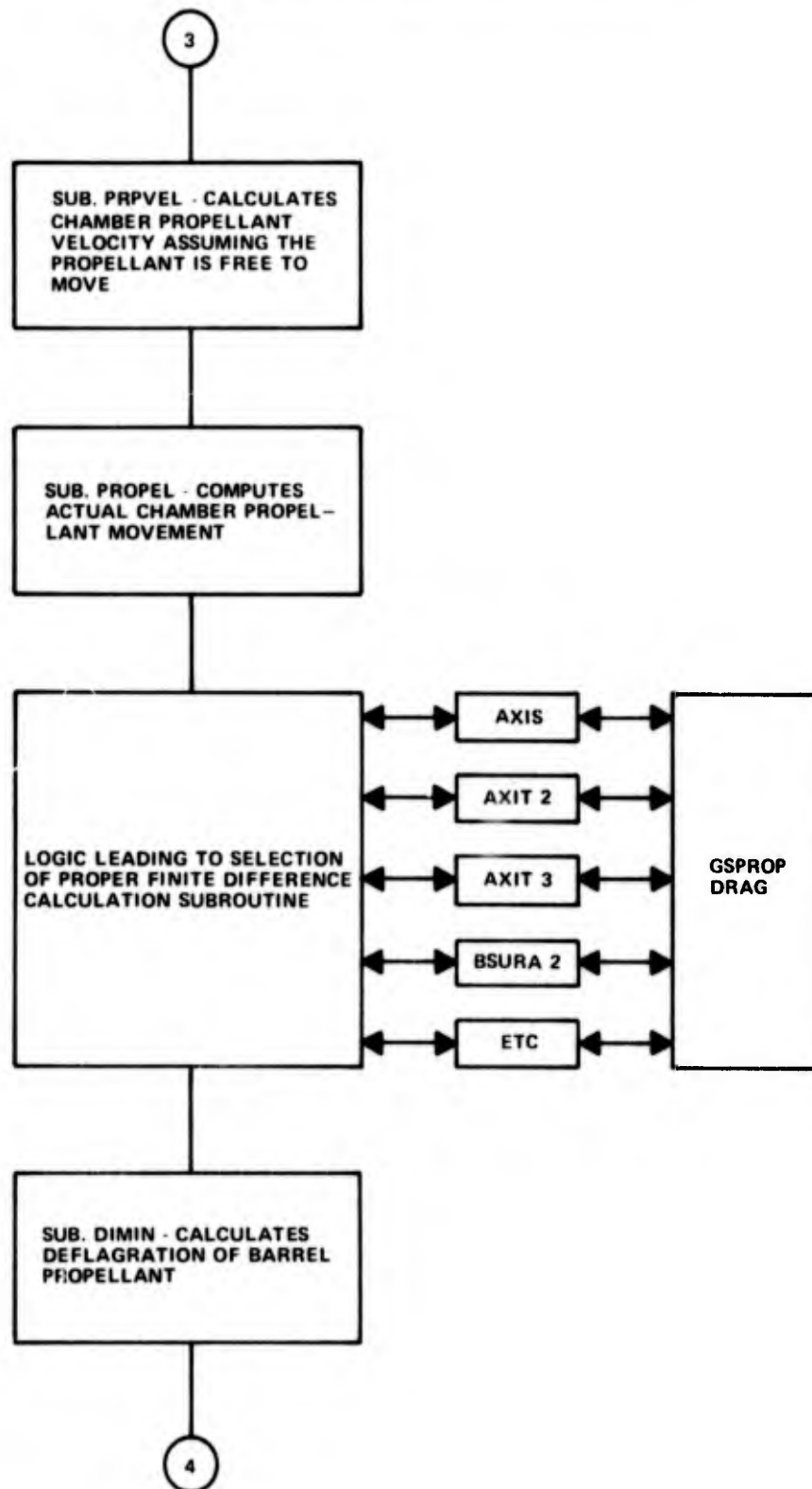


PROGRAM FLOW CHART (CONT.)

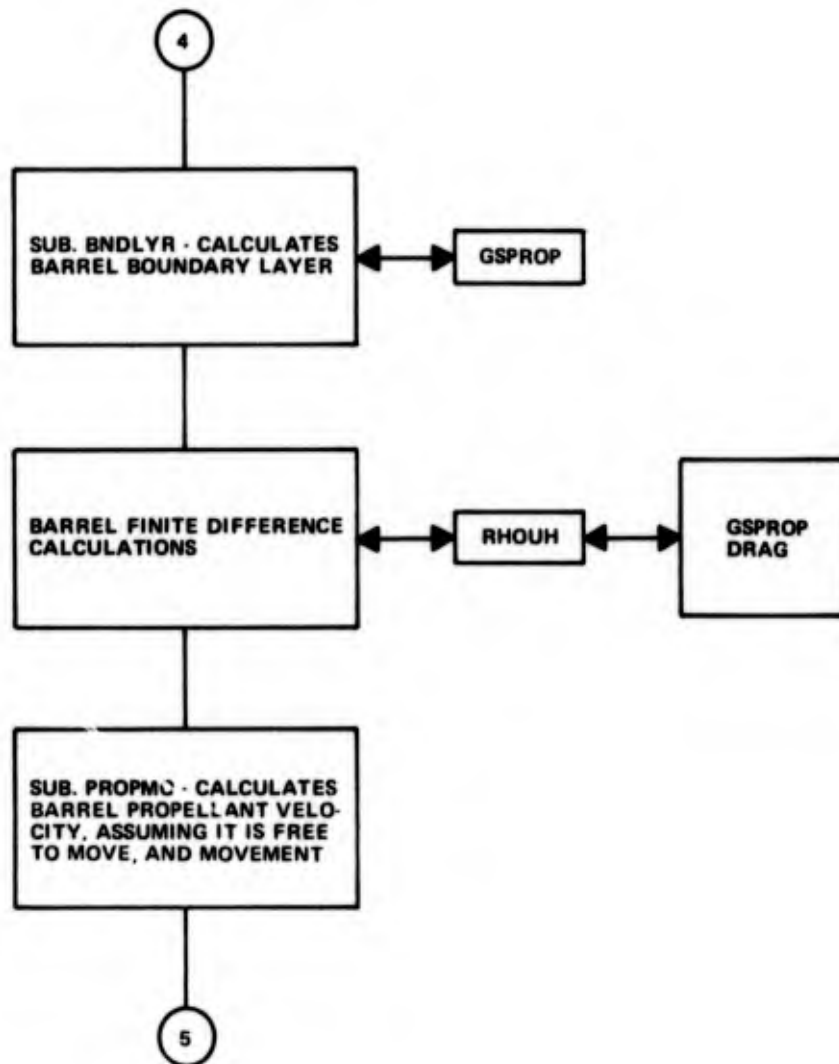




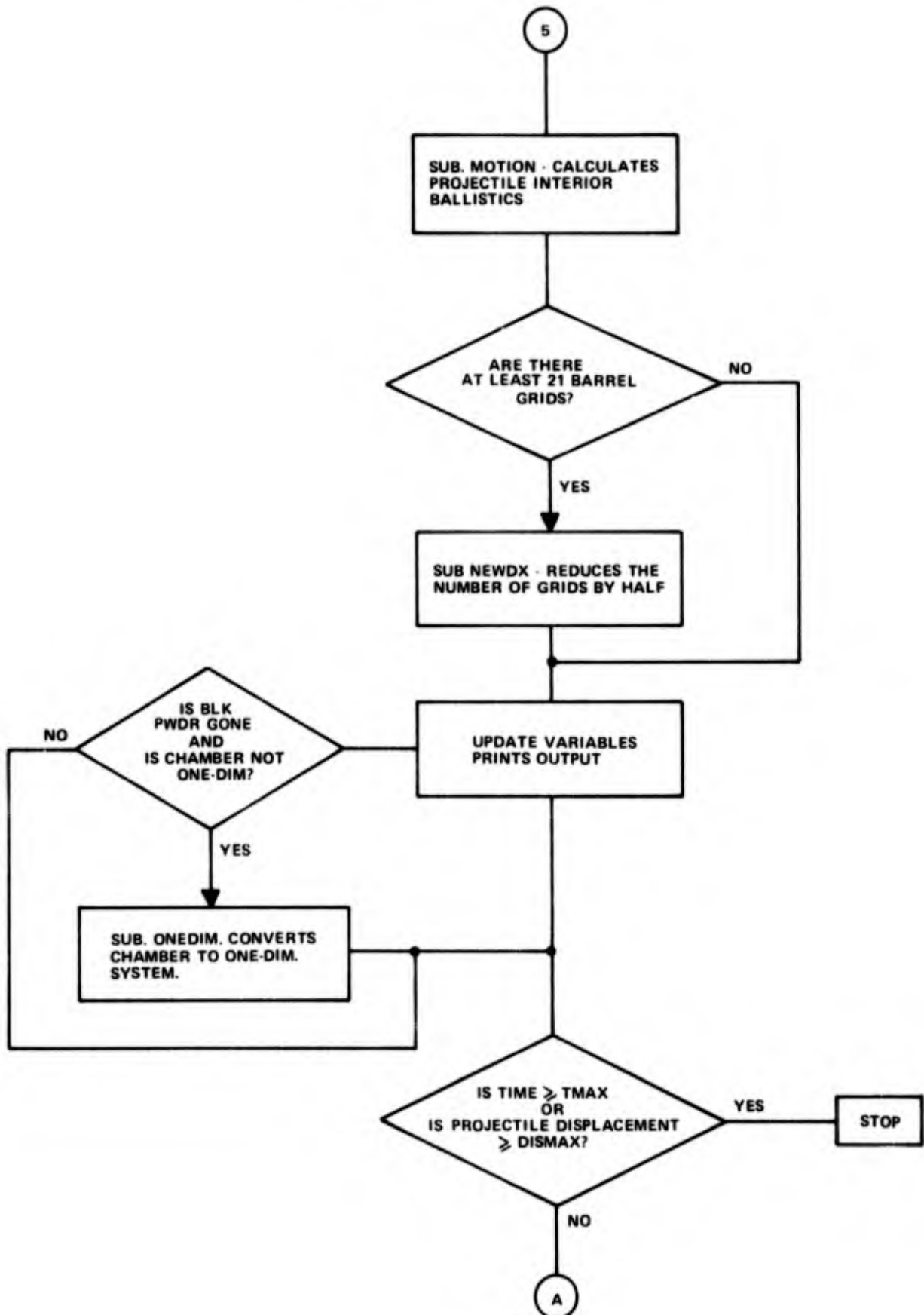
PROGRAM FLOW CHART (CONT.)



PROGRAM FLOW CHART (CONT.)



PROGRAM FLOW CHART (CONT.)



APPENDIX B
FORTRAN IV MACHINE LISTING

PROGRAM P10/(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)	MOD0530 1
C MAIN	MAIN 3
COMMON/OUT/IEP(2),DEP(2),CEP(2),FLOOR,PDALST	MOD0925 1
COMMON/PRMFLC/OUTPRF,UPRM,UPRM,PRPAR,FJSHR,CSFRKS,PRMCOF	MOD0925 2
COMMON/THICK/THICKT,THOUP,PCOMP,BICE,XNTUB,FAIL,MFAIL(60),	MAIN 4
1 THICK(60),PENET,TYPEL	MOD0925 3
COMMON/HANPL2/OUTPA,AP,VP,BORED,PCHEM,BOREDE,DT2BU,DTUSQ,XLBAK	MAIN 6
COMMON/CALLP/OUTLEF1	MAIN 7
COMMON/CHAM/IA,IB,AB,RE,INGX,NGR,IBEG,LEMDB,IPATH(60,5),AREAG(5),	CHAMIX 2
AREALH,AREAL(60),ICM1,ICM2,DIAP1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX 3
AREAL(60),AREALX,CHAM1,CHAM2,CHAM3,LOPGAP,AREAGP(60),DAVG,	CHAMIX 4
AREALG,DIAPH1,BELERD,BELERD,IPS1,IPS2,RADFS,BPIGN	CHAMIX 5
COMMON/CLUCK/TYPE,DELT	MAIN 5
COMMON/EQNS/LEB,LECR,LECR,LECR,TWOIF,QUICK,EMB,TWOGU,DVAXIS,DVAXIT,	EQNS 2
2 JAX,OR,NA,GO,TWOIF,EMB	EQNS 3
COMMON/INPUTS/C1,C2,C3,C4,T0,TIGR,GCNS,RHOF,PHIO,TF,CA,RHOO,	INPUTS 2
3 MU,PU,DU,STKHOP,HK,LM,TIGMBP,GCNS,TCTM	INPUTS 3
COMMON/F/PRINT,MOUCH,MOUGR,PR1,LEECG(32)	P 2
COMMON/FROMO/FIRE	MAIN 13
COMMON/PROU/LE1,LE2,ME1,ME2,PGUAF,BGSLF,NVSP,LEGNF	MOD0925 4
COMMON/IGNTE/PURAD,PCRG,BKOUT,BGRN,IMPBG,	MOD0925 5
1 LOGIC(5),XLIC(5)	MOD0925 6
COMMON/SPLINT/WFOLEC,WHOLEB	MAIN 14
COMMON/BARRL/PHI(100),KHUG(100),HG(100),UG(100),UP(100),	BARRL 2
1 PU(100),IC(100),FMDUT(100),GL(100),UCHAG(100),FRIC1(100),	BARRL 3
2 GCONV(100),UPH(100),UPH1(100),LHOG(100),UH6(100),UUG(100),	BARRL 4
3 APASS(100),AROF(100),AREER(100),CAPASS(100),UAMOM(100),	BARRL 5
4 AREER(100)	BARRL 6
COMMON/CRG/PHIB(60,5),KHOG(60,5),PBG(60,5),UBG(60,5),	HAG 2
1 VBG(60,5),UPB(60,5),PCH(60,5),LZC(60,5),	HAG 3
2 LOMIG(60),GLAG(60,5),XDRAG(60,5),DOTME(60,5),UPBUT(60,5),	HAG 4
3 PHIB(60,5),PHUG(60,5),PHG(60,5),UBGTU(60,5),	HAG 5
4 VBG(60,5),LBB(60,5),DOTMBG(60),LCTMP(60,5),PHIBP(60,5),	HAG 6
5 PHMIG(60,5),LZK(60),TOP(60,5)	HAG 7
LOGICAL PH11,IDEBUG	PLOG 2
LOGICAL IGN1,ORLE,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG 2
LOGICAL FIRE	MAIN 19
LOGICAL PENET	MOD0604 1
LOGICAL FAIL	MAIN 20
LOGICAL BFLFT	MAIN 21
LOGICAL WHOLEC,WHOLEB	MAIN 22
LOGICAL BKOUT,BGRN	MOD0925 7
LOGICAL PGUAF,PGSLF	MOD0925 8
C	MAIN 23
C	MAIN 24
C	MAIN 25
C	MAIN 26
C	MAIN 27
C	MAIN 28
C	MAIN 29
C	MAIN 30
C	MAIN 31
C	MAIN 32
C	MAIN 33
C	MAIN 34
C	MAIN 35
C	MAIN 36

	BRNOU1 = .FALSE.	MUD0925	9
	BUJAM = .FALSE.	MUD0925	10
	BUSTP = .FALSE.	MUD0925	11
	BUURN = .FALSE.	MUD0925	12
	PERET = .FALSE.	MUD0604	2
C		MAIN	37
C	WHEN PROPELLANT AT ALL GRIDS IS IGNITED, IGNIT WILL BE SET TRUE	MAIN	38
C	IN PRPFIR.	MAIN	39
	IGNIT = .FALSE.	MAIN	40
	FAIL = .FALSE.	MAIN	41
C		MAIN	42
C	FAIL WILL BE SET FALSE IN TUBFAL UNTIL THE TUBE HAS	MAIN	43
C	FAILED COMPLETELY (IFAIL(1) = 2) ALONG ITS ENTIRE LENGTH.	MAIN	44
C		MAIN	45
C		MAIN	46
C	CHED IS FALSE UNTIL TWO CONDITIONS ARE SATISFIED AND THE CHAMBER	MAIN	47
C	IS MADE 1-DIMENSIONAL	MAIN	48
C	1. PROPELLANT IS IGNITED AT ALL GRIDS, I.E., IGNIT IS TRUE	MAIN	49
C	2. A POROSITY CONDITION IS SATISFIED	MAIN	50
	CHED = .FALSE.	MAIN	51
C		MAIN	52
C	WHOLEC WILL BE SET FALSE IN REGRES WHEN PROPELLANT GRAINS SPLINTER	MAIN	53
C	IN AT LEAST ONE CHAMBER GRID. WHOLEC ACTS SIMILARLY IN THE BARREL.	MAIN	54
	WHOLEC = .TRUE.	MAIN	55
	WHOLEB = .TRUE.	MAIN	56
C		MAIN	57
C	BPLEFT WILL BE SET FALSE IN PRIMER WHEN ALL THE BLACK POWDER HAS	MAIN	58
C	BEEN BURNED.	MAIN	59
	BPLEFT = .TRUE.	MAIN	60
	FINC = .FALSE.	MAIN	61
C		MAIN	62
C	FINC WILL BE SET TRUE IN PRPFIR WHEN PROPELLANT IN AT LEAST ONE	MAIN	63
C	GRID IGNITES.	MAIN	64
	CALL CHSET	MAIN	65
	CALL BARSET	MAIN	66
	IBURNP=1:EGG	MUD0925	13
	IMPG=10	MUD0925	14
	NSVP=1	MUD1018	1
C		MAIN	67
C		MAIN	68
C		MAIN	69
C	*****	MAIN	70
C	*****	MAIN	71
C	*****	MAIN	72
C		MAIN	73
10	CONTINUE	MAIN	74
	IF (.NOT. FAIL) CALL TUBFAL	MAIN	75
20	CONTINUE	MAIN	76
	PR12 = .FALSE.	MAIN	77
	IF (MOD (IPRINT, MDOCH) .EQ. 0) PR11 = .TRUE.	MAIN	78
C		MAIN	79
C	*****	MAIN	80
C	SETTING DUMMY GRIDS	MAIN	81
C	*****	MAIN	82
C	SET DUMMY CHAMBER GRID 1 = NGX+1 THE SAME AS BARREL GRID 2	MAIN	83
C	AND DUMMY BARREL GRID 1 AS THE VOLUME-WEIGHTED AVERAGE OF CHAMBER	MAIN	84
C	GRID 1 = NGX.	MAIN	85

C	V06 IS 0.0 AT CHANNEL GRID NGX+1	MAIN	86
C	AREAS AT CHANNEL GRID 1 WERE SET TO 0.0 IN CHSLT	MAIN	87
C		MAIN	88
	NP1 = NGX+1	MAIN	89
	IF (NX .GE. 2) GO TO 140	MAIN	90
	NP1 = NGX - 1	MAIN	91
	DO 130 J=1,NGR	MAIN	92
	KHUGG(NP1,J) = KHUGG(NM1,J)	MAIN	93
	UGG(NP1,J) = 2*VP- UGG(NM1,J)	MOD1205	1
	HUG(NP1,J) = HUG(NM1,J)	MAIN	95
	PCN(NP1,J) = PCN(NM1,J)	MAIN	96
	PHIBG(NP1,J) = PHIBG(NM1,J)	MAIN	97
	UPB(NP1,J) = 2*VP- UPB(NM1,J)	MOD1205	2
130	CONTINUE	MAIN	99
	GO TO 155	MAIN	100
C		MAIN	101
C		MAIN	102
140	CONTINUE	MAIN	103
	DO 150 J=1,NGR	MAIN	104
	KHUGG(NP1,J) = KHUGG(2)	MAIN	105
	UGG(NP1,J) = UG(2)	MAIN	106
	HUG(NP1,J) = HG(2)	MAIN	107
	PCN(NP1,J) = PG(2)	MAIN	108
	PHIBG(NP1,J) = PHI(2)	MAIN	109
	PHIBP(NP1,J) = 1.0	MAIN	110
	UPB(NP1,J) = UP(2)	MAIN	111
150	CONTINUE	MAIN	112
C		MAIN	113
C		MAIN	114
155	CONTINUE	MAIN	115
	IF (CNEB) GO TO 170	MAIN	116
	KHUGG(1) = 0.0	MAIN	117
	UG(1) = 0.0	MAIN	118
	HG(1) = 0.0	MAIN	119
	PG(1) = 0.0	MAIN	120
	PHI(1) = 0.0	MAIN	121
	UP(1) = 0.0	MAIN	122
C		MAIN	123
	DO 160 J=1,NGR	MAIN	124
	KHUG(1) = KHUG(1) + KHUGG(NGX,J)*AREAG(J)	MAIN	125
	UG(1) = UG(1) + UGG(NGX,J)*AREAG(J)	MAIN	126
	HG(1) = HG(1) + HUG(NGX,J)*AREAG(J)	MAIN	127
	PG(1) = PG(1) + PCN(NGX,J)*AREAG(J)	MAIN	128
	PHI(1) = PHI(1) + PHIBG(NGX,J)*AREAG(J)	MAIN	129
	UP(1) = UP(1) + UPB(NGX,J)*AREAG(J)*(1.0 - PHIBG(NGX,J))	MAIN	130
160	CONTINUE	MAIN	131
C		MAIN	132
	KHUG(1) = KHUG(1)/AREACH	MAIN	133
	UG(1) = UG(1)/AREACH	MAIN	134
	HG(1) = HG(1)/AREACH	MAIN	135
	PG(1) = PG(1)/AREACH	MAIN	136
	PHI(1) = PHI(1)/AREACH	MAIN	137
	IF (PHI(1) .GT. 0.999) GO TO 165	MAIN	138
	UP(1) = UP(1)/(AREACH*(1.0 - PHI(1)))	MAIN	139
	GO TO 180	MAIN	140
C		MAIN	141
165	UP(1) = 0.0	MAIN	142

GO TO 160	MAIN	143
C	MAIN	144
170 RHOB(1) = RHOB(RGX,1)	MAIN	145
UB(1) = UB(RGX,1)	MAIN	146
RG(1) = RG(RGX,1)	MAIN	147
PG(1) = PG(RGX,1)	MAIN	148
PHI(1) = PHIB(RGX,1)	MAIN	149
UP(1) = UP(RGX,1)	MAIN	150
C	MAIN	151
180 CONTINUE	MAIN	152
C	MAIN	153
C	MAIN	154
C.....	MAIN	155
C CHAMBER SUBROUTINES	MAIN	156
C.....	MAIN	157
C	MAIN	158
IF(.NOT. BPIGN) CALL BPIR	MAIN	159
C LOGICAL VARIABLE BPLEFT WILL BE SET TO FALSE IN PRIMER WHEN ALL	MAIN	160
C BLACK POWDER IS BURNED.	MAIN	161
IF(BPLEFT) CALL PRIMER	MAIN	162
GAMBF=1.35	MOD0925	16
UPRM=156.1*SGRT((GAMBF-1.0)*HPP)	MOD0925	17
IF(TIME .GE. .0004) DOTMPM=2.66*EXP(-1200.*TIME)/AREAAX	MOD0925	18
IF(TIME .LT. .0004) DOTMPM=4090.*TIME/AREAAX	MOD0925	19
IF(TIME .GT. .005) DOTMPM=0.	MOD0925	20
DOTMPM=PRMCOF*DOTMPM	MOD0925	21
FLAG=0.	MOD0925	22
DO 177 J=1,NGR	MOD0925	23
DO 177 I=1,NGX	MOD0925	24
IF(PHIBP(I,J) .GE. 0.9999) GO TO 177	MOD0925	25
IF(FLAG .EQ. 1.) GO TO 177	MOD0925	26
DOTMP(I,J)=DOTMP(I,J) + DELT*DOTMPM	MOD0925	27
FLAG=1.	MOD0925	28
DOTMPM=DOTMP(I,J)	MOD0925	29
177 CONTINUE	MOD0925	30
IF(.NOT. UNEL) CALL MFLOW	MAIN	163
C	MAIN	164
C PRPFIR IS CALLED UNTIL ALL PROPELLANT IS IGNITED.	MAIN	165
C MUST NOT USE TOTAL POROSITY IN PHIBG ARRAY HERE	MOD0925	31
C	MOD0925	32
IF(.NOT. IGNIT) CALL PRPFIR	MAIN	166
CALL REGRES	MAIN	167
IF(TIME .GT. DELT) CALL PRPVEL	MOD1017	1
IF(TIME .GT. DELT) CALL PROPEL	MOD1017	2
C	MAIN	170
C	MAIN	171
C PUT THE TOTAL POROSITY INTO ARRAY PHIBG FOR USE IN THE PATH	MAIN	172
C SUBROUTINES.	MAIN	173
DO 190 J=1,NGR	MAIN	174
DO 190 I=1,NGX	MAIN	175
PHIBG(I,J) = PHIBG(I,J) + PHIEP(I,J) - 1.0	MAIN	176
190 CONTINUE	MAIN	177
C PATH SUBROUTINES	MAIN	178
XB = -UX	MAIN	179
DO 300 IX=1,NGX	MAIN	180
DO 300 IR=1,NGR	MAIN	181
IPA = IPATH(IX,IR)	MAIN	182

	GO TO (201,202,203,204,205,206,207,208,209,210,211,212),IPA	MAIN	183
201	CALL AXIS	MAIN	184
	GO TO 300	MAIN	185
202	IF(CHAM1) CALL AX11	MAIN	186
	IF(CHAM2) CALL AX12	MAIN	187
	IF(CHAM3 .AND. IR .EQ. 3) CALL AX13	MAIN	188
	IF(CHAM3 .AND. IR .NE. 3) CALL AX12	MAIN	189
	GO TO 300	MAIN	190
203	CALL INTER	MAIN	191
	GO TO 300	MAIN	192
204	CALL BNDT	MAIN	193
	GO TO 300	MAIN	194
205	CALL FSURFA	MAIN	195
	GO TO 300	MAIN	196
206	CALL FSURF1	MAIN	197
	GO TO 300	MAIN	198
207	CALL FSURF1	MAIN	199
	GO TO 300	MAIN	200
208	CALL FSURF1	MAIN	201
	GO TO 300	MAIN	202
209	IF(CHAM1) CALL FSURFA	MAIN	203
	IF(.NOT. CHAM1) CALL BSURAZ	MAIN	204
	GO TO 300	MAIN	205
210	IF(CHAM1) CALL BSURF1	MAIN	206
	IF(CHAM2) CALL BSURF2	MAIN	207
	IF(CHAM3 .AND. IR .EQ. 3) CALL BSURF3	MAIN	208
	IF(CHAM3 .AND. IR .NE. 3) CALL BSURF2	MAIN	209
	GO TO 300	MAIN	210
211	CALL BSURF1	MAIN	211
	GO TO 300	MAIN	212
212	CALL BSURF1	MAIN	213
300	CONTINUE	MAIN	214
C		MAIN	215
C		MAIN	216
C	FIX ARRAY PHIBG SO THAT IT ONLY REPRESENTS POROSITY OF THE	MAIN	217
C	PROPELLANT AND NOT THE TOTAL POROSITY.	MAIN	218
	DO 310 J=1,NCR	MAIN	219
	DO 310 I=1,NCA	MAIN	220
	PHIBG(I,J) = PHIBG(I,J) + 1.0 - PHIBP(I,J)	MAIN	221
310	CONTINUE	MAIN	222
C		MAIN	223
C	*****	MAIN	224
C	BARREL SUBROUTINES	MAIN	225
C	*****	MAIN	226
C		MAIN	227
	IF(IX .LT. 2) GO TO 350	MAIN	228
	CALL DIMIN	MAIN	229
	IF(IX .GT. 2) CALL BNDLYR	MAIN	230
	CALL KROOH	MAIN	231
	CALL PROPPG	MAIN	232
350	CONTINUE	MAIN	233
	CALL MOTION	MAIN	234
C		MAIN	235
C		MAIN	236
C	*****	MAIN	237
C	UPDATE AND PRINT	MAIN	238
C	*****	MAIN	239

C		MAIN	240
	IF (NAX .LT. 21) GO TO 360	MAIN	241
	CALL NEWDA	MAIN	242
	PR11 = .TRUE.	MAIN	243
C		MAIN	244
360	CONTINUE	MAIN	245
	CALL UPDATE	MAIN	246
C		MAIN	247
	IF (XPR .LT. ALBAR) GO TO 370	MAIN	248
	STOP	MAIN	249
C		MAIN	250
370	CONTINUE	MAIN	251
	IF (TIME .LT. TF) GO TO 400	MAIN	252
	WRITE(6,2000) TIME	MAIN	253
	STOP	MAIN	254
C		MAIN	255
400	TIME = TIME + DELT	MAIN	256
	IPRINT = IPRINT + 1	MAIN	257
	GO TO 10	MAIN	258
C		MAIN	259
C		MAIN	260
1000	FORMAT(35L1)	MAIN	261
2000	FORMAT(' TIME =',F13.7,' SO WE STOP')	MAIN	262
2002	FORMAT(1H0,3A,35L3)	MAIN	263
2003	FORMAT(1H1,*ARRAY 1DEBUG*)	MAIN	264
2004	FORMAT(1H0,5A,* 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	MAIN	265
	. 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35*)	MAIN	266
	END	MAIN	267

7	DIAMB1,BORE1,CHAM1,CHAM2,CHAM3,PH1,TYPE,	MOD0925	40
C	RUBP,KNOBP,LPRAU,BPENS,AGENB,EGLEP,EXPEP,PRACOF,	MOD0925	41
	STOPGAP,HBP,CVUBP,CVNEP,TIGNEP,XHP,ALPHBP,	CHSET	37
	* THICK1,PHOOF,PCOMP,BTUB,XNTUB,XNTUB,SPRKS	MOD0925	42
C		CHSET	39
C		CHSET	40
C		CHSET	41
C	*****	CHSET	42
C	READ INPUTS AND CALCULATE CONSTANTS FOR SUBROUTINES	CHSET	43
C	*****	CHSET	44
	DO 2222 I = 1,60	CHSET	45
	IFAIL(I) = 0	CHSET	46
2222	CONTINUE	CHSET	47
	CALL CLEAR(THICK(1),THICK(60))	CHSET	48
C		CHSET	49
	READ(5,CHINP)	CHSET	50
	IF(10EBUG(2)) WRITE(6,CHINP)	CHSET	51
C		CHSET	52
C	THREE DIFFERENT VERSIONS OF THE CHAMBER CAN BE RUN.	CHSET	53
C	IF CHAM1 IS TRUE, THE CHAMBER CONSISTS OF AN INTERDEPENDENT TWO-	CHSET	54
C	DIMENSIONAL SYSTEM.	CHSET	55
C	IF CHAM2 IS TRUE, THE CHAMBER CONSISTS OF TWO INDEPENDENT ONE-	CHSET	56
C	DIMENSIONAL SYSTEMS.	CHSET	57
C	IF CHAM3 IS TRUE, THE CHAMBER CONSISTS OF THREE INDEPENDENT ONE-	CHSET	58
C	DIMENSIONAL SYSTEMS.	CHSET	59
C		CHSET	60
	PN = FLOAT(MPEP)	CHSET	61
	THICK1 = THICK1/12.0	CHSET	62
	DO 1 I=1,60	CHSET	63
1	THICK(I)=THICK1	CHSET	64
	PHOOF = PHOOF*144.0	CHSET	65
	PCOMP = PCOMP*144.0	CHSET	66
	BTUB=BTUB/(12.0*144.0**XNTUB)	CHSET	67
	ALU = ALU/12.	CHSET	68
	GUU = GUU/12.	CHSET	69
	GUU = GUU/12.	CHSET	70
	DIAM1 = DIAM1/12.	CHSET	71
	DIAM2 = DIAM2/12.	CHSET	72
	DIS1 = DIS1/12.	CHSET	73
	DIS2 = DIS2/12.	CHSET	74
	DIS3 = DIS3/12.	CHSET	75
	DIS4 = DIS4/12.	CHSET	76
	DIAMB1 = DIAMB1/12.	CHSET	77
	URAM = URAM/12.	CHSET	78
	GAP = GAP/12.	CHSET	79
	XLBEL = XLBEL/12.	CHSET	80
	TOPGAP = TOPGAP/12.	CHSET	81
	PU = PU*144.	CHSET	82
	AGEN = AGEN/(12.*144.**PEXP)	CHSET	83
	BGEN = BGEN/(12.*144.**PEXP)	CHSET	84
	CGEN = CGEN/12.	CHSET	85
	BPRADU = BPRADU/12.	CHSET	86
	BPENS = BPENS/454.*16.58*1728.	CHSET	87
	AGENB = AGENB/(12.0*144.0**EXPPF)	CHSET	89
	BGENB = BGENB/12.0	CHSET	90
C		CHSET	91
C	INITIALLY THE GAS CONSTANTS ARE THOS, OF THE BLACK POWDER.	CHSET	92

C	AFTER EACH TIME INTERVAL, IN UPDATE, THE GAS CONSTANTS WILL BE	CHSET	93
C	CALCULATED ON THE BASIS OF THE GASES PRESENT.	CHSET	94
	RO = RROBP	CHSET	95
	RMO = RROBP	CHSET	96
	CVU = CVOBP	CHSET	97
	CVH = CVHBP	CHSET	98
C		CHSET	99
C	BURER WAS TAKEN OUT OF BAKING BECAUSE IT IS NEEDED HERE FOR AREA	CHSET	100
C	CALCULATIONS.	CHSET	101
	BURER = BURER/12.	CHSET	102
	BURER = 0.5*BURER	CHSET	103
	BURER = PI*BURER*BURER	CHSET	104
C		CHSET	105
C	DETERMINE DX, IBEGB, IDEGB, IENDB	CHSET	106
	DX = DRAP/FLCATE*(NGX-1)	CHSET	107
	IF(XLDEL + GAP.GT.DRAP) DX = DRAP - XLDEL	CHSET	108
C		CHSET	109
C	IF THE WELL TUBE BEGINS IN GRID 1, PUSH IT INTO GRID 2.	CHSET	110
	BELBEG = DRAP - (XLDEL + GAP)	CHSET	111
	IBEGB=BELBEG/DX+1.2	CHSET	112
	IDEGB=IBEGB	CHSET	113
	DELEND = DRAP - DX	CHSET	114
	IENDB = NGX - IFIX(CAT/LX + 0.5)	CHSET	115
C		CHSET	116
C	FRACT1 AND FRACT2 ARE THE FRACTIONS OF GRIDS IBEGB AND IENDB	CHSET	117
C	RESPECTIVELY THAT THE WELL TUBE OCCUPIES.	CHSET	118
	IF(IBEGB.EQ.1)FRACT1=1.0-2.0*BELBEG/LX	CHSET	119
	IF(IDEGB.GT.1)FRACT1=((FLOAT(IBEGB)-0.5)*DX-BELBEG)/DX	CHSET	120
	FRACT2 = (DELEND - (FLOAT(IENDB) - 1.5)*DX)/DX	CHSET	121
C		CHSET	122
C	DEGCUNS, AND GGCUNS ARE NEEDED IN SUBROUTINE PKPFIR.	CHSET	123
	DN = (1.5*0.0001*CVU*ALD)**0.333	CHSET	124
	GCUNS = 2.0*SQRT(ALPHA/3.141593)/XK	CHSET	125
	GGCUNS = 2.0*SQRT(ALPHA/3.141593)/XKBP	CHSET	126
	MOD0925	MOD0925	43
C		CHSET	127
C	ATPB AND CT ARE NEEDED FOR CALCULATING THE BURN RATE IN REGRES	CHSET	128
	ATPB = AGEN*10 + RGEN	CHSET	129
	CT = CGEN*10	CHSET	130
C		CHSET	131
C	VISG IS A CONSTANT USED IN SUBROUTINE DRAG	CHSET	132
	VISG = LM*GRAV	CHSET	133
C		CHSET	134
C	CALCULATE HW FROM TW AND PU	CHSET	135
	CALL GSPROP(RD,RRO,H,CVU,CVH,CV,PO,HX,TW,RHCDUM,0.0,0.0,GAM,CP,1)	CHSET	136
C		CHSET	137
C		CHSET	138
C	*****	CHSET	139
C	INITIALIZE ARRAY IPATH.	CHSET	140
C	*****	CHSET	141
C		CHSET	142
C	CALL PATHS	CHSET	143
C		CHSET	144
C		CHSET	145
C	*****	CHSET	146
C	DETERMINE DELT AND CALCULATE CONSTANTS FOR FINITE DIFFERENCE EQNS.	CHSET	147
C	*****	CHSET	148

C		CHSET	149
C	START TIME AT 0.0 AND DETERMINE DELT, THE TIME INTERVAL LENGTH	CHSET	150
	TIME = 0.0	CHSET	151
	GU = GRAV*XDLL	CHSET	152
	TWOGU = 2.0*GU	CHSET	153
	DELT = DELTA*DX/SGRT(TWOGU*HMAX)	CHSET	154
	TWODI = 2.0*DELT	CHSET	155
	DTUX = DELT/LX	CHSET	156
	DTUR = DELT/LR	CHSET	157
	T2UR = 0.5*DTUR	CHSET	158
	T2UX = 0.5*DTUX	CHSET	159
	TWOTUR = 2.0*DTUR	CHSET	160
C		CHSET	161
C		CHSET	162
C		CHSET	163
C	*****	CHSET	164
C	CALCULATE CROSS-SECTIONAL AREAS ASSOCIATED WITH THE CHAMBER	CHSET	165
C	*****	CHSET	166
C		CHSET	167
C		CHSET	168
C	CALL AREAS	CHSET	169
C		CHSET	170
C		CHSET	171
C		MUD0301	2
C	USING THE AREAS JUST CALCULATED, CALCULATE THE INITIAL VOLUME	MUD0301	3
C	OF CHAMBER RADIAL ROWS 1, 2 AND 3. ALSO CALCULATE THE TOTAL	MUD0301	4
C	CHAMBER VOLUME.	MUD0301	5
	ARROW1 = (FLOC1(RGA) - 0.5)*DX*AREAX	MUD0301	6
C		MUD0301	7
	ARROW2 = AREAR(1)*DX*0.5	MUD0301	8
	DO 100 I = 2,NGX	MUD0301	9
	ARROW2 = ARROW2 + AREAR(I)*DX	MUD0301	10
100	CONTINUE	MUD0301	11
C		MUD0301	12
	ARTOT = ARROW1 + ARROW2	MUD0301	13
	IF(CHANG) GO TO 120	MUD0301	14
C		MUD0301	15
	ARROW3 = AREAG(1)*DX*0.5	MUD0301	16
	DO 110 I = 2,NGX	MUD0301	17
	ARROW3 = ARROW3 + AREAG(I)*DX	MUD0301	18
110	CONTINUE	MUD0301	19
	ARTOT = ARTOT + ARROW3	MUD0301	20
C		MUD0301	21
120	CONTINUE	MUD0301	22
C	*****	CHSET	172
C	INITIALIZE CHAMBER MATRIX	CHSET	173
C	*****	CHSET	174
C		CHSET	175
	CALL CLEAR(XL(1,1),DL(60,5))	CHSET	176
	CALL CLEAR(XLUT(1,1),DLUT(60,5))	CHSET	177
	CALL CLEAR(PHIDC(1,1),IBP(60,5))	CHSET	178
	CALL CLEAR(BPRAD(1,1),OPRAD(60,5))	CHSET	179
	CALL CLEAR(PFORCE(1,1),FFORCT(60,5))	CHSET	180
C		CHSET	181
C	OPB IS INITIALLY 0.0	CHSET	182
C	INITIALLY SET ALL POROSITIES TO 1.0 AND THEN CHANGE THOSE THAT HAVE	CHSET	183
C	BLACK POWDER OR PROPELLANT	CHSET	184

DO 510 J=1,NEX	CHSET	185
DO 510 I=1,NEX	CHSET	186
PHIBG(I,J) = 1.0	CHSET	187
PHIBD(I,J) = 1.0	CHSET	188
PHIBP(I,J) = 1.0	CHSET	189
PHIBD(I,J) = 1.0	CHSET	190
510 CONTINUE	CHSET	191
C CALCULATE INITIAL POROSITIES OF GRIDS CONTAINING BLACK POWDER	CHSET	192
C	CHSET	193
C CALL BEIRIT	CHSET	194
C	CHSET	195
C	CHSET	196
DO 1001 I=1,NEX	CHSET	197
DO 1000 J=1,2	CHSET	198
BPRAU(I,J)=BPRAUO	CHSET	199
1000 IBP(I,J)=10	CHSET	200
1001 CONTINUE	CHSET	201
C NUMBER OF GRIDS OF TUBE	CHSET	202
NG1=18EGG-18EGC	CHSET	203
NG13 = NG1/3	CHSET	204
LIM1 = 18EGC + 1613	CHSET	205
18EG2 = LIM1 + 1	CHSET	206
LIM2 = LIM1 + NG13	CHSET	207
18EG3 = LIM2 + 1	CHSET	208
C	CHSET	209
C SINCE THE INTEGER DIVISION BY 3 MIGHT INVOLVE TRUNCATION THE LAST	CHSET	210
C THIRD OF THE TUBE MAY HAVE MORE THAN NG13 GRIDS.	CHSET	211
C	CHSET	212
DO 530 I=18EG2,LIM1	CHSET	213
BPRAU(I,1) = BPRAUO	CHSET	214
IBP(I,1) = 10	CHSET	215
530 CONTINUE	CHSET	216
DO 540 I=18EG3,LIM2	CHSET	217
BPRAU(I,1) = BPRAUO	CHSET	218
IBP(I,1) = 10	CHSET	219
540 CONTINUE	CHSET	220
DO 550 I=18EG3,18EG3	CHSET	221
BPRAU(I,1) = BPRAUO	CHSET	222
IBP(I,1) = 10	CHSET	223
550 CONTINUE	CHSET	224
C	CHSET	225
C PROPELLANT IS PACKED IN GRIDS ABOVE THE BELL TUBE ONLY. GRAIN	CHSET	226
C DIMENSIONS AND GRAIN SURFACE TEMPERATURE WILL BE LEFT 0.0 AT OTHER	CHSET	227
C GRIDS.	CHSET	228
IF (CHARG) NG = 2	CHSET	229
IF (.NOT. CHARG) NG = NG1	CHSET	230
DO 570 I=18EG3,18EG3	CHSET	231
12K(I) = 10	CHSET	232
DO 560 J=2,NR	CHSET	233
XL(I,J) = ALU	CHSET	234
DU(I,J) = 0.0	CHSET	235
DA(I,J) = 0.10	CHSET	236
12C(I,J) = 10	CHSET	237
ALUDT(I,J)=XL0	CHSET	238
DUUDT(I,J)=0.00	CHSET	239
	CHSET	240
	CHSET	241
	CHSET	242
	CHSET	243
	CHSET	244
	CHSET	245
	CHSET	246
	CHSET	247

	DITOT(1,0)=D10		
500	CONTINUE	CHSET	248
570	CONTINUE	CHSET	249
C		CHSET	250
C		CHSET	251
C	CALCULATE VOLUME ABOVE THE BELL TUBE AND POROSITY OF GRIDS	CHSET	252
C	WITH PROPELLANT.	CHSET	253
	IBP1 = IREGM + 1	CHSET	254
	IBP1 = IENDM - 1	CHSET	255
	VOLCHG=AREAR(IREGM)*FRAC11*DX/2.	CHSET	256
	IF(IREGM .GT. 1) VOLCHG = VOLCHG+2.0	CHSET	257
	DO 580 I=IBP1,IBP1	CHSET	258
	VOLCHG = VOLCHG + AREAR(I)*DX	CHSET	259
580	CONTINUE	CHSET	260
C		CHSET	261
	VOLCHG = VOLCHG + AREAR(IENDM)*FRAC12*DX	CHSET	262
C		CHSET	263
	RHOE = CHGT/VOLCHG	CHSET	264
	PH10=1.0-RHOE/RHOP	CHSET	265
C		MOD1111	1
	DO 585 J=2,N	CHSET	267
	PH10(I,REGM,J)=1.0-FRAC11 + FRAC11*(1.0-RHOE/RHOP)	CHSET	268
585	CONTINUE	MOD0925	44
C		CHSET	270
	DO 590 J=2,N	CHSET	271
	DO 590 I=IBP1,IBP1	CHSET	272
	PH10(I,J)=1.0-RHOE/RHOP	CHSET	273
590	CONTINUE	MOD0925	45
C		CHSET	275
	DO 595 J=2,N	CHSET	276
	PH10(I,ENDM,J)=1.0-FRAC12 + FRAC12*(1.0-RHOE/RHOP)	CHSET	277
595	CONTINUE	MOD0925	46
C	DETERMINE HU AND RHOD FROM PG AND T0 AND GSPROP	CHSET	279
C	CALL GSPROP(HU,RHO,N,CVO,CVH,CV,P0,H0,T0,RHCO,0.0,0.0,0.0,GAM,CP,1)	CHSET	280
	DO 600 J=1,NGH	CHSET	282
	DO 600 I=1,NGX	CHSET	283
	PH(I,J) = P0	CHSET	284
	H0(I,J) = H0	CHSET	285
	RH0(I,J) = RH00	CHSET	286
	U0(I,J) = U0	CHSET	287
	V0(I,J) = V0	CHSET	288
	T0(I,J) = T0	CHSET	289
600	CONTINUE	CHSET	290
C		CHSET	291
	IF(10LEEND(3)) WRITE(6,2006)	CHSET	292
	DO 630 J=1,NGH	CHSET	293
	IF(10LEEND(3)) WRITE(6,2007) J,(PH10(I,J),I=1,NGX)	CHSET	294
630	CONTINUE	CHSET	295
C		CHSET	296
	IF(10LEEND(3)) WRITE(6,2008)	CHSET	297
	DO 640 J = 1,NGH	CHSET	298
	IF(10LEEND(3)) WRITE(6,2007) J,(PH10(I,J),I=1,NGX)	CHSET	299
640	CONTINUE	CHSET	300
C		CHSET	301
C		CHSET	302
	*****	CHSET	303
		CHSET	304

C	SET UP HOLLS	CHSET	305
C	*****	CHSET	306
C		CHSET	307
C	SET UP HOLLS IN BELL TUBE AND ANY PSEUDO HOLES BETWEEN RADIAL ROWS	CHSET	308
C	ONE AND TWO.	CHSET	309
C	CALL HOLSET	CHSET	310
C		CHSET	311
C	GET HOLE AREA AT EACH GRID BETWEEN RADIAL ROWS ONE AND TWO.	CHSET	312
C	CALL HOLLS	CHSET	313
C		CHSET	314
C	GET HOLE AREA AT EACH GRID BETWEEN RADIAL ROWS TWO AND THREE IF	CHSET	315
C	CHAM3 IS TRUE. IT IS CALCULATED USING AN AVERAGE DIAMETER	CHSET	316
C	OBTAINED FROM SURROUNDING AREAS.	CHSET	317
C	IF(.NOT. CHAM3) GO TO 700	CHSET	318
C	AREAH2 = 0.05*(1/DELTA)*0.5*(DAVG - TOPGAP)*TOPGAP*	CHSET	319
C	3 SQRT(2.0/(DAM - 1.0))	CHSET	320
C		CHSET	321
C		CHSET	322
C	700 CONTINUE	CHSET	323
C		CHSET	324
C	*****	CHSET	325
C	COMPUTE TOTAL MASS IN THE SYSTEM	CHSET	326
C	*****	CHSET	327
C		CHSET	328
C	LOGIC IS NOT WRITTEN FOR CHAM1 TRUE.	CHSET	329
C	IF(CHAM1) WRITE(6,2000)	CHSET	330
C	IF(CHAM1) GO TO 600	CHSET	331
C		CHSET	332
C	TO FIND THE TOTAL GAS MASS, SUM THE PRODUCT OF TOTAL POROSITY*	CHSET	333
C	DENSITY OF GAS*VOLUME AT EACH GRID.	CHSET	334
C	SINCE GAS DENSITY AND GRID LENGTH ARE CONSTANT, MULTIPLY BY THE	CHSET	335
C	AFTER SUMMING.	CHSET	336
C	FIRST SUM ALONG THE AXIS. THERE ALL AREAS ARE AREAAX SO FACTOR IT.	CHSET	337
C	SUM = 0.5*(PHIBG(1,1) + PHIBP(1,1) - 1.0)	CHSET	338
C	DO 720 I=2,NXA	CHSET	339
C	SUM = SUM + PHIBG(1,I) + PHIBP(1,I) - 1.0	CHSET	340
C	720 CONTINUE	CHSET	341
C	TOTM = SUM*AREAAX	CHSET	342
C		CHSET	343
C	SUM = 0.5*(PHIBG(1,2) + PHIBP(1,2) - 1.0)*AREAR(1)	CHSET	344
C	DO 730 I=2,NXA	CHSET	345
C	SUM = SUM + (PHIBG(1,I) + PHIBP(1,I) - 1.0)*AREAR(I)	CHSET	346
C	730 CONTINUE	CHSET	347
C	TOTM = TOTM + SUM	CHSET	348
C		CHSET	349
C	IF(CHAM2) GO TO 750	CHSET	350
C	*****PHIBG(1,3) AND PHIBP(1,3) ARE 1.0 FOR ALL I	CHSET	351
C	SUM = 0.5*AREAGP(1)	CHSET	352
C	DO 740 I=2,NXA	CHSET	353
C	SUM = SUM + AREAGP(I)	CHSET	354
C	740 CONTINUE	CHSET	355
C	TOTM = TOTM + SUM	CHSET	356
C		CHSET	357
C	750 CONTINUE	CHSET	358
C	TOTM = TOTM*PHUB*DX	CHSET	359
C		CHSET	360
C	ADD IN MASS OF BLACK POWDER AND PROPELLANT.	CHSET	361

	TOTM = TOTM + BPCMG + CHWT	CHSET	362
C		CHSET	363
800	CONTINUE	CHSET	364
C		CHSET	365
	NAMELIST/CHKIN/XLO,DOO,DIO,RAUMOL,PHI,EGEN,XCL,NROWH,	CHSET	366
	1 NHOLLS,AREAH,11PL,BELT,TWOGJ,UTCA,DIOR,T2CR,12UX,TWODR,PHIO,	CHSET	367
	2 RO,KMOU,10,UVAXIS,UVAXII,UCONS,ATP,UCT,VTSC,HK,DIAM1,DIAM2,	CHSET	368
	3 DIS1,DIS2,DIS3,DIS4,OHED,IGNIT,AGLA,CGEN,	CHSET	369
	4 DIAMBT,CHAM2,MUFLD,BORER,FOREA,AKLAAX,AREACH,	CHSET	370
	5 BPRADD,BPDENS,ADENBF,RO,KRO,CHWT,EPCHG,	MOD1215	3
	6 UKAM,CAP,XLBEL,UA,1BEGC,1REGB,IEND,	CHSET	372
	7 BELDCH,ADD,BLEND,FRACT1,FRACT2,VOLCHG,RHCB,	CHSET	373
	8 RHOBP1,RHOBP2,RHOBP3,NGT,NG13,L1A,1BEG2,L1M2,1REG3,	CHSET	374
	9 TOPGAP,DAVG,CAP,AREAH2,TOTM,DM,CBCNS,AH,FRACT,IPS1,IPS2	CHSET	375
	IF (1DEBGG(4)) WRITE(6,CHKIN)	CHSET	376
C		CHSET	377
C		CHSET	378
C		CHSET	379
	RETURN	CHSET	380
C		CHSET	381
	2000 FORMAT(//,' LOGIC FOR FINDING TOTM WHEN CHAP1 IS TRUE IS NOT WRITTEN &LEN YL1')	CHSET	382
	2001 FORMAT(//,' ARRAY PHIBG',/)	CHSET	383
	2007 FORMAT(/,' RADIAL ROW',I2,/, (20X,10F10.6))	CHSET	384
	2000 FORMAT(//,' ARRAY PHIBP',/)	CHSET	385
	END	CHSET	386
		CHSET	387

SUBROUTINE PATHS

COMMON/CHAP/IX,IX,AX,AX,NGX,NGX,IBEG,IBEG,IPATH(60,5),AREAG(5),
 1 AREACH,AREACH(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,
 2 AREAR(60),AREARX,CHAM1,CHAM2,CHAM3,LOPGAP,AREAGP(60),DAYG,
 3 AREAR2,DIAM1,DIAM2,IBEG,IBEG,IPS1,IPS2,KADFS,BPIGN
 COMMON/IFPRINT,MOUCH,MOUCH,PR11,IBEG(35)
 LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
 LOGICAL PR11,IBEG

PATHS 2
 CHAMIX 2
 CHAMIX 3
 CHAMIX 4
 CHAMIX 5
 P 2
 CHAMLOG 2
 PLOG 2
 PATHS 7
 PATHS 8
 PATHS 9
 PATHS 10
 PATHS 11
 PATHS 12
 PATHS 13
 PATHS 14
 PATHS 15
 PATHS 16
 PATHS 17
 PATHS 18
 PATHS 19
 PATHS 20
 PATHS 21
 PATHS 22
 PATHS 23
 PATHS 24
 PATHS 25
 PATHS 26
 PATHS 27
 PATHS 28
 PATHS 29
 PATHS 30
 PATHS 31
 PATHS 32
 PATHS 33
 PATHS 34
 PATHS 35
 PATHS 36
 PATHS 37
 PATHS 38
 PATHS 39
 PATHS 40
 PATHS 41
 PATHS 42
 PATHS 43
 PATHS 44
 PATHS 45
 PATHS 46
 PATHS 47
 PATHS 48
 PATHS 49
 PATHS 50
 PATHS 51
 PATHS 52
 PATHS 53
 PATHS 54
 PATHS 55

SUBROUTINE PATHS INITIALIZES ARRAY IPATH.

VALUES OF IPATH CORRESPOND TO THE PAIR SUBROUTINES IN THE
 FOLLOWING WAY--

1 - AXIS 2 - AXIT 3 - INTER 4 - BNDY
 5 - FSURFA 6 - FSURFI 7 - FSURFI 8 - FSURFB
 9 - BSURFA 10 - BSURFI 11 - SURFI 12 - BSURFB

IF CHAM2 IS TRUE THE CHAMBER CONSISTS OF TWO ONE-DIMENSIONAL ROWS,
 ONE ROW USING AXIS ROUTINES, THE OTHER USING AXIT ROUTINES.

IF CHAM3 IS TRUE THE CHAMBER CONSISTS OF THREE ONE-DIMENSIONAL
 ROWS, TWO ROWS LIKE THOSE WHEN CHAM2 IS TRUE AND THE THIRD USING
 SIMILAR ROUTINES. VALUES OF IPATH CORRESPOND TO THE NEW ROUTINES
 AS FOLLOWS--

2 - AXIT2,AXIT3
 9 - BSURFA2
 10 - BSURFI2,BSURFI3

IF (CHAM2 .OR. CHAM3) GO TO 20

WHEN I=1, BSURFA, FSURFI, AND BSURFB ARE CALLED.

IPATH(1,1) = 9

J = 1

20 IPATH(1,J) = 11

J = J+1

IF (J.LT.NGX) GO TO 10

IPATH(1,J) = 12

FROM I=2 UNTIL THE BELL IGNITER TUBE, AXIS, INTER, AND BNDY
 ARE CALLED.

I=2

20 IF (I.GE.IBEG) GO TO 40

IPATH(1,1) = 1

J=2

30 IPATH(1,J) = 3

J=J+1

IF (J.LT.NGX) GO TO 30

IPATH(1,J) = 4

I=I+1

GO TO 20

FOR VALUES OF I WHERE THE BELL IGNITER TUBE IS AXIS, AXIT, INTER,
 AND BNDY ARE CALLED.

FOR THE GRID ON THE SURFACE OF THE BELL IGNITER TUBE AT EITHER
 END, INTER IS CALLED.

40 IF (I.GE.IBEG) GO TO 60

IPATH(1,1) = 1

IPATH(1,2) = 2

J=3

50	IPATH(1,J) = 3	PATHS	56
	J=J+1	PATHS	57
	IF(J.LT.NGX) GO TO 50	PATHS	58
	IPATH(1,J) = 4	PATHS	59
	I=I+1	PATHS	60
	GO TO 40	PATHS	61
C		PATHS	62
C	AFTER THE BELL IGNITER TUBE IS PASSED, AXIS, INT R, AND BNDY ARE	PATHS	63
C	CALLED.	PATHS	64
60	IF(1.01.NGX) GO TO 90	PATHS	65
	IPATH(1,1) = 1	PATHS	66
	J=2	PATHS	67
70	IPATH(1,J) = 3	PATHS	68
	J=J+1	PATHS	69
	IF(J.LT.NGX) GO TO 70	PATHS	70
	IPATH(1,J) = 4	PATHS	71
	I=I+1	PATHS	72
	GO TO 60	PATHS	73
C		PATHS	74
C		PATHS	75
C	THE FOLLOWING LOGIC IS USED FOR BOTH CHAM2 AND CHAM3 TRUE	PATHS	76
80	CONTINUE	PATHS	77
	IPATH(1,1) = 9	PATHS	78
	IPATH(1,2) = 10	PATHS	79
	IPATH(1,3) = 10	PATHS	80
	DO 85 I=2,NGX	PATHS	81
	IPATH(1,1) = 1	PATHS	82
	IPATH(1,2) = 2	PATHS	83
	IPATH(1,3) = 2	PATHS	84
85	CONTINUE	PATHS	85
C		PATHS	86
C		PATHS	87
90	CONTINUE	PATHS	88
	IF(.NOT. IDEBUG(5)) RETURN	PATHS	89
	WRITE(6,2000)	PATHS	90
	DO 95 J=1,NGX	PATHS	91
	WRITE(6,2001) J,(IPATH(I,J),I=1,NGX)	PATHS	92
95	CONTINUE	PATHS	93
	RETURN	PATHS	94
C		PATHS	95
2000	FORMAT(///,' ARRAY IPATH',/)	PATHS	96
2001	FORMAT(' RADIAL ROW',I2,/,10X,30(14),/,10X,20(14))	PATHS	97
	END	PATHS	98

SUBROUTINE HOLSET	HOLSET	2
COMMON/CHAN/IX,IXL,AD,SD,NOX,NGR,IPLUG,IENDR,IPATH(60,5),AREAG(5),	CHAMIX	2
1 AREACH,AREAC(60),ICUNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
2 AREAR(60),AREAA,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMIX	4
3 AREAR2,DIAMBT,BELBEG,BELBEG,IPS1,IPS2,KAUFS,BPIGN	CHAMIX	5
COMMON/EUNS/EULA,I2OR,I2UX,TWOTDR,DILN,HMB,IWOGJ,DVAXIS,DVAXIT,	EUNS	2
4 UX,UX,UX,UX,IWOT,HBP	EUNS	3
COMMON/HOLEZ/BELBOL(25),BROOF,NHOLES(25),XCL(25),AREAH(60),	HOLEA	2
5 AH(60),FRAC(60)	HOLEA	3
DIMENSION DTEMP(50),NTEMP(50),RTEMP(50)	HOLSET	6
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C	HOLSET	6
C THE DISTANCE (IN INCHES) BETWEEN THE CENTER LINE OF HOLES IN ROW I	HOLSET	9
C AND THE CENTER LINE OF THE PREVIOUS ROW OF HOLES WAS INPUT INTO	HOLSET	10
C XCL(I). XCL(I) CONTAINS THE DISTANCE OF THE CENTER LINE OF HOLES	HOLSET	11
C IN ROW I FROM THE BEGINNING OF THE BELL TUBE.	HOLSET	12
C NUMBER OF HOLES IN ROW I WAS INPUT INTO NHOLES(I).	HOLSET	13
C RADIUS OF HOLES IN ROW I WAS INPUT INTO RADHOL(I).	HOLSET	14
C PSEUDO HOLES WILL BE PUT BEFORE AND AFTER THE BELL TUBE AND	HOLSET	15
C TREATED THE SAME AS HOLES ON THE BELL TUBE.	HOLSET	16
C THE POSITION OF THE CENTER LINE OF HOLES IN ROW I (IN FEET)	HOLSET	17
C WILL BE PUT INTO XCL(I).	HOLSET	18
C	HOLSET	19
C TEMPORARILY STORE THE INPUT DATA ABOUT THE HOLES.	HOLSET	20
IF (NRHOW .EQ. 0) GO TO 102	HOLSET	21
DO 100 I=1,NRHOW	HOLSET	22
DTEMP(I) = XCL(I)/12.	HOLSET	23
NTEMP(I) = NHOLES(I)	HOLSET	24
RTEMP(I) = RADHOL(I)/12.	HOLSET	25
100 CONTINUE	HOLSET	26
102 CONTINUE	HOLSET	27
C	HOLSET	28
C PUT PSEUDO HOLES BEFORE THE BELL TUBE.	HOLSET	29
C KAUPS, THE RADIUS OF THE PSEUDO HOLES, IS CALCULATED TO GIVE ABOUT	HOLSET	30
C TWICE AS MUCH AREA AS HOLES ON THE BELL TUBE.	HOLSET	31
KAUPS=.1*SQRT(DIAMBT*UX)	MOD925	48
IPS1 = IBEGB	HOLSET	33
C	HOLSET	34
DO 105 I=1,IPS1	HOLSET	37
XCL(I)=FLOAT(I-1)*UX+UX/4.	MOD925	49
NHOLES(I) = 1	HOLSET	39
RADHOL(I) = KAUPS	HOLSET	40
105 CONTINUE	HOLSET	41
IF (BELBEG.LT.0X)RADHOL(1)=RADHOL(1)*S.QT(BELBEG/UX)	HOLSET	42
C	HOLSET	43
C SET UP HOLES ON THE BELL TUBE.	HOLSET	44
IF (NRHOW .EQ. 0) GO TO 112	HOLSET	45
I1 = IPS1 + 1	HOLSET	46
XCL(I1) = BELBEG + DTEMP(1)	HOLSET	47
NHOLES(I1) = NTEMP(1)	HOLSET	48
RADHOL(I1) = RTEMP(1)	HOLSET	49
IF (NRHOW .EQ. 1) GO TO 115	HOLSET	50
DO 110 I=2,NRHOW	HOLSET	51
I1 = I1 + 1	HOLSET	52
XCL(I1) = XCL(I1-1) + DTEMP(I)	HOLSET	53
NHOLES(I1) = NTEMP(I)	HOLSET	54
RADHOL(I1) = RTEMP(I)	HOLSET	55

IF(ACL(11) + RADHOL(11) .LT. BELLEND) GO TO 110	HOLSET	56
ILAST = 1 - 1	HOLSET	57
WRITE(6,2010) NKROWH,ILAST	HOLSET	58
11 = 11 - 1	HOLSET	59
GO TO 115	HOLSET	60
110 CONTINUE	HOLSET	61
112 IF(NKROWH .EQ. 0) 11 = IPS1	HOLSET	62
C	HOLSET	63
C THE LAST VALUE OF 11 IS THE NUMBER OF HOLES SO FAR.	HOLSET	64
C SET UP PSEUDO HOLES AFTER THE BELL TUBE.	HOLSET	65
115 NKROWH=11	HOLSET	66
IPS2 = IENDB	HOLSET	67
C	HOLSET	68
C IF THE BELL TUBE EXTENDS BEYOND GRID POINT IENDB, THERE SHOULD NOT	HOLSET	69
C BE A PSEUDOHOLE AT GRID POINT IENDB.	HOLSET	70
IF(BELLEND .GT. FLOAT(IENDB-1)*DX) IPS2 = IENDB + 1	HOLSET	71
DO 120 I=IPS2,NKX	HOLSET	72
NKROWH=NKROWH+1	HOLSET	73
ALL(NKROWH)=FLOAT(I-1)*UX	HOLSET	74
NHOLE(NKROWH)=1	HOLSET	75
RADHOL(NKROWH) = RADPS	HOLSET	76
IF(NKROWH.EQ. 85)GO TO 140	HOLSET	77
120 CONTINUE	HOLSET	78
C	HOLSET	79
C CLEAR HOLE ARRAY ENTRIES WHERE THERE ARE NO HOLES	HOLSET	80
NK1 = NKROWH + 1	HOLSET	81
CALL CLEAR(ACL(NK1),ALL(85))	HOLSET	82
CALL CLEAR(RADHOL(NK1),RADHOL(85))	HOLSET	83
DO 130 I=NK1,85	HOLSET	84
NHOLE(I) = 0	HOLSET	85
130 CONTINUE	HOLSET	86
C	HOLSET	87
140 CONTINUE	HOLSET	88
C	HOLSET	89
RETURN	HOLSET	90
2010 FORMAT(//,15,' WAS INPUT AS THE NUMBER OF ROWS OF HOLES ON THE BEL	HOLSET	91
SE TUBE, BUT ONLY ',14,' FIT ON THE TUBE')	HOLSET	92
END	HOLSET	93

SUBROUTINE HOLES		HOLES	2
C*	COMMON/CHAM/IX,IX,XL,FB,NOX,NOR,IBLOC,LEND,IPATH(60,5),AREAG(5),	HOLES	3
	AREAL(60),AREAL(60),ICN1,ONED,DIAM1,CIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	2
1	AREAL(60),AREAL(60),CHAM1,CHAM2,CHAM3,ICPGAP,AREAGP(60),DAVG,	CHAMIX	3
2	AREAL2,DIAM2,ICCN2,ICCN2,IPS1,IPS2,RADES,BPIGN	CHAMIX	4
3	COMMON/GRS/IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	CHAMIX	5
4	IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	FUNS	2
5	IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	LYNS	3
6	COMMON/HOLE/IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	HOLEA	2
7	IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	HOLEA	3
8	IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	HOLES	7
9	IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,	CHAMLOG	2
10	DATA F1/3.1415927	HOLES	9
C		HOLES	10
C		HOLES	11
C*	SUBROUTINE HOLES CALCULATES THE HOLE AREA EXPOSED TO EACH AXIAL	HOLES	12
C*	GRID. IT FORMS ARRAY AREAH. AREAH(I) GIVES THE EXPOSED HOLE	HOLES	13
C*	AREA AT GRID I. THE HOLE AREA EXPOSED TO A GRID IS CALCULATED	HOLES	14
C*	AS THE HOLE AREA LEFT OF THE RIGHT BOUNDARY OF THE GRID MINUS	HOLES	15
C*	THE HOLE AREA LEFT OF THE LEFT BOUNDARY OF THE GRID.	HOLES	16
C*		HOLES	17
C*	NRROW = NUMBER OF ROWS OF HOLES	HOLES	18
C*	ACCL(I) = CENTER LINE OF HOLES IN ROW I	HOLES	19
C*	RADHOL(I) = RADIUS OF HOLES IN ROW I	HOLES	20
C*	TA(I) = HOLE AREA OF HOLE ROW I THAT ALREADY IS ASSIGNED TO	HOLES	21
C*	A GRID. INITIALLY TA(I) IS SET AS 0. AFTER AREAXP IS	HOLES	22
C*	CALCULATED, TA(I) IS SET TO AREAXP.	HOLES	23
C*		HOLES	24
C		HOLES	25
	CALL CLEAR(TA(1),TA(65))	HOLES	26
	CALL CLEAR(AREAH(1),AREAH(60))	HOLES	27
C		HOLES	28
C*	I1 INDICES THE GRIDS	HOLES	29
	X = -DX	HOLES	30
	DO 40 I1=1,NX	HOLES	31
C*		HOLES	32
C*	X IS THE COORDINATE OF THE CENTER OF GRID I1	HOLES	33
	X = X + DX	HOLES	34
	GRUEND = X + .5*DX	HOLES	35
C*		HOLES	36
C*	DETERMINE THE AREA OF EACH ROW OF HOLES THAT LIES BEFORE THE	HOLES	37
C*	END OF THE GRID	HOLES	38
C		HOLES	39
C*	I2 INDICES THE ROWS OF HOLES	HOLES	40
	DO 30 I2 = 1,NRROW	HOLES	41
C*		HOLES	42
C*	IF THE END OF GRID I1 IS TO THE LEFT OF HOLES IN ROW I2 (AND THUS	HOLES	43
C*	THE REST OF THE ROWS) NO MORE HOLE AREA IS EXPOSED TO GRID I1	HOLES	44
	IF (GRUEND .LT. ACCL(I2) - RADHOL(I2)) GO TO 40	HOLES	45
	FEN = FLOAT(NHOLES(I2))	HOLES	46
C		HOLES	47
C*	IF THE END OF THE GRID IS TO THE RIGHT OF THE HOLES IN ROW I2, ADD	HOLES	48
C*	THE ENTIRE AREA OF THE HOLES IN ROW I2 TO AREAXP	HOLES	49
C*	OTHERWISE THE END OF THE GRID LIES WITHIN HOLES IN ROW I2 AND	HOLES	50
C*	THE AMOUNT OF EXPOSED AREA WILL BE DETERMINED.	HOLES	51
	IF (GRUEND .LT. ACCL(I2) + RADHOL(I2)) GO TO 10	HOLES	52
	AREAXP = FEN*PI*RADHOL(I2)*RADHOL(I2)	HOLES	53

GO TO 20		HOLES	54
C		HOLES	55
C		HOLES	56
10	TERM = GRDEND - XCL(12)	HOLES	57
C	FIND HALF OF THE CENTRAL ANGLE TO THE CHORD AT THE GRID END	HOLES	58
	THETA = ACOS(ABS(TERM)/RADHOL(12))	HOLES	59
C		HOLES	60
C	FIND AREA OF CIRCULAR SECTOR WHOSE CENTRAL ANGLE IS 2*THETA AND	HOLES	61
C	AREA OF TRIANGLE FORMED BY THE CHORD AND RADIUS	HOLES	62
	RSQ = RADHOL(12)*RADHOL(12)	HOLES	63
	AREAS = THETA*RSQ	HOLES	64
	AREAT = SQRT(RSQ - TERM*TERM)*ABS(TERM)	HOLES	65
C		HOLES	66
C	EXPOSED AREA FROM ONE HOLE IS SECTOR AREA - TRIANGLE AREA ON THE	HOLES	67
C	HOLE AREA MINUS THIS	HOLES	68
	AREAXP = (AREAS - AREAT)*FHH	HOLES	69
	IF(TERM .GT. 0.0) AREAXP = PI*RSQ*FHH - AREAXP	HOLES	70
C		HOLES	71
20	AREAH(11) = AREAH(11) + AREAXP - TAILC	HOLES	72
C		HOLES	73
	TAIL(2) = AREAXP	HOLES	74
30	CONTINUE	HOLES	75
C		HOLES	76
	IF(AREAH(11) .LT. 0.00001) AREAH(11) = 0.0	HOLES	77
40	CONTINUE	HOLES	78
C		HOLES	79
C		HOLES	80
C	CALCULATE AH(I), THE BELL TUBE AREA IN GRID I. AH(I) DOES NOT	HOLES	81
C	INCLUDE AREA OF PSEUDO-HOLES.	HOLES	82
	CALL CLEAR(AH(1),AH(60))	HOLES	83
	DO 60 I=IBEG, IEND	HOLES	84
	AH(I) = AREAH(I)	HOLES	85
60	CONTINUE	HOLES	86
C		HOLES	87
C	IF THERE IS A PSEUDO HOLE AT GRID IBEG OR AT GRID IEND, ITS	HOLES	88
C	AREA SHOULD NOT HAVE BEEN INCLUDED.	HOLES	89
	APSEUD = PI*RADPS*RADPS	HOLES	90
	IF(IPS2 .EQ. IEND) AH(IEND) = AH(IEND) - APSEUD	HOLES	91
	IF(AH(IEND) .LT. 0.) AH(IEND) = 0.	HOLES	92
	IN=IBEG+1	MOD0925	50
	DO 70 I=IN,NEX	MOD0925	51
C		MOD0925	52
C	CALCULATE FRACT(I), THE FRACTIONAL VOLUME OF A GRID NOT UNDER THE	HOLES	93
C	INFLUENCE OF THE BELL TUBE HOLES. THE VOLUME CONSIDERED TO BE	HOLES	94
C	INFLUENCED IS BASED ON THREE TIMES THE HEMISPHERICAL VOLUME	HOLES	95
C	COMPUTED FROM THE RADIUS OF A TUBE HOLE.	HOLES	96
	FRACT(I) = 1.0 - 2.0*SQRT(AH(I)*AH(I)*AF(I)/PI)/(AREAR(I)*DX)	HOLES	97
	IF(FRACT(I) .LT. 0.0) FRACT(I) = 0.0	HOLES	99
70	CONTINUE	HOLES	100
	RETURN	HOLES	101
	END	HOLES	102
		HOLES	103

```

SUBROUTINE AREAS
COMMON/DBNKLZ/DBNKLZ,XI,VP,BORED,BORER,BOREG,DT2BD,DTDSQ,XLRAK
COMMON/CHAP/IA,IB,IC,IG,NGX,NGR,IEGB,IEGBR,IPATH(60,5),AREAG(5),
1 AREACH,AREAC(60),IGR1,IGR2,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,
2 AREAR(60),AREARX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
3 AREAM2,DIAM1I,BELLEN,BELBEG,IPSI,IPSI2,RADES,BPIGN
COMMON/EQNS/IE1,IE2,IE3,IE4,IE5,IE6,IE7,HMB,TWOGU,DVAXIS,DVAXIT,
4 IEA,IEB,IEC,IED,IEF
COMMON/PR/PR11,PR12,PR13,PR14,PR15,PR16,PR17,PR18,PR19,PR20,PR21,PR22,PR23,PR24,PR25,PR26,PR27,PR28,PR29,PR30,PR31,PR32,PR33,PR34,PR35,PR36,PR37,PR38,PR39,PR40,PR41,PR42,PR43,PR44,PR45,PR46,PR47,PR48,PR49,PR50,PR51,PR52,PR53,PR54,PR55,PR56,PR57,PR58,PR59,PR60,PR61,PR62,PR63,PR64,PR65,PR66,PR67,PR68,PR69,PR70,PR71,PR72,PR73,PR74,PR75,PR76,PR77,PR78,PR79,PR80,PR81,PR82,PR83,PR84,PR85,PR86,PR87,PR88,PR89,PR90,PR91,PR92,PR93,PR94,PR95,PR96,PR97,PR98,PR99,PR100
COMMON/LOGICAL/IGR1,IGR2,IGR3,IGR4,IGR5,IGR6,IGR7,IGR8,IGR9,IGR10,IGR11,IGR12,IGR13,IGR14,IGR15,IGR16,IGR17,IGR18,IGR19,IGR20,IGR21,IGR22,IGR23,IGR24,IGR25,IGR26,IGR27,IGR28,IGR29,IGR30,IGR31,IGR32,IGR33,IGR34,IGR35,IGR36,IGR37,IGR38,IGR39,IGR40,IGR41,IGR42,IGR43,IGR44,IGR45,IGR46,IGR47,IGR48,IGR49,IGR50,IGR51,IGR52,IGR53,IGR54,IGR55,IGR56,IGR57,IGR58,IGR59,IGR60,IGR61,IGR62,IGR63,IGR64,IGR65,IGR66,IGR67,IGR68,IGR69,IGR70,IGR71,IGR72,IGR73,IGR74,IGR75,IGR76,IGR77,IGR78,IGR79,IGR80,IGR81,IGR82,IGR83,IGR84,IGR85,IGR86,IGR87,IGR88,IGR89,IGR90,IGR91,IGR92,IGR93,IGR94,IGR95,IGR96,IGR97,IGR98,IGR99,IGR100
COMMON/PR11,PR12,PR13,PR14,PR15,PR16,PR17,PR18,PR19,PR20,PR21,PR22,PR23,PR24,PR25,PR26,PR27,PR28,PR29,PR30,PR31,PR32,PR33,PR34,PR35,PR36,PR37,PR38,PR39,PR40,PR41,PR42,PR43,PR44,PR45,PR46,PR47,PR48,PR49,PR50,PR51,PR52,PR53,PR54,PR55,PR56,PR57,PR58,PR59,PR60,PR61,PR62,PR63,PR64,PR65,PR66,PR67,PR68,PR69,PR70,PR71,PR72,PR73,PR74,PR75,PR76,PR77,PR78,PR79,PR80,PR81,PR82,PR83,PR84,PR85,PR86,PR87,PR88,PR89,PR90,PR91,PR92,PR93,PR94,PR95,PR96,PR97,PR98,PR99,PR100
DATA FILE/,785596/

```

```

AREAS 2
AREAS 3
CHAMIX 2
CHAMIX 3
CHAMIX 4
CHAMIX 5
EQNS 2
EQNS 3
P 2
CHAMLOG 2
PLOG 2
AREAS 9
AREAS 10
AREAS 11
AREAS 12
AREAS 13
AREAS 14
AREAS 15
AREAS 16
AREAS 17
AREAS 18
AREAS 19
AREAS 20
AREAS 21
AREAS 22
AREAS 23
AREAS 24
AREAS 25
AREAS 26
AREAS 27
AREAS 28
AREAS 29
AREAS 30
AREAS 31
AREAS 32
AREAS 33
AREAS 34
AREAS 35
AREAS 36
AREAS 37
AREAS 38
AREAS 39
AREAS 40
AREAS 41
AREAS 42
AREAS 43
AREAS 44
AREAS 45
AREAS 46
AREAS 47
AREAS 48
AREAS 49
AREAS 50
AREAS 51
AREAS 52
AREAS 53
AREAS 54

```

```

SUBROUTINE AREAS CALCULATES CROSS-SECTIONAL AREAS AND VOLUME
INCREMENTS ASSOCIATED WITH THE CHAMBER.

AREAX - AREA OF BELL IGNITER TUBE
AREAG(1) - AREA OF GAP AT TOP OF CHAMBER AT AXIAL GRID 1
AREAC(1) - AREA OF ENTIRE CHAMBER AT AXIAL GRID 1 WHEN CHAMBER
BECOMES ONE DIMENSIONAL, SO THAT CALCULATIONS IN AXIS GO
SMOOTHLY. AREAC(1) IS INITIALLY SET TO 1.0 IF CHAM1 IS
TRUE AND AREAX OTHERWISE.
AREAR(1) - AREA OF ENTIRE CHAMBER MINUS THE BELL IGNITER TUBE
AT AXIAL GRID 1 IF CHAM1 OR CHAM2 IS TRUE. IF CHAM3 IS
TRUE, AREAGP(1) IS ALSO SUBTRACTED.
AREAG(0) - GENERALIZED AREA OF RADIAL GRID J, USED FOR
SETTING DUMMY BARREL GRID 1 PROPERTIES. IF CHAM2 IS
TRUE AREAG(1) IS AREAX AND AREAG(2) IS AREAR(NGX).
IF CHAM3 IS TRUE AREAG(1) AND AREAG(2) ARE THE SAME AS
WHEN CHAM2 IS TRUE AND AREAG(0) IS AREAGP(NGX). IF CHAM1
IS TRUE, AREAG(0) IS SET USING OR.
AREAC(1) = AREAX + . . . + AREAR(NGR)
IF CHAM1 IS TRUE, AREAC IS USED FOR MAKING THE CHAMBER ONE
DIMENSIONAL AT EVENT 1. IF CHAM2 IS TRUE, AREAX AND AREAR(1) ARE
USED AT GRID 1. IF CHAM3 IS TRUE, AREAGP(1) IS ALSO USED.
FOR DUMMIE AND DUMMIE TERMS VOLUME INCREMENTS ARE NEEDED.
DVAXIS - VOLUME INCREMENT OF GRIDS FOR J = 1.
DVAXIT - VOLUME INCREMENT OF GRIDS FOR J = 2. DVAXIT WOULD
NEED TO BE AN ARRAY FOR CHAM2 OR CHAM3 TRUE BUT IT IS NOT
PRODUCED THEN.

AREAX = PI*DIAM1*DIAM1/4
CALCULATE ARRAY AREAR AND FROM IT GET AREAR
CALCULATE ARRAY AREAGP IF CHAM3 IS TRUE.
ALSO CALCULATE AVERAGE DIAMETER TO BE USED IN CALCULATING PSEUDO-
NOSES IF CHAM3 IS TRUE.
DAVG = 0.0
TWOGAP = 2.0*TOPGAP
AP = 0.0

THE CHAMBER DIAMETER IS DIAM1 UP TO DIS1
DIAPSO = DIAM1*DIAM1
IF(CHAM3) D2 = DIAM1 - TWOGAP
IF(CHAM3) D2SO = 1.0*D2
DO 250 I=1,NGX

```

	ITEMP = 1	AREAS	55
	AREAC(1) = FIDF*DIAMSG	AREAS	56
	IF(CHAM3) AREAC(1) = AREAC(1) - FIDF*D2SQ	AREAS	57
	IF(CHAM3) DAVG = DAVG + DIAM1	AREAS	58
	XB = XB + DX	AREAS	59
	IF(XB .GT. DIS1) GO TO 260	AREAS	60
250	CONTINUE	AREAS	61
	GO TO 350	AREAS	62
C		AREAS	63
C		AREAS	64
C	THE CHAMBER DIAMETER DECREASES LINEARLY FROM DIS1 TO DIS2	AREAS	65
260	CONTINUE	AREAS	66
	I = ITEMP	AREAS	67
	II = I + 1	AREAS	68
	IF(ABS(DIS1 - DIS2) .GT. 0.00001) GO TO 265	AREAS	69
	SLOPE = 0.0	AREAS	70
	GO TO 268	AREAS	71
265	SLOPE = (DIAM1 - DIAM2)/(DIS1 - DIS2)	AREAS	72
268	CONTINUE	AREAS	73
	GO 270 I=II,NGA	AREAS	74
	ITEMP = 1	AREAS	75
	DIAM = DIAM1 + SLOPE*(XB - DIS1)	AREAS	76
	IF(CHAM3) D2 = DIAM - TWO GAP	AREAS	77
	AREAC(1) = FIDF*DIAM*DIAM	AREAS	78
	IF(CHAM3) AREAC(1) = AREAC(1) - FIDF*D2*D2	AREAS	79
	IF(CHAM3) DAVG = DAVG + DIAM	AREAS	80
	XB = XB + DX	AREAS	81
	IF(XB .GT. DIS2) GO TO 280	AREAS	82
270	CONTINUE	AREAS	83
	GO TO 350	AREAS	84
C		AREAS	85
C		AREAS	86
C	THE CHAMBER DIAMETER IS DIAM2 BETWEEN DIS2 AND DIS3	AREAS	87
280	CONTINUE	AREAS	88
	I = ITEMP	AREAS	89
	II = I + 1	AREAS	90
	DIAMSG = DIAM2*DIAM2	AREAS	91
	IF(CHAM3) D2 = DIAM2 - TWO GAP	AREAS	92
	IF(CHAM3) D2SQ = D2*D2	AREAS	93
	GO 290 I=II,NGA	AREAS	94
	ITEMP = 1	AREAS	95
	AREAC(1) = FIDF*DIAMSG	AREAS	96
	IF(CHAM3) AREAC(1) = AREAC(1) - FIDF*D2*D2	AREAS	97
	IF(CHAM3) DAVG = DAVG + DIAM2	AREAS	98
	XB = XB + DX	AREAS	99
	IF(XB .GT. DIS3) GO TO 300	AREAS	100
290	CONTINUE	AREAS	101
	GO TO 350	AREAS	102
C		AREAS	103
C		AREAS	104
C	THE CHAMBER DIAMETER DECREASES LINEARLY BETWEEN DIS3 AND DIS4	AREAS	105
300	CONTINUE	AREAS	106
	I = ITEMP	AREAS	107
	II = I + 1	AREAS	108
	IF(ABS(DIS3 - DIS4) .GT. 0.00001) GO TO 305	AREAS	109
	SLOPE = 0.0	AREAS	110
	GO TO 308	AREAS	111

310	SLOPE = (DIAM2 - DIAM1)/DIS3 - DIS4	AREAS	112
315	CONTINUE	AREAS	113
	GO 310 I=1,IGA	AREAS	114
	DIAM = DIAM2 + SLOPE*(XN - DIS3)	AREAS	115
	IF(CHAM3) LP = DIAM - TWO GAP	AREAS	116
	AREAC(I) = FID*DIAM*DIAM	AREAS	117
	IF(CHAM3) AREAGP(I) = AREAC(I) - FIDF*D2*D2	AREAS	118
	IF(CHAM3) LAVG = LAVG + DIAM	AREAS	119
	XN = XN + DX	AREAS	120
311	CONTINUE	AREAS	121
C		AREAS	122
310	CONTINUE	AREAS	123
	IF(CHAM3) LAVG = LAVG/FLOAT(IGA)	AREAS	124
	IGP1 = IGA + 1	AREAS	125
	AREAC(IGP1) = BOREA	AREAS	126
	IF(CHAM3) AREAGP(IGP1) = BOREA - FIDF*(BOREC - TWO GAP)**2	AREAS	127
C		AREAS	128
C	FOR VERSION 3 AREAC(I) IS CALCULATED DIFFERENTLY THAN FOR VERSIONS	AREAS	129
C	1 AND 2.	AREAS	130
	IF(CHAM3) GO TO 365	AREAS	131
C		AREAS	132
	GO 360 I=1,NGP1	AREAS	133
	AREAC(I) = AREAC(I) - AREAX	AREAS	134
360	CONTINUE	AREAS	135
	GO TO 375	AREAS	136
C		AREAS	137
365	CONTINUE	AREAS	138
	GO 370 I=1,NGP1	AREAS	139
	AREAC(I) = AREAC(I) - AREAX - AREAGP(I)	AREAS	140
370	CONTINUE	AREAS	141
C		AREAS	142
375	CONTINUE	AREAS	143
	IF(CHAM2 OR CHAM3) GO TO 400	AREAS	144
C		AREAS	145
C	CALCULATIONS WHEN CHAM1 IS TRUE	AREAS	146
	GO 380 I=1,NGP1	AREAS	147
	AREAC(I) = 1.0	AREAS	148
380	CONTINUE	AREAS	149
C		AREAS	150
	NGR1 = NGR - 1	AREAS	151
	FIDRSW = FIDR*LN	AREAS	152
	AREAC(1) = 0.25*FIDRSW	AREAS	153
	GO 390 J=2,NGR1	AREAS	154
	AREAC(J) = 2.0*FIDRSW*FLOAT(J-1)	AREAS	155
390	CONTINUE	AREAS	156
	AREAC(NGR) = FIDRSW*(FLOAT(NGR) - 1.0)	AREAS	157
	AREACH = FIDRSW*FLOAT(NGR1)**2	AREAS	158
C***	SHOUL THIS BE AREAC(1)*LA	AREAS	159
	LVAXIS = FIDRSW*LA	AREAS	160
C	LVAXIT IS FID*(N + 1/4)*LN*LN	AREAS	161
	LVAXIT = 1.25*LVAXIS	AREAS	162
	GO TO 300	AREAS	163
C		AREAS	164
C		AREAS	165
C	CALCULATIONS WHEN CHAM2 OR CHAM3 IS TRUE	AREAS	166
400	CONTINUE	AREAS	167
	GO 410 I=1,NGP1	AREAS	168

AREAC(1) = AREAAX	AREAS	169
410 CONTINUE	AREAS	170
AREAG(1) = AREAAX	AREAS	171
AREAG(2) = AREAR(NGX)	AREAS	172
IF(CHAM3) AREAG(3) = AREAGP(NGX)	AREAS	173
AREACH = AREAG(1) + AREAG(2)	AREAS	174
IF(CHAM3) AREACH = AREACH + AREAG(3)	AREAS	175
UVAXIS = AREAAX*UX	AREAS	176
	AREAS	177
	AREAS	178
500 CONTINUE	AREAS	179
	AREAS	180
IF(.NOT. IDEBUG(6)) RETURN	AREAS	181
WRITE(6,2002)	AREAS	182
WRITE(6,2003) (AREAR(I),I=1,NGP1)	AREAS	183
WRITE(6,2004)	AREAS	184
WRITE(6,2005) (AREAC(I),I=1,NGP1)	AREAS	185
WRITE(6,2005) (AREAG(I),I=1,NGR)	AREAS	186
IF(CHAM3) WRITE(6,2006)	AREAS	187
IF(CHAM3) WRITE(6,2003) (AREAGP(I),I=1,NGP1)	AREAS	188
RETURN	AREAS	189
	AREAS	190
2002 FORMAT(///,' ARRAY AREAR',//)	AREAS	191
2003 FORMAT(9X,10F11.7,//)	AREAS	192
2004 FORMAT(///,' ARRAY AREAC',//)	AREAS	193
2005 FORMAT(///,' ARRAY AREAG',///,20X,5F11.7)	AREAS	194
2006 FORMAT(///,' ARRAY AREAGP',//)	AREAS	195
END	AREAS	196

	CONE=((PINT-FLU)/(XLU-XINT))*AREAPL	MOD1111	17
C		BARSET	41
C	INITIALLY THE PROJECTILE IS NOT MOVING	BARSET	42
	XP = XUL	BARSET	43
	VP = 0.0	BARSET	44
	DXPKIM = DX	MOD0301	24
C		BARSET	45
C		BARSET	46
C	CONSTANTS FOR SUBROUTINE UNOLYR	BARSET	47
	BORLDB = BORED/8.0	BARSET	48
	DTUSG = DELT*FURED*BORED	BARSET	49
	DTZBD = -0.5*DELT/BORED	BARSET	50
C		BARSET	51
C		BARSET	52
C	CONSTANT FOR SUBROUTINE PROPMO	BARSET	53
	GRHOP = GRAV*DELT/RHOP	BARSET	54
C		BARSET	55
C	FINISH TALLING THE GAS MASS. NOTE THAT PHI IS 1.0 NOW.	BARSET	56
	IF(NX.EQ. 1) GO TO 30	BARSET	57
	IF(NX.EQ. 2) GO TO 25	BARSET	58
	TOTM = TOTM + (FLOAT(NX - 2) + 0.5)*RHUO*BOREA*DX	BARSET	59
	GO TO 30	BARSET	60
25	TOTM = TOTM + 0.5*RHOD*BOREA*DX	BARSET	61
30	CONTINUE	BARSET	62
C		BARSET	63
	NAMLIST/BARCHK/RADPB,XOB,DEPTH,BORL,XOB,CS,SN,CON1,CON2,	BARSET	64
	\$ CON3,CON4,CON5,AREAPB,XP,VP,BOREA,G,RHOP,FORER,BORLDB,DTUSG,	BARSET	65
	\$ DTZBD,TOTM	BARSET	66
	IF(IDEBUG(6)) WRITE(6,BARCHK)	BARSET	67
C		BARSET	68
C		BARSET	69
C	*****	BARSET	70
C	INITIALIZE BARREL ARRAYS	BARSET	71
C	*****	BARSET	72
	CALL CLEAR(PHI(1), UALNEM(100))	BARSET	73
	CALL CLEAR(XLB(1), UL1B(100))	BARSET	74
C		BARSET	75
	IF(NX.EQ. 1) GO TO 60	BARSET	76
	DO 50 I=2,NX	BARSET	77
	RHUG(I) = RHUO	BARSET	78
	HU(I) = HU	BARSET	79
	PG(I) = PG	BARSET	80
	UG(I) = UO	BARSET	81
	TG(I) = TO	BARSET	82
	AMASS(I) = BOREA	BARSET	83
	AMUM(I) = BOREA	BARSET	84
	ALNEM(I) = BOREA	BARSET	85
	UAMASS(I) = BOREA	BARSET	86
	UAMUM(I) = BOREA	BARSET	87
	UALNEM(I) = BOREA	BARSET	88
C		BARSET	89
C	PMOUT AND UP HAVE ALREADY BEEN CLEARED	BARSET	90
50	CONTINUE	BARSET	91
C		BARSET	92
C	SET AREAS AT BARREL GRID 1 TO HORE AREA OF BARREL.	BARSET	93
60	CONTINUE	BARSET	94
	AMASS(1) = BOREA	BARSET	95

AMOM(1) = MOKLA	HARSET	96
AMENK(1) = MOKLA	HARSET	97
JAMASS(1) = BOKLA	HARSET	98
JAMOM(1) = BOKLA	HARSET	99
JAMENK(1) = BOKLA	HARSET	100
	HARSET	101
	HARSET	102
SET ALL ENTRIES IN ARRAYS PHI AND UPHI TO 1.0 SO THAT WHEN GRIDS ARE	HARSET	103
THERE WILL BE NO PROPELLANT IN THEM.	HARSET	104
DO GO I=1,100	HARSET	105
PHI(I) = 1.0	HARSET	106
UPHI(I) = 1.0	HARSET	107
GO CONTINUE	HARSET	108
RETURN	HARSET	109
END	HARSET	110

SUBROUTINE MFLOW		MFLOW	2
C		MFLOW	3
C	COMMON/CHAM/IX,IR,IB,NGX,NGR,IBEGE,IBNDB,IPATH(60,5),AREAG(5),	MFLOW	4
C	AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,	CHAMIX	2
C	AREAR(60),AREARX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),DAVG,	CHAMIX	3
C	AREAR2,DIARH1,BELLEN1,BELBEG,IPS1,IPS2,HADFS,HPIGN	CHAMIX	4
C	COMMON/GASCON/KQ,RNO,CVU,CVH	CHAMIX	5
C	COMMON/CLOCK/TIME,DEL.T	MFLOW	6
C	COMMON/HOLEX/RADHOL(25),NROWH,NHOLES(25),XCL(85),AREAH(60),	MUD202	2
C	AH(60),FRACT(60)	HOLEA	2
C	COMMON/INPUTS/C1,C2,C3,C4,T,TIGN,OLCNS,RHOF,PHIO,TF,CA,RHUO,	HOLEA	3
C	HU,PU,DU,GTIRHP,RP,GR,TIGMBP,QBCONS,TOTM	INPUTS	2
C	COMMON/BAG/PHIBG(60,5),RHOBG(60,5),PBG(60,5),UBG(60,5),	INPUTS	3
C	1 VBG(60,5),UPB(60,5),PCB(60,5),T2C(60,5),	BAG	2
C	2 DOTMIG(60),QBAG(60,5),XCRAG(60,5),DOTMB(60,5),UPBDT(60,5),	BAG	3
C	3 PHIBDT(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	4
C	4 VBGTD(60,5),TBC(60,5),DOTMIG(60),CJMP(60,5),PHIRP(60,5),	BAG	5
C	5 PHIPTD(60,5),TZK(60),TBP(60,5)	BAG	6
C	DIMENSION DOTM(60)	BAG	7
C	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,HPIGN	MFLOW	10
C		CHAMLOG	2
C		MFLOW	12
C		MFLOW	13
C		MFLOW	14
C	SUBROUTINE MFLOW CALCULATES THE MASS OF GAS FLOWING BETWEEN TWO	MFLOW	15
C	RADIAL ROWS THROUGH HOLES IN THE BELL IGNITER TUBE AND THROUGH	MFLOW	16
C	PSEUDO HOLES. MFLOW CALCULATES THE ENTRIES OF ARRAY DOTMIG AND IF	MFLOW	17
C	CHAM3 IS TRUE, THE ENTRIES OF ARRAY DOTMBG.	MFLOW	18
C	DOTMIG(I) IS THE GAS MASS FLOWING BETWEEN GRIDS (1,1) AND (1,2).	MFLOW	19
C	+ DOTMIG(I) OCCURS IN THE PATH ROUTINES FOR GRIDS WHERE J=1 AND	MFLOW	20
C	- DOTMIG(I) WHERE J=2.	MFLOW	21
C	DOTMBG(I) IS THE GAS MASS FLOWING BETWEEN GRIDS (1,2) AND (1,3)	MFLOW	22
C	WHEN RADIAL ROWS 2 AND 3 ARE CONSIDERED INDEPENDENT ONE-DIMENSIONAL	MFLOW	23
C	SYSTEMS.	MFLOW	24
C	+ DOTMBG(I) OCCURS IN THE PATH ROUTINES FOR GRIDS WHERE J=2 AND	MFLOW	25
C	- DOTMBG(I) OCCURS WHERE J=3.	MFLOW	26
C	ARRAYS DOTMIG AND DOTMBG ARE CLEARED IN MAIN AT EACH TIME INTERVAL.	MFLOW	27
C		MFLOW	28
C	NGA1 = NGA - 1	MFLOW	29
C	ID = 1	MFLOW	30
C	IF (CHAM3) ID = 2	MFLOW	31
C	J = 1	MFLOW	32
C	GO TO 20	MFLOW	33
C		MFLOW	34
C	ID = 1	MFLOW	35
C	J = 2	MFLOW	36
C		MFLOW	37
C	20 CONTINUE	MFLOW	38
C	JP1 = J + 1	MFLOW	39
C	CALL CLEAR(DOTM(1),DOTM(60))	MFLOW	40
C		MFLOW	41
C		MFLOW	42
C	*****	MFLOW	43
C	COMPUTE GAS MASS FLOW PER UNIT AREA	MFLOW	44
C	*****	MFLOW	45
C	DO 60 I=1,NGA	MFLOW	46
C		MFLOW	47

	PR = PCH(1,J)	MFLOW	48
	PA = PCH(1,JP1)	MFLOW	49
C		MFLOW	50
C*	IF PR=PA THERE IS NO FLOW	MFLOW	51
	IF (ABS(PR - PA) .LT. 0.001) GO TO 80	MFLOW	52
C		MFLOW	53
C*	DETERMINE THE DIRECTION OF THE FLOW. IF PR IS GREATER THAN PA,	MFLOW	54
C*	GAS FLOWS OUT OF THE TUBE. OTHERWISE IT FLOWS INTO THE TUBE.	MFLOW	55
	IF (PR .LT. PA) GO TO 40	MFLOW	56
	HRR = HBG(1,J)	MFLOW	57
C		MFLOW	58
C*	PCOMP IS AN APPROXIMATE STATIC PRESSURE FOR CHOKED (MACH NO. 1)	MFLOW	59
C*	FLOW. ASSUME PCOMP IS NOT LESS THAN PA.	MFLOW	60
	PCOMP = 0.53*PR	MFLOW	61
	IF (PCOMP .LT. PA) PCOMP = PA	MFLOW	62
C		MFLOW	63
	CALL GSPROP(RU,RRU,R,CV0,CVH,CV,FLOMP,HRR,TDUM,RHODUM,	MFLOW	64
3	UBG(1,J),0.0,GMM,CP,3)	MFLOW	65
	GMM1 = GMM - 1.0	MFLOW	66
	CONS1 = 2.0/(GMM + 1.0)	MFLOW	67
	PWR = GMP/GMM1	MFLOW	68
	CONS2 = 2.0/GMM1	MFLOW	69
C		MFLOW	70
C*	PSTAT IS SONIC STATIC PRESSURE	MFLOW	71
	PSTAT = PR*CONS1**PWR	MFLOW	72
C		MFLOW	73
C*	FM IS MACH NUMBER. IF PA IS LESS THAN PSTAT, FM=1. OTHERWISE	MFLOW	74
C*	FM IS NOT 1. AND MUST BE CALCULATED.	MFLOW	75
	FM = 1.0	MFLOW	76
	IF (PA .LT. PSTAT) GO TO 10	MFLOW	77
	FM = SQRT(((PR/PA)**(GMM1/GMP) - 1.0)*CONS2)	MFLOW	78
	PSTAT = PA	MFLOW	79
C		MFLOW	80
10	HSTAT = HRR/(1.0 + FM*FM/CONS2)	MFLOW	81
C		MFLOW	82
C	SINCE GAS IS FLOWING FROM GRID (I,J) TO GRID (I,J+1), DOTH SHOULD	MFLOW	83
C	BE NEGATIVE.	MFLOW	84
C***	IS PHIBG(I,2) OKAY	MFLOW	85
	DOTH(1) = -.203*GMM*PSTAT*FM*CA/SQRT(GMM1*HSTAT*PHIBG(I,2))	MFLOW	86
	GO TO 80	MFLOW	87
C		MFLOW	88
C		MFLOW	89
C		MFLOW	90
C*	THE CODING WHEN PR IS LESS THAN PA IS ESSENTIALLY THE SAME AS ABOVE	MFLOW	91
C	WITH PR AND PA INTERCHANGED.	MFLOW	92
40	CONTINUE	MFLOW	93
	HAA = HBG(1,JP1)	MFLOW	94
	PCOMP = 0.53*PA	MFLOW	95
	IF (PCOMP .LT. PR) PCOMP = PR	MFLOW	96
	CALL GSPROP(RU,RRU,R,CV0,CVH,CV,FLOMP,HAA,TDUM,RHODUM,	MFLOW	97
3	UBG(1,JP1),0.0,GMM,CP,3)	MFLOW	98
C		MFLOW	99
	GMM1 = GMM - 1.0	MFLOW	100
	CONS1 = 2.0/(GMM + 1.0)	MFLOW	101
	PWR = GMP/GMM1	MFLOW	102
	CONS2 = 2.0/GMM1	MFLOW	103
C		MFLOW	104

	PSIAT = PA*CONS1**PMR	MFLOW	105
C		MFLOW	106
	PM = 1.0	MFLOW	107
	IF (PM .LT. PSIAT) GO TO 50	MFLOW	108
	PM = SQRT(((PA/PM)**(GMM1/GPM) - 1.0)*CONS2)	MFLOW	109
	PSIAT = PM	MFLOW	110
50	HSTAT = HAA/(1.0 + PM*PM/CONS2)	MFLOW	111
		MFLOW	112
C	SINCE GAS IS FLOWING FROM GRID (I,J+1) TO GRID (I,J),DOTM SHOULD	MFLOW	113
C	BE POSITIVE,	MFLOW	114
C****IS PHIBG(I,2) OKAY		MFLOW	115
	DOTM(I) = .203*GMM*PSIAT*PM*CA/SQRT(GMM1*HSTAT*PHIBG(I,2))	MFLOW	116
C		MFLOW	117
80	CONTINUE	MFLOW	118
	IF(CHANS .AND. ID .EQ. 1) GO TO 120	MFLOW	119
C		MFLOW	120
C		MFLOW	121
C*****		MFLOW	122
C	FILL ARRAY DOTMIG	MFLOW	123
C*****		MFLOW	124
C		MFLOW	125
C	AVERAGE GAS MASS FLOW AT EACH GRID.	MFLOW	126
	DOTMIG(I) = 0.5*(DOTM(I) + DOTM(2))*AREAH(I)	MFLOW	127
	DO 100 I=2,NGX1	MFLOW	128
	DOTMIG(I) = 0.25*(DOTM(I-1) + DOTM(I) + DOTM(I) + DOTM(I+1))*	MFLOW	129
	AREAH(I)	MFLOW	130
100	CONTINUE	MFLOW	131
	DOTMIG(NGX) = 0.5*(DOTM(NGX1) + DOTM(NGX))*AREAH(NGX)	MFLOW	132
C		MFLOW	133
	GO TO (200,5),ID	MFLOW	134
C		MFLOW	135
C		MFLOW	136
C*****		MFLOW	137
C	FILL ARRAY DOTMBG	MFLOW	138
C*****		MFLOW	139
C		MFLOW	140
120	CONTINUE	MFLOW	141
	DOTMBG(I) = 0.5*(DOTM(I) + DOTM(2))*AREAH2	MFLOW	142
	DO 130 I=2,NGX1	MFLOW	143
	DOTMBG(I) = 0.25*(DOTM(I-1) + DOTM(I) + DOTM(I) + DOTM(I+1))*	MFLOW	144
	AREAH2	MFLOW	145
130	CONTINUE	MFLOW	146
	DOTMBG(NGX) = 0.5*(DOTM(NGX1) + DOTM(NGX))*AREAH2	MFLOW	147
C		MFLOW	148
200	RETURN	MFLOW	149
C		MFLOW	150
	END	MFLOW	151

[illegible]

STDOFF=DELBE6	MOD0605	17
BGBHN=.TRUE.	MOD0925	80
DWDP=CLP(1)*(DIAMBT/DLP(1))*2.	MOD0925	81
PHIBP(IBE6FP,1) = 1.0	MOD0925	82
PHIBP(IBE6FP + 1,1) = 1.	MOD1115	3
PHIBP(IBE6H,1)=PHIBP(IBE6B,1)-DWDP/DFLENS*AREAAX*DX	MOD1115	4
PHIP1U(IBE6FP+1,1) = PHIBP(IBE6FP+1,1)	MOD0925	86
PHIP1U(IBE6FP,1)=PHIBP(IBE6FP,1)	MOD0925	87
PHIP1U(IBE6H,1)=PHIBP(IBE6H,1)	MOD1115	5
30 CONTINUE	MOD0605	23
FLAG=0.	MOD0925	88
DO 100 J=1,NCR	BPFIR	17
IF(J .EQ. 2) FLAG=0.	MOD0925	89
DO 100 I=1,NCX	BPFIR	18
IF(PHIBP(I,J) .GE. .999) GO TO 100	MOD1002	5
IF(TBP(I,J) .GE. TIGNBP) GO TO 100	BPFIR	20
BFI0N = .FALSE.	BPFIR	21
C	BPFIR	22
C GAS TEMPERATURE	BPFIR	23
TTX = TBO(I,J)	BPFIR	24
TLFP = TTX*SQRT(TTX)	BPFIR	25
C	BPFIR	26
C GAS VISCOSITY	BPFIR	27
TXMU = C1*TEMP/(TTX + C2)	BPFIR	28
C	BPFIR	29
C THERMAL CONDUCTIVITY OF GAS	BPFIR	30
TXK = C3*TEMP/(TTX + C4)	BPFIR	31
C	BPFIR	32
C REYNOLDS NUMBER	BPFIR	33
RETXBP = KNOBG(I,J)*ABS(UBG(I,J))*BPRAD(I,J)/TXMU	BPFIR	34
IF(RETXBP .LT. .00001) RETXBP=.00001	MOD1017	3
C	BPFIR	35
C RUSSELL NUMBER	BPFIR	36
TANUBP = 0.3*RETXBP**0.62 + 0.5*SQRT(PCF(I,J)/144.)	BPFIR	37
C	BPFIR	38
C BPRAD(I,J) SHOULD NOT BE LESS THAN 0.001, BECAUSE OTHERWISE	BPFIR	39
C PHIBP(I,J) WOULD HAVE BEEN SET TO 1.0 IN PRIPER.	BPFIR	40
ADVEP = 3.0*(1.0 - PHIBP(I,J))/BPRAD(I,J)	BPFIR	41
C	BPFIR	42
C HEAT FLUX TO BLACK POWDER	BPFIR	43
QCUNBP = TANUBP*TXK/BPRAD(I,J)*(TTX - TBP(I,J))	BPFIR	44
IBE6C=IBE6H	MOD0925	90
IF(.NOT. BGBHN .AND. TBP(IBE6FP,1) .GE. TIGNBP) GO TO 50	MOD0925	91
IF(FLAG1 .EQ. 1. .AND. FLAGJ .EQ. 1.) GO TO 50	MOD0925	92
SPDJ=0.0	BPFIR	48
THKJ2=0.5*(DIAM1-DIAMBT)-TOPGAP	BPFIR	49
IF(J.EQ.2)SPDJ=0.5*(THKJ2+DIAMBT)	BPFIR	50
DISPR=SPDJ+STDOFF	BPFIR	51
IF(TIME .GE. 0.0004)QPMK=3260.*EXP(-30.5*DISPR-600.*TIME)	MOD0925	44
3 + QSPKRS	MOD0925	95
IF(TIME .LT. 0.0004) QPMK=1.28E5*SQRT(TIME)*EXP(-30.5*DISPR)	MOD0925	96
3 + QSPKRS	MOD0925	97
QPMK=SQRT(PRFICOF)*QPMK	MOD0925	98
IF(QPMK.GT.QCUNBP) QCUNBP=QPMK	BPFIR	54
IF(J .EQ. 1) FLAGI=1.	MOD0925	99
FLAGJ=1.	MOD0925	100
IF(IBE6UG(35) .AND. PR11) WRITE(6,57) QPMK,DISPR,I,J	MOD0925	101

```

57 FORMAT(1Z,4X,7HOPFR = 114.5,3X,6HDISFR = 114.5,3X,4H1 = 12.3X,
$ 4HJ = 12)
50 CONTINUE
IF (CONBF .LT. 0.001) GO TO 100
CBAG(1,0) = CONBF*ADVRP
TEMP = 2.0*QBCONS*CONBF
TEFFBP = ((TBP(1,0) - 100)/TEMP)**2
IF (TEFFBP .GT. TIME) TEFFBP = TIME
TBP(1,0) = TBP(1,0) +
+ TEMP*(SQRT(TEFFBP) + DELT) - SQRT(TEFFBP)
IF (PDRAG .LE. 0.6*DIAPRT) GO TO 90
TBP(1BEGP,2) = TIGRBP
TBP(1BEGP,2) = TIGRBP
90 CONTINUE
IF (TBP(1,0) .LT. 11000) GO TO 100
IF (100000(9)) WRITE(6,2000) TIME,1,0
100 CONTINUE
IF (PENET) GO TO 150
IF (TBP(1BEGP,0) .LT. 760.) GO TO 150
IF (TBP(1BEGP+1,0) .LT. 760.) GO TO 150
BGBRN = .TRUE.
150 CONTINUE
C
RETURN
C
2000 FORMAT(1Z,10X,TIME =',14.8,' BLACK POWDER AT GRID',14,14,
$ ' IS IGNITED')
END

```

MOD0925	102
MOD0925	103
BPFIR	55
BPFIR	56
BPFIR	57
MOD0925	104
BPFIR	59
BPFIR	60
BPFIR	61
BPFIR	62
MOD1215	4
MOD1215	5
MOD1215	6
MOD1215	7
BPFIR	63
BPFIR	64
BPFIR	65
MOD0925	105
MOD1115	6
MOD1115	7
MOD1115	8
MOD0925	116
BPFIR	66
BPFIR	67
BPFIR	68
BPFIR	69
BPFIR	70
BPFIR	71

SUBROUTINE PRIMER	PRIMER	2
COMMON/CALLP/BPLEFT	PRIMER	3
COMMON/IGNTE/PORAD,BPCHG,BKNOUT,BGKRN,TMPRG,	MOD1002	6
3 BEGIC(5),XLIC(5)	MOD1002	7
COMMON/PRRFLO/LCTMPF,UPRM,UPRM,PRFAP,POSHR,CSPKS,PRMCOF	MOD0925	118
COMMON/CLOCK/TIME,DELT	PRIMER	5
COMMON/BPT/TEP(2),DEP(2),CLP(2),FLOON,PUALST	MOD0925	119
COMMON/FAILEL/THICK1,PHOUP,PCOMP,BICE,XNTUB,FAIL,MFAIL(60),	MOD0925	120
1 THICK(60),PENET,TIFCE	MOD0925	121
COMMON/CHAM/IX,IX,XB,IB,NGA,NGK,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAMIX	2
3 AREACH,AREAL(60),IGNIT,ORLO,BTAM1,LIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
3 ARELAK(60),AREAA,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMIX	4
3 AREAH2,DIAB1,BELBEG,BELBEG,IPS1,IPS2,KADFS,BPIGN	CHAMIX	5
COMMON/EQNS/LTUX,TZUR,TZUX,TWOTDR,DICH,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
3 UX,UR,IX,GO,TWODI,HEP	EQNS	3
COMMON/INPUTS/C1,C2,C3,C4,TU,TIGN,ECOS,RHOF,PHIO,TF,CA,RHOO,	INPUTS	2
3 HU,PU,UU,OTRHOP,HK,DM,TIGNBP,QBCONS,TOTM	INPUTS	3
COMMON/PRIMV/BPUENS,BPRAD(60,5),AGENP,BGENP,EXPBP	PRIMV	2
COMMON/BAG/PHIBG(60,5),RHORG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VB(60,5),UPB(60,5),PCH(60,5),L2C(60,5),	BAG	3
2 UOTMIG(60,5),GBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHBDT(60,5),RHBDT(60,5),HBGT(60,5),UBGT(60,5),	BAG	5
4 VBGT(60,5),TBG(60,5),UOTMBG(60,5),LCTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPD(60,5),TZK(60,5),TBP(60,5)	BAG	7
LOGICAL IGNIT,ORLO,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
LOGICAL BPLEFT	PRIMER	12
LOGICAL PENET	MOD0605	24
LOGICAL BKNOUT,BGKRN	MOD0925	122
LOGICAL PRI1	MOD0925	123
DATA FORTPI/4,186790/	PRIMER	13
C	PRIMER	14
C	PRIMER	15
C	PRIMER	16
C CALCULATE PRIMER VELOCITY AND MASS FLOW RATE	PRIMER	17
C	PRIMER	18
IMB2=1DEGB	MOD0925	124
TEMP=.00001	MOD0925	125
C	PRIMER	23
C	PRIMER	24
C	PRIMER	25
C BPLEFT WILL BECOME TRUE IF THERE IS SOME BLACK POWDER LEFT AND	PRIMER	26
C THEN PRIMER WILL BE CALLED AGAIN.	PRIMER	27
C BPLEFT = .FALSE.	PRIMER	28
C	PRIMER	29
DO 60 J = 1,NGK	PRIMER	30
DO 60 I = 1,NGA	PRIMER	31
IF(PENET .AND. J .EQ. 1 .AND. I .EQ. 1BE2) GO TO 42	MOD1215	8
IF (PHIBP(I,J) .GE. 0.999) GO TO 35	PRIMER	32
IF(BPRAD(I,J) .LT. .0001) GO TO 35	MOD0925	126
BPLEFT = .TRUE.	MOD0925	127
IF(TBP(I,J) .LT. TIGNBP) GO TO 35	PRIMER	33
IF(PCH(I,J) .LT. 0.0) WRITE(6,7000) PCH(I,J),I,J	PRIMER	36
7000 FORMAT(1H ,*PCH = *,F10.0,2I2)	PRIMER	37
K = AGENP*PCH(I,J)**EXPBP + BGLADP	PRIMER	38
BURNL = K*DELT	PRIMER	39
VOLD = FORTPI*BPRAD(I,J)**3	PRIMER	40
BPRAD(I,J) = BPRAD(I,J) - BURNL	PRIMER	41

	IF (BPRAD(1,J).GE. 0.001) GO TO 40	PRIMER	42
	WRITE(6,2000) 1,J,BPRAD(1,J)	PRIMER	43
30	BPRAD(1,J) = 0.0	PRIMER	44
	PHIBP(1,J) = 1.0	PRIMER	45
	GO TO 55	PRIMER	46
40	CONTINUE	PRIMER	47
	VNEW = FORTFI*BPRAD(1,J)**3	PRIMER	48
	DELTAV = VOLD - VNEW	PRIMER	49
	PRDV = (1.0 - PHIBP(1,J))/VOLD	PRIMER	50
	TEMP = PRDV*DELTAV	PRIMER	51
	DOTMP(1,J) = TEMP*DELTAV	PRIMER	52
	C NOW COMPUTE EFFECT ON PROBABILITY OF FLAME SPREAD	MOD0925	128
	C AND COMPARE WITH EFFECT OF BURPING RATE	MOD0925	129
	IF (1.EQ. 1BB2 .AND. J.EQ. 1) GO TO 42	MOD0925	130
	GO TO 48	MOD0925	131
42	CONTINUE	MOD0925	132
	IF (BRNDUT) GO TO 48	MOD0925	133
	PRESC=PCH(1BB2,2)/144.	MOD0925	134
	IF (PRESC .LT. 0.) PRESC=0.	MOD0925	135
	AREOT=.00004*(PRESC**2.2)	MOD1002	8
	DELTIN=.159*(AREOT/TEMP(1))*DELT	MOD0925	137
	PERAD=DELTIN + PERAD	MOD0925	138
	IF (PERAD .LT. .5*TEMP(1)) GO TO 45	MOD0925	139
	WRITE(6,430) TIME	MOD0925	140
430	FORMAT(2,3X,7H) TIME = 214.7,3X,23H) UNIFORM PAD BURNED AWAY)	MOD0925	141
	PHIBP(1BB2,2) = 1.0	MOD0925	142
	PHIPTD(1BB2,2)=1.0	MOD0925	143
	BRNDUT = .TRUE.	MOD0925	144
	GO TO 48	MOD0925	145
C	THE ROUTE TO STATE 48 MUST BE TAKEN TO	MOD0925	146
C	CONTINUE COMPUTATION FOR GRID(2,1), LGRID	MOD0925	147
C	FROM STATE 42 ON TO HERE CONCERNS J=2 GRIDS ONLY	MOD0925	148
45	CONTINUE	MOD0925	149
46	PHIP=PHIBP(1BB2,2)	MOD0925	150
47	IF (PHIP .GT. .9999) PHIP=.9999	MOD0925	151
	DLVOL=.283*PERAD*DELTIN	MOD0925	152
	IF (DLVOL .LE. 0.) DLVOL=.001	MOD0925	153
	DENS=(TEMP(1)/TEMP(1))/10.785*(DEP(1)+LEP(1)-(CIAMBT**2.))	MOD0925	154
	DTMP=TEMP(1)*DENS*DLVOL	MOD0925	155
	DOTMP(1BB2,2) = DOTMP(1BB2,2) + DTMP	MOD0925	156
	PHIBP(1BB2,2)=PHIBP(1BB2,2)+DLVOL	MOD0925	157
	PHIPTD(1BB2,2) = PHIBP(1BB2,2)	MOD0925	158
48	CONTINUE	MOD0925	159
	PHIPTD(1,J) = PHIBP(1,J) + TEMP	PRIMER	54
	GO TO 56	MOD0925	160
C		PRIMER	56
50	PHIPTD(1,J) = PHIBP(1,J)	PRIMER	57
50	IF (PHIBP(1,J) .GT. .9999) PHIBP(1,J)=1.0	MOD0925	161
	IF (PHIPTD(1,J) .GT. .9999) PHIPTD(1,J)=1.0	MOD0925	162
60	CONTINUE	PRIMER	58
C		PRIMER	59
	RETURN	PRIMER	60
C		PRIMER	61
2000	FORMAT(/// 'RADIUS OF BLACK POWDER AT GRID', I3,I3,' IS',F10.4)	PRIMER	62
	END	PRIMER	63

	SUBROUTINE PRPFIR	PRPFIR	2
C		PRPFIR	3
C	SUBROUTINE PRPFIR CALCULATES PROPELLANT HEAT TRANSFER AND	PRPFIR	4
C	TEMPERATURE RISE LEADING TO IGNITION.	PRPFIR	5
C	TZC(I,J) = SURFACE TEMPERATURE OF PROPELLANT NOT UNDER THE	PRPFIR	6
C	INFLUENCE OF BELL TUBE HOLES AT GRID (I,J)	PRPFIR	7
C	TZR(I) = SURFACE TEMPERATURE OF PROPELLANT UNDER THE INFLUENCE	PRPFIR	8
C	OF BELL TUBE HOLES AT GRID (I,2)	PRPFIR	9
C		PRPFIR	10
C		PRPFIR	11
	COMMON/CHAM/IX,IR,XB,KB,NGX,NGR,IEGB,IEGB,IPATH(60,5),AREAG(5),	CHAMIX	2
	AREACH,ARELAC(60),IGNIT,OE,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
	AREAK(60),AREAAK,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAMIX	4
	AREAH2,DIAM1,BELENB,BELBEG,IPS1,IPS2,RADPS,BPIGN	CHAMIX	5
	COMMON/EQNS/DTOX,TZDR,TZUX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
	UX,UR,NX,GJ,TWOUI,HBP	EQNS	3
	COMMON/CLOCK/TIME,DELT	PRPFIR	14
	COMMON/GRAIN/ XL(60,5), DO(60,5), DI(60,5), FN,	GRAIN	2
	1 ALDI(60,5), DDTDI(60,5), DITDI(60,5), XLO, DOO, DIO,	GRAIN	3
	2 XLB(100), DOB(100), DIB(100), UXLB(100), LDOB(100), UDIB(100)	GRAIN	4
	COMMON/HOLEA/KALHOL(85),NRUWH,NHOLES(85),XCL(85),AREAH(60),	HOLEA	2
	3 AH(60),FRACT(60)	HOLEA	3
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCNS,RHOF,PHIU,TF,CA,RHOO,	INPUTS	2
	3 HU,PU,QU,GTRHOP,HX,DM,TIGNBP,QBCONS,TOTM	INPUTS	3
	COMMON/P/PRINT,MODCH,MODGR,PR11,IDEBUG(35)	P	2
	COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
	1 VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),	BAG	3
	2 DOTMB(60), UBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),	BAG	4
	3 PHIBD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
	4 VBGTD(60,5), TBG(60,5), DOTMBG(60), COTMP(60,5), PHIBP(60,5),	BAG	6
	5 PHIBTD(60,5), TZR(60), TBP(60,5)	BAG	7
	COMMON/PROMO/ FIRE	PRPFIR	20
	DIMENSION UHOLE(60)	PRPFIR	21
	LOGICAL ROW2	PRPFIR	22
	LOGICAL IGNIT,UNEL,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL PRI1,IDEBUG	PLOG	2
	LOGICAL FIRE	PRPFIR	25
C	FIRE REMAINS TRUE AFTER THE FIRST PROPELLANT GRID IGNITES	PRPFIR	26
C		PRPFIR	27
C	ARRAY UBAG IS CLEARED IN UPDATE AT EACH TIME INTERVAL.	PRPFIR	28
C	UBAG IS USED IN CALCULATING UPDATED ENTHALPY IN THE PATH ROUTINES.	PRPFIR	29
C	A TERM MAY HAVE ALREADY BEEN ADDED TO UBAG IN SUBROUTINE UPFIR.	PRPFIR	30
C		PRPFIR	31
C	IGNIT WILL REMAIN TRUE ONLY IF AT EACH GRID THERE IS NO	PRPFIR	32
C	PROPELLANT OR THE PROPELLANT IS IGNITED.	PRPFIR	33
	IGNIT = .TRUE.	PRPFIR	34
	DO 100 J=1,NGR	PRPFIR	35
	ROW2 = J.EQ. 2	PRPFIR	36
	DO 95 I=1,NGX	PRPFIR	37
	IF(PHIBG(I,J).GT. 0.999) GO TO 95	MOD1017	4
C		PRPFIR	39
C	CALCULATE RISE IN TZC.	PRPFIR	40
	IF(TZC(I,J).GE. TIGN) GO TO 95	PRPFIR	41
	IGNIT = .FALSE.	PRPFIR	42
	TX = TBG(I,J)	PRPFIR	43
	TEMP = TX*SUHT(TX)	PRPFIR	44
C		PRPFIR	45

C	GAS VISCOSITY	PRPFIR	46
	TXMU = C1*TEMP/(ITX + C2)	PRPFIR	47
C	THERMAL CONDUCTIVITY OF GAS	PRPFIR	48
	TXK = C3*TEMP/(ITX + C4)	PRPFIR	49
C	REYNOLDS NUMBER	PRPFIR	50
	TEMP1 = RHOBG(I,J)*DM/TXMU	PRPFIR	51
	REIX = TEMP1*ABS(UBG(I,J))	PRPFIR	52
	IF(REIX .LT. .00001) REIX=.00001	PRPFIR	53
		MOD1017	5
C	NUSSELT NUMBER	PRPFIR	55
	TEMP2 = 0.5*SQRT(PCH(I,J)/144.)	PRPFIR	56
	TANUS = 0.3*REIX**0.62 + TEMP2	PRPFIR	57
		PRPFIR	58
C	SURFACE AREA OF PROPELLANT IN GRID PER VOLUME INCREMENT	PRPFIR	59
	ADV = 4.0*(1.0 - PHIBG(I,J))*(DO(I,J) + FN*DI(I,J))/	PRPFIR	60
	(DO(I,J)+DU(I,J) - FN*DI(I,J)*DI(I,J)) + 0.5/XL(I,J)	PRPFIR	61
		PRPFIR	62
C	HEAT FLUX TO PROPELLANT--BTU/FT**2 - SEC	PRPFIR	63
	TEMP3 = TXK/DM	PRPFIR	64
	QCONVA = TANUS*TEMP3*(ITX - TZC(I,J))	PRPFIR	65
	IF(QCONVA .LT. 0.001) GO TO 50	PRPFIR	66
	TEMP = QCONVA*ADV	PRPFIR	67
	IF(ROW2) TEMP = TEMP*FRACT(1)	PRPFIR	68
	QBAG(I,J) = QBAG(I,J) + TEMP	PRPFIR	69
	TEMP = QCONS*QCONVA	PRPFIR	70
	TEFF = ((TZC(I,J) - T0)/TEFF)*2	PRPFIR	71
	IF(TEFF .GT. TIME) TEFF = TIME	PRPFIR	72
		PRPFIR	73
C	HEAT TRANSFER CALCULATION USING SEPI-INFINITE HEAT CONDUCTION	PRPFIR	74
C	EQUATION WITH AN EFFECTIVE TIME	PRPFIR	75
	TZC(I,J) = TZC(I,J) + TEMP*	PRPFIR	76
	(SQRT(TEFF + DELT) - SQRT(TEFF))	PRPFIR	77
	IF(TBP(I,J) .GE. TIGNBP) TZC(I,J)=TIGN	PRPFIR	78
	IF(TZC(I,J) .LT. TIGN) GO TO 50	MOD1205	3
	FRACT(1) = 1.0	PRPFIR	79
	FIKE = .TRUE.	PRPFIR	80
	IF(IDEBUG(16)) WRITE(6,2000) TIME,I,J	PRPFIR	81
		PRPFIR	82
C		PRPFIR	83
C		PRPFIR	84
C	CALCULATE RISE IN TZR	PRPFIR	85
50	CONTINUE	PRPFIR	86
	IF(.NOT. ROW2) GO TO 95	PRPFIR	87
	IF(FRACT(1) .GE. 0.9999) GO TO 95	PRPFIR	88
	IF(TZR(1) .GE. TIGN) GO TO 95	PRPFIR	89
	UNHLE(1) = DOTMIG(1)/(0.7*RHCGB(I,1)*AH(1))	PRPFIR	90
		PRPFIR	91
C	RESULTANT VELOCITY	PRPFIR	92
	UGAS = UBG(I,J)	PRPFIR	93
	IF(I .EQ. 1) UGAS = UBG(I+1,J)/2.	PRPFIR	94
	IF(I .EQ. NGX .AND. NX .EQ. 1) UGAS = UEG(I-1,J)/2. + UBG(I,J)	PRPFIR	95
	/2.	PRPFIR	96
	SVRTUH = SVRT(UGAS*UGAS + VBG(I,J)*VBG(I,J))	PRPFIR	97
	REIXH = TEMP1*SVRTUH	PRPFIR	98
	TXNUSH = 0.3*REIXH**0.62 + TEMP2	PRPFIR	99
		PRPFIR	100

	QCONVH = TANOSH*TEMP3*(TTX - TZR(I))	PRPFIR	101
	IF(QCONVH .LT. 0.001) GO TO 95	PRPFIR	102
	QBAG(I,J) = QBAG(I,J) + QCONVH*ADV*(1.0 - FRACT(I))	PRPFIR	103
	TEMP = QCONS*QCONVH	PRPFIR	104
	TEFF = ((TZR(I) - T0)/TEMP)**2	PRPFIR	105
	IF(TEFF .GT. 11ME) TEFF = 11P2	PRPFIR	106
	TZR(I) = TZR(I) + TEMP*(SQRT(TEFF + DELT) - SQRT(TEFF))	PRPFIR	107
	IF(TZR(I) .LT. TIGN) GO TO 95	PRPFIR	108
	IF(IDEBUG(11)) WRITE(6,2001) TIME,I,J	PRPFIR	109
95	CONTINUE	PRPFIR	110
	IF(TZC(1,J) .LT. TZC(2,J)) TZC(1,J) = TZC(2,J)	PRPFIR	111
100	CONTINUE	PRPFIR	112
	IF(TZC(1,2).GE.11GN)FRACT(1)=1.0	PRPFIR	113
C		PRPFIR	114
	RETURN	PRPFIR	115
C		PRPFIR	116
	2000 FORMAT(1/,' TIME =',E14.8,' PROPELLANT AT GRID',I4,I4,' IS IGNITED',	PRPFIR	117
	\$)	PRPFIR	118
	2001 FORMAT(1/,' TIME =',E14.8,' PROPELLANT UNDER INFLUENCE OF BELL TUBE	PRPFIR	119
	\$ HOLES IN GRID',I4,I4,' IS IGNITED')	PRPFIR	120
	END	PRPFIR	121

SUBROUTINE REGRES	REGRES	2
COMMON/BURN/ATPB,C1,PEXP	REGRES	3
COMMON/CHAM/IX,IR,XB,NB,NGX,NGR,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAMIX	2
AREACH,AREAL(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMIX	4
AREAM2,DIAMB1,BELBEG,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAMIX	5
COMMON/EQNS/LTUX,T2UR,T2UX,TWOTUR,BTGR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
UX,UR,NX,GJ,IWOT,HBP	EQNS	3
COMMON/GRAIN/XL(60,5), UD(60,5), DI(60,5), FN,	GRAIN	2
XLDT(60,5), UDDT(60,5), DTD(60,5), XLO, DDO, DIO,	GRAIN	3
XLH(100), DOB(100), DIR(100), UXLB(100), LDOB(100), UDIB(100)	GRAIN	4
COMMON/HOLEA/RADHOL(85),RHOH,WHOLEA(65),XCL(85),AREAH(60),	HOLEA	2
AH(60),FRACT(60)	HOLEA	3
COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,WLONS,RHOF,PHIU,TF,CA,RHOO,	INPUTS	2
H0,P0,U0,GTRHOP,HW,DM,TIGNP,QBCONS,TOTM	INPUTS	3
COMMON/SPLINT/WHOLEC,WHOLEH	REGRES	9
COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
VBG(60,5), UPB(60,5), PCH(60,5), I2C(60,5),	BAG	3
DUIMIG(60), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
VBGTD(60,5), TBG(60,5), DOTMBG(60), DOTMP(60,5), PHIBP(60,5),	BAG	6
PHIPTD(60,5), IZR(60), ITP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
LOGICAL WHOLEC,WHOLEH	REGRES	12
LOGICAL ROW2,PART	REGRES	13
DATA PIDE/.785398/	REGRES	14
C	REGRES	15
C SUBROUTINE REGRES CALCULATES UPDATED GRAIN DIMENSIONS AND POROSITY	REGRES	16
C DUE TO BURNING OF M6 PROPELLANT. PHIBD IS USED IN THE PATH	REGRES	17
C ROUTINES AND UPDATED GRAIN DIMENSIONS ARE USED IN PROPEL.	REGRES	18
C	REGRES	19
C	REGRES	20
C ARRAY UOTMB IS CLEARED IN UPDATE AT EACH TIME INTERVAL	REGRES	21
DO 50 J=1,NGR	REGRES	22
ROW2 = J .EQ. 2	REGRES	23
DO 48 I=1,NGX	REGRES	24
C	REGRES	25
IF (PHIBG(I,J).GE.0.99999) GO TO 45	REGRES	26
PART = ROW2 .AND. FRACT(I) .LE. 0.9999	REGRES	27
C	REGRES	28
C IF PROPELLANT IN THE GRID IS NOT IGNITED, DO NOT DO THE BURN	REGRES	29
C CALCULATIONS	REGRES	30
IF (TZC(I,J).GE.TIGN) GO TO 10	REGRES	31
IF (PART .AND. IZR(I) .LT. TIGN) GO TO 45	REGRES	32
IF (.NOT. PART .AND. TZC(I,J) .LT. TIGN) GO TO 45	REGRES	33
10 CONTINUE	REGRES	34
R = ATPB*PCH(I,J)**PEXP + CT	REGRES	35
BURNL = R*HOUT	REGRES	36
C	REGRES	37
C UPDATE GRAIN LENGTH,	REGRES	38
XLDT(I,J) = XL(I,J) - BURNL	REGRES	39
C	REGRES	40
C SEE IF GRAIN HAS SPLIT INTO SPLINTERS	REGRES	41
C*****NOTE THAT OLD DIMENSIONS ARE BEING TESTED,	REGRES	42
IF (DO(I,J) .LT. 3.0*DI(I,J)) GO TO 50	REGRES	43
C	REGRES	44
C	REGRES	45

C	UPDATE OTHER DIMENSIONS	REGRES	46
	DUOTDT(I,J) = DU(I,J) - BURNL	REGRES	47
	DITDT(I,J) = DI(I,J) + BURNL	REGRES	48
C		REGRES	49
C	CALCULATE OLD AND NEW VOLUMES OF A GRAIN.	REGRES	50
	VOLD = PLOF*XL(I,J)*(DO(I,J)*DU(I,J) - FN*DI(I,J)*DI(I,J))	REGRES	51
	VNEW = PLOF*XLTD(I,J)*(DOTD(I,J)*DOTCT(I,J) -	REGRES	52
	FN*DI(I,J)*DITDT(I,J))	REGRES	53
	GO TO 30	REGRES	54
C		REGRES	55
C	CALCULATE OLD AND NEW GRAIN VOLUMES,	REGRES	56
C	AFTER THE GRAIN HAS SPLINTERED, VALUES FOR THE CROSS-SECTIONAL	REGRES	57
C	AREA OF THE PARTICLES GO INTO ARRAY LC AND VALUES FOR PERIMETER	REGRES	58
C	GO INTO ARRAY D1. IF THE GRAIN HAS JUST SPLINTERED, AREA AND	REGRES	59
C	PERIMETER HAVE TO BE INITIALIZED. IF THE GRAIN HAS JUST	REGRES	60
C	SPLINTERED DO(I,J) IS APPROXIMATELY 3.*DI(I,J) AND IF NOT	REGRES	61
C	DO(I,J) WILL BE LESS THAN DI(I,J).	REGRES	62
	CONTINUE	REGRES	63
20	IF(DO(I,J) .LE. DI(I,J)) GO TO 25	REGRES	64
	WHOLEC = .FALSE.	REGRES	65
	AREA = PLOF*(DO(I,J)*DU(I,J) - FN*DI(I,J)*DI(I,J))	REGRES	66
	D1(I,J) = 3.14*(DO(I,J) + FN*DI(I,J))	REGRES	67
	DU(I,J) = AREA	REGRES	68
C		REGRES	69
25	DELR = BURNL*0.5	REGRES	70
	DUOTDT(I,J) = DU(I,J) - DI(I,J)*DELR	REGRES	71
C		REGRES	72
	IF(DUOTDT(I,J) .GE. 1.0E-7) GO TO 27	REGRES	73
	DUOTDT(I,J) = 0.0	REGRES	74
	DITDT(I,J) = 0.0	REGRES	75
	XLTD(I,J) = 0.0	REGRES	76
	PHIBTD(I,J) = 1.0	REGRES	77
	GO TO 48	MOD0925	163
C		REGRES	80
27	CONTINUE	REGRES	81
C	ASSUME THE RATIO OF PERIMETER SQUARED TO CROSS-SECTIONAL AREA IS	REGRES	82
C	CONSTANT	REGRES	83
	DITDT(I,J) = SQR(DI(I,J)*DI(I,J)/DO(I,J)*DUOTDT(I,J))	REGRES	84
C		REGRES	85
C	VOLUME IS LENGTH TIMES CROSS-SECTIONAL AREA	REGRES	86
	VOLD = XL(I,J)*DU(I,J)	REGRES	87
	VNEW = XLTD(I,J)*DUOTDT(I,J)	REGRES	88
C		REGRES	89
C		REGRES	90
30	IF(VNEW .LE. 0.0) GO TO 45	REGRES	91
C		REGRES	92
	DELTAV = VOLD - VNEW	REGRES	93
C		REGRES	94
C	CALCULATE NUMBER OF GRAINS PER GRID/VOLUME OF GRID	REGRES	95
	PNDV = (1.0 - PHIBG(I,J))/VOLD	REGRES	96
	IF(PART) PNDV = PNDV*(1.0 - FRAC(I))	REGRES	97
	TEMP = PNDV*DELTAV	REGRES	98
C		REGRES	99
C	CALCULATE GAS MASS GENERATED BY BURNING PROPELLANT/GRID VOLUME	REGRES	100
	DUIMB(I,J) = TEMP*RHUP	REGRES	101
C		REGRES	102

C	UPDATE POROSITY	REGRES	103
	PHI1D(I,J) = PHIBG(I,J) + TEPP	REGRES	104
	IF(PART) DOTDT(I,J) = DOTDT(I,J)*.1.0 - FRACT(I) + DO(I,J)*	REGRES	105
3	FRACT(I)	REGRES	106
	IF(PART) DITDT(I,J) = DITDT(I,J)*.1.0 - FRACT(I) + DI(I,J)*	REGRES	107
3	FRACT(I)	REGRES	108
	IF(PART) XLTDT(I,J) = XLTDT(I,J)*.1.0 - FRACT(I) + XL(I,J)*	REGRES	109
3	FRACT(I)	REGRES	110
	GO TO 48	REGRES	111
C		REGRES	112
45	PHI1D(I,J) = PHIBG(I,J)	REGRES	113
C		REGRES	114
48	CONTINUE	REGRES	115
50	CONTINUE	REGRES	116
C		REGRES	117
	RETURN	REGRES	118
	END	REGRES	119

	SUBROUTINE PRPVEL		
	COMMON/CHAM/IX,IR,XB,HB,NGX,NGR,IBLUE,IBNDB,IPATH(60,5),AREAG(5),	MOD0925	164
	1 AREACH,AREAC(60),IGNIT,ONED,DIAP1,DIAM2,DIS1,DIS2,DIS3,DIS4,	MOD0925	165
	2 AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,TUPGAP,AREAGP(60),DAVG,	MOD0925	166
	3 AREAH2,DIAMUT,BELLEND,BELLEH,IPSI,IPSC,RADPS,BPIGN	MOD0925	167
	COMMON/CLUCK/TIME,DELT	MOD0925	168
	COMMON/EQNS/UTDX,T2DR,T2DX,TWOTDR,DICK,HMB,TWOGJ,OVAXIS,OVAXIT,	MOD0925	169
	4 UX,UR,UX,GO,TWOT,MBP	MOD0925	170
	COMMON/FORCE/PFORCE(60,5),PFORDT(60,5)	MOD0925	171
	COMMON/P/IPRINT,MOUCH,MODGR,PR11,IDEBUG(35)	MOD0925	172
	COMMON/IGTE/PLRAD,BPCH,GRMOUT,BGBRN,TMPBG,	MOD0925	173
	5 BEGTC(5),XLTC(5)	MOD0925	174
	COMMON/FAILED/THICKT,PHOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),	MOD0925	175
	1 THICK(60),PENET,TYFCE	MOD0925	176
	COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENER,BGENER,EXPBP	MOD0925	177
	COMMON/PRU/LF1,LF2,MF1,MF2,BGJAM,BGSTF,NVSP,IBGNP	MOD0925	178
	COMMON/PRMFLO/DOTMPF,UPKM,UPKM,PRXP,POSHR,CSPRKS,PRMCOF	MOD0925	179
	COMMON/BP1/TEP(2),LEP(2),CEP(2),FLOUK,PDALST	MOD0925	180
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,WLCKS,RHOF,PHIU,TF,CA,RHOO,	MOD0925	181
	6 MO,PC,UU,BIRHOP,HK,UM,FIGNBP,OBCONS,TOTM	MOD0925	182
	COMMON/TUBE/IBENDTC(5),ITUBE,IBEGTC(5),INBP	MOD0925	183
	COMMON/BAG/PHIBG(60,5),RHORG(60,5),FBG(60,5),UBG(60,5),	MOD1115	9
	1 VBG(60,5),UPH(60,5),PCH(60,5),IZL(60,5),	MOD0925	184
	2 UOIMIG(60),UBAG(60,5),XDRAG(60,5),DOTMB(60,5),UPBDT(60,5),	MOD0925	185
	3 PHIBTD(60,5),RHUETD(60,5),HBGTD(60,5),LBGTD(60,5),	MOD0925	186
	4 VBGTD(60,5),TBG(60,5),DOTMBG(60,5),CUTMP(60,5),PHIBP(60,5),	MOD0925	187
	5 PHIPTD(60,5),TZK(60),TBP(60,5)	MOD0925	188
	DIMENSION PPKOP(60,5)	MOD0925	189
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,PIGN	MOD0925	190
	LOGICAL PR11,IDEBUG	MOD0925	191
	LOGICAL BGJAM,BGSTF,PENET	MOD0925	192
	DATA GRAV/32.16/	MOD0925	193
	DATA INUV/O/	MOD0925	194
	DATA MBKN/O/	MOD1115	10
	CALL CLEAR(PFORCE(1,1),PFORDT(60,5))	MOD1115	11
	CALL CLEAR(PPKOP(1,1),PPKOP(60,5))	MOD0925	195
C*		MOD0925	196
C	SUBROUTINE PRPVEL CALCULATES UPDATED PROPELLANT VELOCITY	MOD0925	197
C	(ASSUMING PROPELLANT IS FREE TO MOVE)	MOD0925	198
C*		MOD0925	199
	DO 31 I=1,NGX	MOD0925	200
	DO 31 J=1,NGR	MOD0925	201
C*		MOD0925	202
C	PHIBTD VALUES SHOULD BE PUT INTO ARRAY PHIRG AND USED HERE	MOD0925	203
C*		MOD0925	204
	PHIAVE=(4.0-PHIBP(I,J)-PHIBP(I+1,J)-PHIBG(I,J)-PHIBG(I+1,J))/2.0	MOD0925	205
	IF(I.NE.1) PHIAVE=PHIAVE/2.0+(4.0-PHIBP(I,J)-PHIBP(I-1,J)	MOD0925	206
	+PHIBG(I,J)-PHIBG(I-1,J))/4.0	MOD0925	207
	IF(PHIAVE.NE.0.0) XDRAG(I,J)=XDRAG(I,J)+	MOD0925	208
	* (2.0-PHIBG(I,J)-PHIBP(I,J))/PHIAVE	MOD0925	209
	PHIBG(I,J)=PHIBTD(I,J)	MOD0925	210
	PHIBP(I,J)=PHIPTD(I,J)	MOD0925	211
	PHIBG(I,J)=PHIBG(I,J)+PHIBP(I,J)-1.0	MOD0925	212
C		MOD0925	213
	IF(PHIBG(I,J).GE.0.999) GO TO 31	MOD0925	214
C*		MOD0925	215
C	CALCULATE TOTAL PRESSURE GRADIENT	MOD0925	216
		MOD0925	217

C*	IPA = IPATH(1,J)	MOD0925	218
	GO TO (10,10,10,10,15,15,15,15,20,20,20,20),IPA	MOD0925	219
10	DPDX = (PCH(I-1,J) - PCH(I+1,J)) * 0.5/DX	MOD0925	220
	GO TO 25	MOD0925	221
15	DPDX = (PCH(I-1,J) - PCH(I,J))/DX	MOD0925	222
	GO TO 25	MOD0925	223
20	DPDX = (PCH(I,J) - PCH(I+1,J))/DX	MOD0925	224
25	CONTINUE	MOD0925	225
30	PPROM(I,J) = XDRAG(I,J) + (DPDX - XDRAG(I,J)) * (1. - PHIBG(I,J))	MOD0925	226
	PHIBG(I,J) = PHIBG(I,J) + 1.0 - PHIBF(I,J)	MOD0925	227
C		MOD0925	228
C	THE CURRENT VALUES FOR ARRAY PHIBG MUST BE RECOVERED	MOD0925	229
C	BECAUSE CHANGES ARE GOING TO BE MADE IN PHIBF, PHIBPPPP THAT IS	MOD0925	230
C		MOD0925	231
31	CONTINUE	MOD0925	232
100	DO 900 J=1,NGR	MOD0925	233
	I=1	MOD0925	234
	ACCLL=0.	MOD0925	235
	ACCLM = 0.	MOD0925	236
	FLAG = 0.	MOD0925	237
	LF1=0	MOD0925	238
	LF2=0	MOD0925	239
	MF1=0	MOD0925	240
	MF2=0	MOD0925	241
	SUMFCL = 0.	MOD0925	242
	SUMASS = 0.	MOD0925	243
	TYFCE=0.	MOD0925	244
	TOTMAS=0.	MOD0925	245
	ICOUNT = 0	MOD0925	246
	ICONI=0	MOD0925	247
	FORCE=0.	MOD0925	248
101	CONTINUE	MOD0925	249
	STDOFF=BELBEG-TEP(1)	MOD0925	250
	IF(UNED) GO TO 110	MOD0925	253
	IF(.NOT. PENET .AND. J .EQ. 1) GO TO 800	MOD1215	9
	IF(J .GT. 1) GO TO 110	MOD1215	10
	PRMXP=2060.46*(EXP(-.635*STDOFF/FTUBR)) * TIME	MOD1215	11
	IF(TIME .GE. .0004) PRMXP=1.33E6*(EXP(-.635*STDOFF/FTUBR-1200.	MOD1205	5
	1 * TIME))	MOD1205	6
	PRMXP=PRMCOF*PRMXP	MOD0925	256
	BGFCL = PRMXP	MOD0925	257
	FORCE=AREAAX*(BGFCL-PCH(1BGNP,J))	MOD1215	12
	ALGHT=0.	MOD0925	260
	IF(FORCE .LT. 0.) FORCE=0.	MOD0925	261
	IF(BGJAM) GO TO 104	MOD0925	262
	KBRN = KBRN + 1	MOD0925	263
	IF(KBRN .GT. 150) GO TO 103	MOD1115	21
102	CONTINUE	MOD1115	22
	IF(FORCE .GT. TYFCE) GO TO 103	MOD0925	264
	I = I+1	MOD0925	265
	IF (I .EQ. 5) GO TO 800	MOD0925	266
	FORCE=FORCE + PPROM(I,J)*DX*AREAAX	MOD0925	267
	GO TO 102	MOD0925	268
103	WEGHT = (1-PHIBP(1,1))*BPUENS*DX/2.*AREAAX	MOD0925	269
	BGJAM=.TRUE.	MOD0925	270
	WRITE(6,1031) TIME	MOD0925	271
		MOD0925	272

1031	FORMAT(17,5X,CHTIME = L14.7, X+23HIGNITER BAG TIES FAILED)	MOD0925	273
104	CONTINUE	MOD0925	274
	DO 1032 I=1,N6X	MOD0925	275
	IRGNP=1	MOD0925	276
	IF (PHIHP(I,J) .LT. .999) GO TO 1033	MOD0925	277
1032	CONTINUE	MOD0925	278
1033	CONTINUE	MOD0925	279
	I=IRGNP	MOD0925	280
1041	IF (PHIHP(I,1) .GE. .999) GO TO 1044	MOD0925	281
	ALGHT = (1. - PHIHP(I,J))*BPDENS*DX*AREAA*LEGHT	MOD0925	282
	NEND = 1	MOD0925	283
	I = I + 1	MOD0925	284
	GO TO 1041	MOD0925	285
1044	CONTINUE	MOD0925	286
	IF (BGSIP) GO TO 1146	MOD0925	287
	IF (.NOT. BGSIP) GO TO 1045	MOD0925	288
	IMOV=IMOV+1	MOD1115	23
	IIMOV=MOD(IMOV,100)	MOD1115	24
	IF (IIMOV .NE. 0) GO TO 1045	MOD1115	25
	NEND=NEND+1	MOD0925	289
1146	CONTINUE	MOD0925	290
	IRGNP=IRGNP+1	MOD0925	291
1045	CONTINUE	MOD0925	292
	LL1=IRGNP+1	MOD0925	293
	ISTPP=IENDTC(ITUBE)+1	MOD1115	26
	IF (INBP .LT. ISTPP) INBP=ISTPP	MOD1115	27
	IF (INBP .GT. IENDB) INBP=IENDB	MOD1205	7
	IF (BGSIP) NEND=INBP	MOD1111	19
	VOLBG=FLOAT(NEND-IRGNP+1)*AREAA*DX	MOD0925	295
	PHIPP=1.-WEGHT/(BPDENS*VOLBG)	MOD0925	296
	ELTCHG=VOLBG/AREAA	MOD0925	297
	LBP=ELTCHG/DX	MOD0925	298
	FRAT =ELTCHG/DX-FLOAT(LBP)	MOD0925	299
C		MOD1111	20
C	IRGNP IDENTIFIES THE FIRST GRID WITH POWDER AFTER IGNITER	MOD1111	21
C	PAU FAILURE. LL1 IDENTIFIES THE FIRST PACKED GRID	MOD1111	22
	IF (PHIPP .LT. 0.001) PHIPP=0.001	MOD0925	303
	PHIHP(IRGNP,J) = 1.-FRAT*(1.-PHIPP)	MOD1111	23
	IF (BGSIP) GO TO 1047	MOD0925	305
	IF (NEND .LT. INBP) GO TO 1047	MOD0925	306
	BGSIP=.TRUE.	MOD1111	24
	WRITE(6,1046)	MOD0925	308
1046	FORMAT(13X,13HBGSIP IS TRUE)	MOD1111	25
	GO TO 109	MOD1111	26
1047	CONTINUE	MOD0925	309
105	CONTINUE	MOD0925	310
	DO 1055 I=IRGNP,NEND	MOD0925	311
1055	FORCE=FORCE + PPROP(I,J)*DX*AREAA	MOD1111	27
	IF (FORCE .GT. 350.) FORCE=350.	MOD0925	313
	IF (FORCE .LT. 10.) FORCE=10.	MOD0925	314
	SMLTC=XLTC(1)+XLTC(2)+XLTC(3)+XLTC(4)+XLTC(5)	MOD0925	315
	ELT = SMLTC*(1.- .03145*((FORCE-10.)*.545))	MOD1111	28
	VULP=AREAA*ELT	MOD1111	29
	LLBP=ELT/DX	MOD0925	317
	IF (LL1 .LT. ELTCHG) GO TO 109	MOD0925	318
	IRGNP=NEND-LLBP	MOD0925	319
		MOD1111	30

FRATN=LL1/UX-FLOAT(LLBP)	MOD0925	321
LL1=IBGNP+1	MOD0925	322
PHIBP(IBGNP,J)=1.-WEIGHT*FRATN/(VOLP*BPDENS)	MOD0925	323
PHIPP=1.-WEIGHT/(VOLP*BPDENS)	MOD0925	324
IF (PHIPP .LT. 0.001) PHIPP=0.001	MOD1111	31
107 CONTINUE	MOD0925	326
PHIPTD(IBGNP,J)=PHIBP(IBGNP,J)	MOD0925	327
DO 1070 I=LL1,NEND	MOD0925	328
PHIPTD(I,J)=PHIPP	MOD0925	329
PHIBP(I,J)=PHIPP	MOD0925	330
1090 CONTINUE	MOD0925	331
NVSP=IBGNP-1	MOD0925	332
DO 1190 I=1,NVSP	MOD0925	333
PHIBP(I,J)=1.0	MOD0925	334
PHIPTD(I,J)=1.0	MOD0925	335
1190 CONTINUE	MOD0925	336
1200 LF1=LL1	MOD0925	337
LF2=NEND	MOD1111	32
GO TO 480	MOD0925	339
110 CONTINUE	MOD0925	340
IF (LF2 .GT. 0 .AND. PHIBG(I,J) .LE. .2*PHI0) GO TO 534	MOD1215	13
IF (PHIBG(I,J) .LE. .2*PHI0) GO TO 200	MOD1215	14
FACPH=1.-PHIBG(I,J)	MOD0925	343
IF (FACPH .LT. .001) FACPH = 0.	MOD1017	6
IF (I .EQ. 1) GO TO 111	MOD1215	15
PFORCE(I-1,J) = (1.45E3/(1.0-PHI0)*(PHI0/PHIBG(I-1,J)-1.0))/UX	MOD1215	16
PFORCE(I+1,J) = (1.45E3/(1.0-PHI0)*(PHI0/PHIBG(I+1,J)-1.0))/UX	MOD0301	25
IF (PFORCE(I-1,J) .LT. 0.) PFORCE(I-1,J) = 0.	MOD1215	18
IF (PFORCE(I+1,J) .LT. 0.) PFORCE(I+1,J) = 0.	MOD1215	19
FORCH = PPMOP(I,J) + PFORCE(I-1,J) - PFORCE(I+1,J)	MOD1215	20
IF (PHIBG(I,J) .GT. PHI0) FORCH = PPMOP(I,J)	MOD1215	21
111 CONTINUE	MOD1215	22
GMASS=KMOF*FACPH	MOD1002	9
ACCL = 0.	MOD0925	346
IF (GMASS .EQ. 0.) GO TO 120	MOD0925	347
C	MOD0925	348
C* GET PROPELLANT VELOCITY IN EACH GRID	MOD0925	349
C	MOD0925	350
ACCL = FORCH/GMASS	MOD1215	23
120 DELUP = ACCL*DELT*GRAV	MOD0925	352
UPBDT(I,J) = UPB(I,J) + DELUP	MOD0925	353
IF (I .EQ. 1 .AND. UPBDT(I,J) .LT. 0.) UPBDT(I,J) = 0.	MOD0925	354
IF (I .EQ. NGX) GO TO 800	MOD0925	355
150 I=I+1	MOD0925	356
GO TO 110	MOD0925	357
C	MOD0925	358
C* IDENTIFY THE FIRST GRID THAT IS PACKED FULL WITH PROPELLANT	MOD0925	359
C	MOD0925	360
200 LF1 = 1	MOD0925	361
C	MOD0925	362
USUM=0.	MOD1017	7
N=1	MOD1017	8
202 CONTINUE	MOD1017	9
C	MOD1017	10
C FIRST GET ACCELERATION IN GRID I,J	MOD1017	11
FCPH=1.-PHIBG(I,J)	MOD1017	12
IF (FCPH .LT. .001) FCPH=0.	MOD1017	13

	GMASS=RHOP*FCPH	MOD1017	14
	ACCL=0.	MOD1017	15
	IF(GMASS .EQ. 0.) GO TO 211	MOD1017	16
	ACCL=PPROP(I,J)/GMASS	MOD1017	17
211	DELUP=ACCL*DELT*GRAV	MOD1017	18
	UPBUT(I,J)=UPB(I,J)+DELUP	MOD1017	19
C		MOD1017	20
C	NOW TEST FOR SEPARATION OF PACKED GRIDS INTO TWO GROUPS	MOD1017	21
C		MOD1017	22
	IF(I .EQ. LF1) GO TO 212	MOD1017	23
	IF(UPBUT(I,J) .GT. UPP0) GO TO 525	MOD1017	24
212	CONTINUE	MOD1017	25
	SUMFCE = SUMFCE + PPROP(I,J)	MOD1017	26
	SUMASS = SUMASS + RHOP*FCPH	MOD1017	27
	ACCLF = 0.	MOD1017	28
	IF(SUMASS .EQ. 0.) GO TO 231	MOD1017	29
	ACCLF = SUMFCE/SUMASS	MOD1017	30
231	ACCL = ACCLF	MOD1017	31
	DELUP = ACCL*DELT*GRAV	MOD1017	32
	USUM = UPB(I,J) + USUM	MOD1017	33
	UPP0 = USUM/N + DELUP	MOD1017	34
	N = N + 1	MOD1017	35
	IF(LF1 .EQ. 1 .AND. UPP0 .LT. 0.) GO TO 480	MOD1017	36
	GO TO 520	MOD1017	37
480	DO 500 I=LF1,LF2	MOD1017	38
500	UPBUT(I,J) = 0.	MOD1017	39
	IF(LF2 .EQ. NGX) GO TO 800	MOD1017	40
520	I=I+1	MOD0925	417
	IF(PH1BG(I,J) .GT. 0.2*PH10) FLAG=0.	MOD1215	24
	IF(I .GT. NGX) GO TO 530	MOD0925	418
	IF(PH1BG(I,J) .GT. 0.2*PH10) GO TO 530	MOD1215	25
C		MOD0925	420
C*	IDENTIFY THE LAST GRID IN THE FIRST GROUP OF PACKED GRIDS	MOD0925	421
C		MOD0925	422
	LF2=1	MOD0925	423
	GO TO 202	MOD0925	424
525	FLAG=1.	MOD0925	425
530	DO 535 I=LF1,LF2	MOD0925	426
535	UPBUT(I,J) =UPP0	MOD1115	29
	IF(I .GT. NGX) GO TO 800	MOD0925	428
	IF(FLAG) 700,700,534	MOD0925	429
534	MF1=1	MOD0925	430
	M = 1	MOD1017	41
	USUM = 0.	MOD1017	42
535	TOTFCE=TOTFCE + PPROP(I,J)	MOD0925	431
C		MOD0925	432
C*	FIND AVERAGE VELOCITY INCREASE FOR SEPARATED GROUP OF PACKED GRIDS	MOD0925	433
C		MOD0925	434
	FFPH=1.-PH1BG(I,J)	MOD0925	435
	IF(FCPH .LT. .001) FCPH = 0.	MOD1017	43
	TOTMAS=TOTMAS+RHOP*FFPH	MOD1002	11
	ACCLM = 0.	MOD0925	436
	IF(TOTMAS .EQ. 0.) GO TO 541	MOD0925	439
	ACCLM = TOTFCE/TOTMAS	MOD0925	440
541	DELUP=ACCLM*DELT*GRAV	MOD0925	441
	MF2 = 1	MOD1017	44
	USUM=UPB(I,J) + USUM	MOD1017	45

UPPU = USUM/M + DELUP	MOD1017	46
M = M + 1	MOD1017	47
600 CONTINUE	MOD0925	464
GO TO 680	MOD0925	465
620 DO 660 I=MF1,MF2	MOD0925	466
660 UPBUT(I,J)=0.	MOD0925	467
GO TO 800	MOD1017	48
680 I = MF2 + 1	MOD0925	469
IF(I .GT. NGX) GO TO 665	MOD0925	470
IF(PHABG(I,J) .GT. 0.2*PH10) GO TO 665	MOD1215	26
GO TO 535	MOD0925	472
665 DO 667 I=MF1,MF2	MOD0925	473
667 UPBUT(I,J) = UPPU	MOD1115	30
670 CONTINUE	MOD0925	475
IF(MF2 .EQ. NGX) GO TO 800	MOD0925	476
700 FLAG = 0.	MOD0925	477
720 GO TO 110	MOD0925	478
800 CONTINUE	MOD0925	479
IF(PK11) WRITE(6,731)LF1,LF2,J	MOD0925	480
731 FUMMA(1/.8X,4HMF1=15,3X,4HMF2=15,3X,2FJ=13)	MOD0925	481
IF(PK11) WRITE(6,733)MF1,MF2,J	MOD0925	482
733 FUMMA(1/.8X,4HMF1=15,3X,4HMF2=15,3X,2FJ=13)	MOD0925	483
900 CONTINUE	MOD0925	484
RETURN	MOD0925	485
END	MOD0925	486

SUBROUTINE PROPEL		PROPEL	2
C*	THIS SUBROUTINE PRODUCES THE ACTUAL MOVEMENT OF PROPELLANT	PROPEL	3
C		PROPEL	4
	COMMON/CHAM/IX,IX,XB,RB,NGX,NGK,IBLGB,IENDB,IPATH(60,5),AREAG(5),	CHAMIX	2
1	AREACH,AREAL(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
2	AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),UAVG,	CHAMIX	4
3	AREAH2,DIAMBT,BLEND,BLBEG,IPS1,IPS2,RADFS,BPIGN	CHAMIX	5
	COMMON/CLOCK/TIME,DELTA	PROPEL	6
	COMMON/EQNS/DTUX,T2DR,T2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
3	UX,UR,NX,GJ,TWOT,HBP	EQNS	3
	COMMON/GRAIN/ XL(60,5), DO(60,5), DI(60,5), FN,	GRAIN	2
1	XLTDI(60,5), DOTDT(60,5), DITDT(60,5), XLQ, DOQ, DIO,	GRAIN	3
2	ALB(100), DOB(100), DIB(100), UXLB(100), LDOB(100), UDIB(100)	GRAIN	4
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,WCNS,RHOF,PHIO,TF,CA,RHOD,	INPUTS	2
3	MO,P0,U0,GTRHOP,HW,UN,TIGNBP,QBCNS,TOTM	INPUTS	3
	COMMON/SPLINT/WHOLEC,WHOLEB	PROPEL	10
	COMMON/PRU/LF1,LF2,MF1,MF2,BGJAM,BGSTP,NVSP,IBGNP	MOD0925	487
	COMMON/P/IPRINT,MODCH,MOUGH,PR11,IDLBGC(35)	MOD0925	488
	COMMON/BAG/PHIBG(60,5), RHORG(60,5), RBG(60,5), UBG(60,5),	BAG	2
1	VBG(60,5), UPH(60,5), PCH(60,5), LCG(60,5),	BAG	3
2	DOTHBG(60,5), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
3	PHIBTD(60,5), RHOBTD(60,5), HBGT(60,5), UBGTD(60,5),	BAG	5
4	VBGTD(60,5), THG(60,5), DOTMBG(60,5), DOTMP(60,5), PHIBP(60,5),	BAG	6
5	PHIPTD(60,5), TZK(60,5), TBP(60,5)	BAG	7
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL WHOLEC,WHOLEB	PROPEL	13
	LOGICAL BPLEFT	MOD0925	489
	LOGICAL PR11,IDEBUG	MOD0925	490
	LOGICAL BGJAM,BGSTP	MOD0925	491
C		PROPEL	14
C	ARRAY UPBDT IS CLEARED IN MAIN AT EACH TIME INTERVAL	PROPEL	15
C	UPDATED VALUES OF POROSITY (FROM REGRES) WERE PUT INTO ARRAY PHIBG	PROPEL	16
C	IN PRPVEL. UPDATED VALUES OF UPB (FROM PRPVEL) WERE PUT INTO	PROPEL	17
C	ARRAY UPB IN PRPVEL. UPDATED GRAIN DIMENSIONS (FROM REGRES) WERE	PROPEL	18
C	PUT INTO ARRAYS XL, DO, DI IN REGRES.	PROPEL	19
C*		PROPEL	20
	DO 10 J=1,NGK	PROPEL	21
	DO 10 I=1,NGX	PROPEL	22
	PHIBG(I,J) = PHIBTD(I,J)	PROPEL	23
	XL(I,J) = XLTDI(I,J)	PROPEL	24
	DO(I,J) = DOTDT(I,J)	PROPEL	25
	DI(I,J) = DITDT(I,J)	PROPEL	26
10	CONTINUE	PROPEL	27
C		PROPEL	28
	DO 80 J=1,NGK	PROPEL	29
	DO 70 I=1,NGX	PROPEL	30
	INCR = 1	PROPEL	31
C	TEST VELOCITY IN ITH GRID TO DETERMINE INTO WHICH ADJACENT	PROPEL	32
C	GRID THE PROPELLANT WANTS TO FLOW.	PROPEL	33
	IF(UPBDT(I,J).LT.0.0)INCR = -1	PROPEL	34
	IF(UPBDT(I,J).EQ.0.0)INCR = 0	PROPEL	35
	ITWO = 1 + 2 * INCR	PROPEL	36
	IONE = 1 + INCR	PROPEL	37
	UPI = UPBDT(I,J)	PROPEL	38
	IF (I .EQ. 1) GO TO 15	PROPEL	39
	IF (J .EQ. 2) GO TO 12	PROPEL	40
	AREAX = AREAL (I)	PROPEL	41

ARONE = AREAL (IONE)	PROPEL	42
ARTWO = AREAL (ITWO)	PROPEL	43
AREAP1 = AREAL (I+1)	PROPEL	44
AREAP1 = AREAL (I-1)	PROPEL	45
GO TO 18	PROPEL	46
12 AREAX = AREAR (I)	PROPEL	47
ARONE = AREAR (IONE)	PROPEL	48
ARTWO = AREAR (ITWO)	PROPEL	49
AREAP1 = AREAR (I+1)	PROPEL	50
AREAP1 = AREAR (I-1)	PROPEL	51
GO TO 18	PROPEL	52
15 IF (J.EQ. 2) GO TO 16	PROPEL	53
AREAX = AREAL (I)	PROPEL	54
ARONE = AREAL (IONE)	PROPEL	55
ARTWO = AREAL (ITWO)	PROPEL	56
AREAP1 = AREAL (I+1)	PROPEL	57
GO TO 18	PROPEL	58
16 AREAX = AREAR (I)	PROPEL	59
ARONE = AREAR (IONE)	PROPEL	60
ARTWO = AREAR (ITWO)	PROPEL	61
AREAP1 = AREAR (I+1)	PROPEL	62
18 CONTINUE	PROPEL	63
C CALCULATE THE ADJACENT GRID MASS BALANCE PARAMETERS	PROPEL	64
DMP11 = (1.0 - PHIBG(IONE,J))*DELT*ARONE*ABS(UPBDT(IONE,J))	PROPEL	65
DMP1 = (1.0 - PHIBG(I,J))*DELT*AREAX*ABS(UPBDT(I,J))	PROPEL	66
DMP12 = 0.0	PROPEL	67
C DETERMINE IF THE ADJACENT GRID LIES ON A BOUNDARY	PROPEL	68
IF(IONE.EQ.1) GO TO 20	PROPEL	69
IF(FLOAT(INCRE)*UPBDT(ITWO,J).LE.0.0) DMP12=(1.0 - PHIBG(ITWO,J))	PROPEL	70
1 *ABS(UPBDT(ITWO,J))*DELT*ARTWO	PROPEL	71
C PERFORM MASS BALANCE ON I+1*INCRE GRID	PROPEL	72
PHI11 = PHIBG(IONE,J) + (DMP11 - DMP12 - DMP1) / (ARONE*DX)	PROPEL	73
GO TO 30	PROPEL	74
20 PHI11 = PHIBG(IONE,J) + 2.0*(DMP11 - DMP1) / (ARONE*DX)	PROPEL	75
C IF THE RESULTING AMOUNT OF PROPELLANT IN THE I+INCRE GRID	MOD0925	492
C IS MORE THAN IT CAN HOLD, SET VELOCITIES EQUAL IN THE TWO GRIDS	MOD0925	493
30 CONTINUE	MOD1215	27
IF(ABS(UPI).LT. 0.001)UPI=0.0	MOD0925	495
UPBDT(I,J) = UPI	PROPEL	82
C CALCULATE PARAMETERS FOR THE ITH GRID MASS BALANCE.	PROPEL	83
DMP1P1 = 0.0	PROPEL	84
DMP1 = (1.0 - PHIBG(I,J))*DELT*AREAX*ABS(UPBDT(I,J))	PROPEL	85
DMP1M1 = 0.0	PROPEL	86
XLU = XL(I+1,J)	PROPEL	87
XLL = XL(I-1,J)	PROPEL	88
UUU = UU(I+1,J)	PROPEL	89
UUL = UU(I-1,J)	PROPEL	90
UIU = UI(I+1,J)	PROPEL	91
UIL = UI(I-1,J)	PROPEL	92
UPU = UPB(I+1,J)	PROPEL	93
UPL = UPB(I-1,J)	PROPEL	94
I2CU = I2C(I+1,J)	PROPEL	95
I2CL = I2C(I-1,J)	PROPEL	96
C DETERMINE IF THE ITH GRID LIES ON A BOUNDARY FOR	PROPEL	97
C SPECIAL TREATMENT.	PROPEL	98
IF(I.EQ.1) GO TO 40	PROPEL	99
IF(UPBDT(I+1,J).LT. 0.0) DMP1P1 = (1.0 - PHIBG(I+1,J))	PROPEL	100

1	*DELTA*AREAF1*ABS(UPBDT(1+1,J))	PROPEL	101
	IF (UPBDT(1-1,J) .GT. 0.0) DMPIM1 = (1.0 - PHIBG(1-1,J))	PROPEL	102
1	*DELTA*AREAF1*ABS(UPBDT(1-1,J))	PROPEL	103
	GO TO 50	PROPEL	104
40	IF (1 .EQ. 1 .AND. UPBDT(2,J) .LT. 0.0) DMPIP1 = 2.0*	PROPEL	105
1	(1.0 - PHIBG(2,J))*DELTA*AREAF1*ABS(UPBDT(2,J))	PROPEL	106
	DMP1 = 2.0 * DMP1	PROPEL	107
C	MASS BALANCE ON THE ITH GRID TO DETERMINE NEW VALUE	PROPEL	108
C	OF POROSITY	PROPEL	109
50	CONTINUE	MOD1215	28
	IF (1 .EQ. NGX .AND. NX .EQ. 1 .AND. UPBDT(I,J) .GT. 0.)	MOD1215	29
	* DMP1 = 0.	MOD1215	30
	IF (1 .EQ. NGX .AND. NX .EQ. 1) DMP1 = 0.	MOD1215	31
	PHI1 = PHIBG(I,J) + (DMP1-DMPIP1-DMPIM1)/(AREAX*DX)	MOD1215	32
	IF (1.NE.1) GO TO 62	PROPEL	111
	XLL = XLU	PROPEL	112
	DUL = DUU	PROPEL	113
	DIL = DIU	PROPEL	114
	UPL = 0.0	PROPEL	115
	I2CL=I2CU	PROPEL	116
62	IF (PHI1 .GT. .99999) GO TO 63	MOD1205	8
	UPBDT(1,J) = (UPBDT(I,J)*(1.0 - PHIBG(I,J))*AREAX*DX	PROPEL	118
1	+ UPU*DMPIP1 + UPL*DMPIM1 - UPBDT(I,J)*DMP1) /	PROPEL	119
2	((1.0 - PHI1)*AREAX*DX)	PROPEL	120
	IF (UPBDT(1,1) .LT. 0.) UPBDT(1,1)=0.	MOD0925	496
	IF (UPBDT(1,2) .LT. 0.) UPBDT(1,2)=0.	MOD0925	497
	XLDT(I,J) = (XL(I,J)*(1.0 - PHIBG(I,J))*AREAX*DX + XLU	PROPEL	121
1	*DMPIP1 + XLL*DMPIM1 - XL(I,J)*DMP1) / ((1.0 - PHI1)*AREAX*DX)	PROPEL	122
	IF (WHOLEC) DUTDT(1,J) = (DU(I,J)*(1.0-PHIBG(I,J))*AREAX*DX + DUU	PROPEL	123
1	*DMPIP1 + DUL*DMPIM1 - DU(I,J)*DMP1) / ((1.0 - PHI1)*AREAX*DX)	PROPEL	124
	IF (WHOLEC) DILDT(1,J) = (DI(I,J)*(1.0-PHIBG(I,J))*AREAX*DX + DIU	PROPEL	125
1	*DMPIP1 + DIL*DMPIM1 - DI(I,J)*DMP1) / ((1.0 - PHI1)*AREAX*DX)	PROPEL	126
	IF (I2C(1,J) .LT. 0.001) I2C(1,J) = I2CL	PROPEL	127
	IF (I2C(1,J) .LT. .116N) I2C(1,J) = (I2C(1,J)*(1.0-PHIBG(I,J))	PROPEL	128
	*AREAX*DX + I2CU*DMPIP1 + I2CL*DMPIM1	PROPEL	129
	-I2C(1,J)*DMP1)/(1.0-PHI1)*AREAX*DX)	PROPEL	130
63	CONTINUE	MOD0620	1
	PHIBTD(1,J)=PHI1	PROPEL	132
70	CONTINUE	MOD0925	498
C	COMPARE NEW GRAIN SIZE BASED ON A MASS AVERAGE	PROPEL	134
C	OF EXISTING PROPELLANT WITH THAT ADDED.	PROPEL	135
80	CONTINUE	MOD0620	2
C		PROPEL	137
	RETURN	PROPEL	138
	END	PROPEL	139

OLD HAVE RESULTED IN BETTER OPTIMIZATION

	SUBROUTINE DRAG(DRAGX, I, J)	DRAG	2
C		DRAG	3
C	SUBROUTINE DRAG CALCULATES CURRENT AND UPDATED VALUES FOR DRAG	DRAG	4
C	IN THE AXIAL DIRECTION AND UPDATES UICJ AND VTDI.	DRAG	5
C	THE ACTUAL DRAG USED IN THE FINITE DIFFERENCE CALCULATIONS IS	DRAG	6
C	AN AVERAGE OF THE CURRENT AND UPDATED VALUES.	DRAG	7
C		DRAG	8
	COMMON/BARRL/ PH(100), RHOG(100), HG(100), VG(100), UP(100),	BARRL	2
1	PG(100), TG(100), PMDT(100), GL(100), UCRAG(100), FRICT(100),	BARRL	3
2	WCONV(100), DUP(100), UPHI(100), DRUG(100), UHG(100), UUG(100),	BARRL	4
3	AMASS(100), AMOM(100), ALNER(100), LAMASS(100), UAMOM(100),	BARRL	5
4	UAENER(100)	BARRL	6
	COMMON/AVGDT/PHOTDT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	DRAG	10
5	UTDI,VHCAVE,VTDI	DRAG	11
	COMMON/CLOCK/TIME,DELT	DRAG	12
	COMMON/URGCUN/VISG,FRICT	MOD1215	33
	COMMON/GRAIN/ XL(60,5), DO(60,5), DI(60,5), FN,	GRAIN	2
1	XLDT(60,5), DDTDT(60,5), DTDI(60,5), XLO, DOO, DIO,	GRAIN	3
2	XLB(100), LOB(100), DIB(100), UXLB(100), LDOB(100), UDIB(100)	GRAIN	4
	COMMON/PRIMV/BPDENS,BPRAD(60,5),AGLEP,BGENEP,XPBP	PRIMV	2
	COMMON/BAG/PHIBG(60,5), RHORG(60,5), RBG(60,5), UBG(60,5),	BAG	2
1	VBP(60,5), VFB(60,5), PCH(60,5), IZC(60,5),	BAG	3
2	DUIMIG(60,5), GRAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
3	PHIBTD(60,5), RHBTDT(60,5), HBGTDT(60,5), UBGTD(60,5),	BAG	5
4	VBTDT(60,5), TBG(60,5), DOTMBG(60,5), DUTMP(60,5), PHIBP(60,5),	BAG	6
5	PHIPTD(60,5), IZK(60,5), TBP(60,5)	BAG	7
	LOGICAL INB	DRAG	17
	DATA GRAV/32.16/	DRAG	18
C		DRAG	19
C	THE LOGICAL VARIABLE INB IS .TRUE. IF SUBROUTINE DRAG WAS CALLED	DRAG	20
C	FROM A ONE-DIMENSIONAL SYSTEM WHERE THERE IS NO RADIAL VELOCITY	DRAG	21
C	AND .FALSE. IF CALLED FROM A SYSTEM WHERE THERE IS RADIAL VELOCITY.	DRAG	22
C		DRAG	23
C		DRAG	24
C	A AND B ARE THE SAME IN BOTH THE AXIAL AND RADIAL CALCULATIONS.	DRAG	25
C	THE ROUTINE CALLING DRAG SHOULD HAVE ALREADY CHECKED THAT PHIAVE	DRAG	26
C	IS NOT 1.0.	DRAG	27
	IF(INB) GO TO 5	DRAG	28
	DM=(1.5*DO(I,J)*DC(I,J)*XL(I,J))	DRAG	29
	IF(DO(I,J).LT.0.1) DM=0.844*(DOO-DIO)**2*XL(I,J)	DRAG	30
	IF(DM.LE.0.) DM=1.E-5	DRAG	31
	DM=DM**0.333	DRAG	32
	VPROP=0.7854*XL(I,J)*(DO(I,J)*DO(I,J)-7.0*DI(I,J)*DI(I,J))	DRAG	33
	IF(VPROP.LT.1.E-5)VPROP=1.E-5	DRAG	34
	VBP=4.189*BPRAD(I,J)*BPRAD(I,J)*BPRAC(I,J)	DRAG	35
	IF(VBP .LT.1.E-5)VBP=1.E-5	DRAG	36
	TERMP=(PHIBP(I,J)-PHIBG(I,J))/VPROP	DRAG	37
	TERMBP=(1.0-PHIBP(I,J))/VBP	DRAG	38
	IF((TERMP+TERMBP).LT.1.E-5)TERMP=1.E-5	DRAG	39
	VISG=32.16*(DM*TERMP+2.0*BPRAD(I,J)*TERMBP)/(TERMP+TERMBP)	DRAG	40
	GO TO 7	DRAG	41
5	DM=1.5*DOB(I)*DOB(I)*XLB(I)	DRAG	42
	IF(DOB(I).LT.0.1) DM=0.844*(DOO-DIO)**2*XLB(I)	DRAG	43
	IF(DM.LE.0.) DM=1.E-5	DRAG	44
	DM=DM**0.333	DRAG	45
	VISG=32.16*DM	DRAG	46
7	CONTINUE	DRAG	47

IF (VISGG.GT.VTSG)VISGG=VTSG	DRAG	48
IF(PHIAVE .LT. 0.1) PHIAVE=0.1	MOD0925	49
SHAPE=0.7	MOD1018	4
IF(J .EQ. 1 .OR. DO(I,J) .LE. 0. .OR. XL(I,J) .LE. 0.) GO TO 9	MOD1018	5
DEFF=(1.5*DO(I,J)*DO(I,J)*XL(I,J))*0.333	MOD1018	6
SHAPE=DEFF*DEFF*(DO(I,J)*(XL(I,J)+.5*DO(I,J)))	MOD1018	7
IF(SHAPE .LT. 0.1) SHAPE = 0.1	MOD1018	8
9 CONTINUE	MOD1018	9
CONST = 2.0*PCTF*(1.-PHIAVE)/(SHAPE*PHIAVE*VTSG)	MOD1215	34
C	DRAG	50
C	URAG	51
A = CONST*RHUTDT	URAG	52
B = 2.0*PHIRHO/(DLT*GRAV)	DRAG	53
C	URAG	54
C THE FOLLOWING CALCULATIONS INCORPORATE THE SIMULTANEOUS SOLUTIONS	DRAG	55
C OF THE DRAG EQUATION AND FINITE DIFFERENCE EQUATIONS FOR UBG	URAG	56
C TO CALCULATE THE UPDATED QUANTITY FOR UBG EXPLICITLY,	DRAG	57
C CALCULATIONS ARE DONE IN THE FOLLOWING SEQUENCE (DRXT AND	DRAG	58
C DRXTDT ARE THE CURRENT AND UPDATED DRAG IN THE AXIAL DIRECTION)	DRAG	59
C 1) COMPUTE DRXT	DRAG	60
C 2) COMPUTE UTDI = UPBAVE, (CMU), WHICH IS THE	DRAG	61
C SOLUTION OF A QUADRATIC AND THEN GET UTDI	DRAG	62
C 3) COMPUTE DRXTDT	DRAG	63
C 4) COMPUTE DRAGX	DRAG	64
C	URAG	65
DIFFU = UBGAVE - UPBAVE	DRAG	66
IF(ABS(DIFFU).GT..0001)GO TO 10	DRAG	67
DRAGX = 0.0	URAG	68
GO TO 40	DRAG	69
C	URAG	70
10 DRXT = CONST*RHOAVE*DIFFU*ABS(DIFFU)	DRAG	71
C = DRXT - B*(UTDT-UPBAVE)	DRAG	72
DISCRM = B*B - 4.0*A*C	DRAG	73
IF(DISCNM.LT.0.0)GO TO 20	DRAG	74
UMU = (-B + SQRT(DISCNM))/(A+A)	DRAG	75
GO TO 30	URAG	76
C	URAG	77
20 DISCRM = B*B + 4.0*A*C	DRAG	78
UMU = (-B - SQRT(DISCNM))/(A+A)	DRAG	79
C	URAG	80
30 UTDI = UMU + UPBAVE	DRAG	81
DRXTDT = A*UMU*ABS(UMU)	DRAG	82
DRAGX = (DRXT + DRXTDT)*0.5	DRAG	83
C	URAG	84
C	DRAG	85
40 IF(INB) RETURN	URAG	86
C	DRAG	87
C UPDATE VIDI. DRAGX DOES NOT NEED TO BE CALCULATED SINCE IT IS	URAG	88
C NOT USED EXPLICITLY IN THE FINITE DIFFERENCE EQUATIONS.	DRAG	89
IF(ABS(VBGAVE) .LE. 0.0001) RETURN	URAG	90
C	DRAG	91
DRXT = CONST*RHOAVE*VBGAVE*ABS(VBGAVE)	URAG	92
C = DRXT - B*VTDT	DRAG	93
DISCRM = B*B - 4.0*A*C	DRAG	94
IF(DISCNM.LT.0.0)GO TO 60	DRAG	95
VTDT = (-B + SQRT(DISCNM))/(A+A)	URAG	96
RETURN	DRAG	97

```

C
  BV DISCRP = B*B + 4.0*A*C
    VTUT = (B - SQRT(DISCRP))/(A+A)
    RETURN
C
  END

```

```

DRAG 98
DRAG 99
DRAG 100
DRAG 101
DRAG 102
DRAG 103

```

SUBROUTINE GSPROP(R0,KR0,K,CV0,CVH,CV,P,H,T,RHO,U,V,GAM,CP,IPROP)		GSPROP	2
COMMON/EQNS/DUX,T2UR,T2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,		EQNS	2
S DUX,DR,NX,GJ,TWOUT,HBP		EQNS	3
DATA XJUL/778.0/		GSPROP	4
C	IPROP = 1 - GIVEN T AND P	GSPROP	5
C	IPROP = 2 - GIVEN H AND RHO	GSPROP	6
C	IPROP = 3 - GIVEN H AND P	GSPROP	7
C		GSPROP	8
C	H, CV, CP, AND GAM ARE TO BE CALCULATED. ALSO, OF T, P, H, AND RHO	GSPROP	9
C	THE TWO THAT ARE NOT GIVEN ARE TO BE CALCULATED.	GSPROP	10
C		GSPROP	11
C	GO TO (10,20,30),IPROP	GSPROP	12
C		GSPROP	13
C	THE EQUATIONS FOR IPROP=1 ARE VALID ASSUMING GAS VELOCITY IS 0.	GSPROP	14
10	R = 0.5*R0 + SQRT(0.25*R0*R0 + RRC*P/I)	GSPROP	15
	ATEMP = T*CVH	GSPROP	16
	CV = (CV0 + ATEMP*R/XJUL)/(1.0 - ATEMP)	GSPROP	17
	CP = R/XJUL + CV	GSPROP	18
	GAM = CP/CV	GSPROP	19
	H = T*CP	GSPROP	20
	RHO=P/(R*T)	GSPROP	21
	RETURN	GSPROP	22
C		GSPROP	23
20	IF(RHO .LE. 0.) RHO = .00001	GSPROP	24
	R=R0 + KR0*RHO	MOD0925	501
	IF(H .LE. 0.) H=.1	MOD0925	502
	CV = CV0 + CVH*H	MOD0925	503
	CP = R/XJUL + CV	GSPROP	26
	GAM = CP/CV	GSPROP	27
	T = H/CP	GSPROP	28
	P = XJUL*(GAM - 1.0)/GAM*RHO*(H - (U*U+V*V)/TWOGJ)	GSPROP	29
	IF(P .LE. 0.1) P=0.1	GSPROP	30
	RETURN	MOD0925	504
C		GSPROP	31
30	IF(H .LE. 0.) H=.1	GSPROP	32
	CV=CV0 + CVH*H	MOD0925	505
	ATEMP = RHO*T/(H - (U*U+V*V)/TWOGJ)	MOD0925	506
	YTEMP = 0.5*(R0 + ATEMP/XJUL)	GSPROP	34
	R = YTEMP + SQRT(YTEMP*YTEMP + ATEMP*CV)	GSPROP	35
	CP = CV + R/XJUL	GSPROP	36
	GAM = CP/CV	GSPROP	37
	T = H/CP	GSPROP	38
	RHO=P/(R*T)	GSPROP	39
	RETURN	GSPROP	40
	END	GSPROP	41
		GSPROP	42

	SUBROUTINE AXIS	AXIS	2
C	1X IS 1 HERE, IM IS J	AXIS	3
	COMMON/AVGUT/RHOTOT,PHIRHO,PHIAVF,RHCAVE,UBCAVE,UPBAVE,	AXIS	4
3	UIU1,VBGAVE,VTOT	AXIS	5
	COMMON/FAILED/THICKT,PHOUP,PCOMP,BTLE,XNTUB,FAIL,MFAIL(60),	MOD0925	507
1	THICK(60),PLNET,IYFCE	MOD0925	508
	COMMON/CHAM/1,J,XB,RB,NGX,NGK,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAM1	2
3	AREACH,AREAL(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3	AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAM1	4
3	AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
	COMMON/CLOCK/TIME,BELT	AXIS	7
	COMMON/EQNS/DTUX,T2UR,T2UX,TWOTUR,DILN,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
3	UX,UR,NX,GJ,TWOT,MBP	EQNS	3
	COMMON/CASCON/KO,KKO,CVQ,CVH	AXIS	9
	COMMON/INPUTS/C1,C2,C3,C4,I0,TIGN,CLCONS,RHOF,PHIO,TF,CA,RHOO,	MOD0925	509
3	HU,PU,UO,GTRHOP,HU,TIGNBP,QBCONS,TOTM	MOD0925	510
	COMMON/HAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1	VBG(60,5),UPB(60,5),PCH(60,5),IZC(60,5),	BAG	3
2	DOTMBG(60,5),QBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	HAG	4
3	PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	HAG	5
4	VBGTD(60,5),TBG(60,5),DOTMBG(60,5),DOTMP(60,5),PHIBP(60,5),	BAG	6
5	PHIBTD(60,5),TZR(60,5),TBP(60,5)	HAG	7
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL PLNET	MOD0925	511
	DATA GRAV/32.16/	MOD0925	512
C	KB = 0.0	AXIS	13
	AB=AB+UX	AXIS	14
C		AXIS	15
	CALL GSPROP(KO,KKO,K,CVQ,CVH,CV,PCH(1,J),HBE(1,J),TDUM,	AXIS	16
3	RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	AXIS	17
	BUGGER = (GAP - 1.0)/TWOGJ	AXIS	18
	IP1 = 1+1	AXIS	19
	IM1 = 1 -1	AXIS	20
C		AXIS	21
C		AXIS	22
C	IN THIS SUBROUTINE PHIBG REPRESENTS IFL TOTAL POROSITY, NOT JUST	AXIS	23
C	POROSITY OF THE PROPELLANT.	AXIS	24
	F5 = PHIBG(1,J)	AXIS	25
	A5 = AREAC(1)	AXIS	26
	G2 = RHOBG(IM1,J)	AXIS	27
	G4 = RHOBG(IP1,J)	AXIS	28
	G5 = RHOBG(1,J)	AXIS	29
	L2 = G2*UBG(IM1,J)	AXIS	30
	L4 = G4*UBG(IP1,J)	AXIS	31
	L5 = G5*UBG(1,J)	AXIS	32
	H2 = PHIBG(IM1,J) * AREAC(IM1) * E2	AXIS	33
	H4 = PHIBG(IP1,J) * AREAC(IP1) * E4	AXIS	34
	C2 = G2 * HBG(IM1,J)	AXIS	35
	C4 = G4 * HBG(IP1,J)	AXIS	36
	C5 = G5 * HBG(1,J)	AXIS	37
	B2 = BUGGER * UBG(IM1,J) * L2	AXIS	38
	B4 = BUGGER * UBG(IP1,J) * L4	AXIS	39
	B5 = BUGGER * UBG(1,J) * L5	AXIS	40
C		AXIS	41
	PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25	AXIS	42
	RHUAVE = (G2 + G5 + G5 + G4)*0.25	AXIS	43
		AXIS	44

	UBGAVE = (UBG(IP1,J) + UBG(I,J) + UBG(1,J) + UBG(IP1,J))*0.25	AXIS	45
	VUBAVE = 0.0	AXIS	46
	UPBAVE = (UPB(IP1,J) + UPB(I,J) + UPB(1,J) + UPB(IP1,J))*0.25	AXIS	47
	PHITDT = PHIBTD(I,J) + PHIPTD(I,J) - 1.0	AXIS	48
	RHUTDT = (F5*RHOAVE - T2UX*(H4 - H2)/A5 + DELT*DOTMIG(I)/UVAXIS	AXIS	49
	+ DOTMB(I,J) + DOTMP(I,J))/PHITDT	AXIS	50
	PHIRHU = PHITDT*RHUTDT	AXIS	51
	TERM=0.0	AXIS	52
	IF (DOTMIG(I).LT.0.0)TERM=1.0	AXIS	53
		AXIS	54
	UTDT = (F5*(E2 + E5 + E5 + E4)*0.25	AXIS	55
1	- T2UX*(H4*UBG(IP1,J) - H2*UBG(IM1,J))	AXIS	56
2	+ GRAV*AREAC(I)*PCM(IP1,J)	AXIS	57
3	- GRAV*AREAC(I)*PCM(IP1,J))/A5	AXIS	58
4	+ DELT*DOTMIG(I)*UBG(1,J)/UVAXIS*TERF	AXIS	59
	+ DOTMB(I,J)*UPB(I,J))/PHIRHU	AXIS	60
	IF (ABS(UTDT).LT. 0.1)UTDT = 0.0	AXIS	61
		AXIS	62
	VTDT = 0.0	AXIS	63
		AXIS	64
	IF (PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)	AXIS	65
		AXIS	66
	HIGN = HBG(I,1)	AXIS	67
	IF (DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)	AXIS	68
		AXIS	69
	HBTDT(I,J) = GAM*(F5*(C2+B2+C5+C5+H5+B5+C4+B4)/(4.0*GAM)	AXIS	70
1	- T2UX*(H4*HBG(IP1,J) - H2*HBG(IM1,J))/A5	AXIS	71
2	+ DELT*(DOTMIG(I)/UVAXIS*HIGN - QBA6(I,J))	AXIS	72
3	+ DOTMB(I,J)*HMB + DOTMP(I,J)*HEP	AXIS	73
4	- XDRAG(I,J)*UPB(I,J)*DELT/778.	AXIS	74
	- BUGGER*PHIRHU*UTDT*UTDT/GAM)/PHIRHO	AXIS	75
		AXIS	76
	RHUBTD(I,J) = RHOTDT	AXIS	77
	UBGTD(I,J) = UTDT	AXIS	78
	VBGTD(I,J) = VTDT	AXIS	79
	RETURN	AXIS	80
	END	AXIS	81
		AXIS	82

	SUBROUTINE AXIT	AXIT	2
C	1X IS 1 HERE, IR IS J	AXIT	3
	COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	AXIT	4
	3 UDT,VBGAVE,VTDT	AXIT	5
	COMMON/CHAM/1,J,XB,RB,NGX,NGR,IBLGB,IENDB,IPATH(60,5),AREA6(5),	CHAM1	2
	3 AREACH,AREAC(60),IGN11,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
	3 AREAK(60),AREAA,CHAM1,CHAM2,CHAM3,DPGAP,AREAGP(60),DAVG,	CHAM1	4
	3 AREAM2,DIAMB1,BELENB,BELBFG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
	COMMON/CLUCK/TIME,DELT	AXIT	7
	COMMON/EONS/DTUX,T2UR,T2UX,TWOTR,DILR,HMB,TWOGU,DVAXIS,DVAXIT,	EONS	2
	3 UX,UR,NX,GU,TWODT,HBP	EONS	3
	COMMON/GASCOR/RG,RRO,CV,CVH	AXIT	9
	COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
	1 VBG(60,5),UPB(60,5),PCH(60,5),LXC(60,5),	BAG	3
	2 DOTMIG(60),UBAG(60,5),XDRA(60,5),DOTMB(60,5),UPBDT(60,5),	BAG	4
	3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
	4 VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),	BAG	6
	5 PHIPTD(60,5),TZR(60),TBP(60,5)	BAG	7
	LOGICAL IGN11,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	DATA GRAY/32,16/	AXIT	12
C	KB = KB + DR	AXIT	13
	CALL GSPROP(KO,RRO,K,CV,CVH,CV,PCH(1,J),HBG(1,J),TDUM,	AXIT	14
	3 RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	AXIT	15
	BUGGER = (GAP - 1.0)/TWOGU	AXIT	16
	IP1 = 1+1	AXIT	17
	IM1 = 1-1	AXIT	18
	JP1 = J+1	AXIT	19
	F1 = PHIBG(1,JP1)	AXIT	20
	F2 = PHIBG(IM1,J)	AXIT	21
	F4 = PHIBG(IP1,J)	AXIT	22
	F5 = PHIBG(1,J)	AXIT	23
	G1 = F1*RHOBG(1,JP1)	AXIT	24
	G2 = F2*RHOBG(IM1,J)	AXIT	25
	G4 = F4*RHOBG(IP1,J)	AXIT	26
	G5 = F5*RHOBG(1,J)	AXIT	27
	E1 = G1*UBG(1,JP1)	AXIT	28
	E2 = G2*UBG(IM1,J)	AXIT	29
	E4 = G4*UBG(IP1,J)	AXIT	30
	E5 = G5*UBG(1,J)	AXIT	31
	U1 = G1*VBG(1,JP1)	AXIT	32
	C1 = G1*HBG(1,JP1)	AXIT	33
	C2 = G2*HBG(IM1,J)	AXIT	34
	C4 = G4*HBG(IP1,J)	AXIT	35
	C5 = G5*HBG(1,J)	AXIT	36
	B1 = BUGGER*(UBG(1,JP1)**2 + VBG(1,JP1)**2)*G1	AXIT	37
	B2 = BUGGER*(UBG(IM1,J)**2 + VBG(IM1,J)**2)*G2	AXIT	38
	B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2)*G4	AXIT	39
	B5 = BUGGER*(UBG(1,J)**2 + VBG(1,J)**2)*G5	AXIT	40
C*		AXIT	41
	DENOM = F1 + F2 + F4 + F5	AXIT	42
	SIG6 = G1 + G2 + G4 + G5	AXIT	43
	SIGL = E1 + E2 + E4 + E5	AXIT	44
	PHIAVE = DENOM*0.25	AXIT	45
	RHUAVE = SIG6/DENOM	AXIT	46
	UBGAVE = SIGL/SIG6	AXIT	47
	VBGAVE = 0.0	AXIT	48
		AXIT	49

C	UPBAVE = (UPB(1,J) + UPB(1,JP1) + UPB(IM1,J) + UPB(IP1,J))*0.25	AXIT	50
	PHITDT = PHITDT(1,J)	AXIT	51
	TERM=0.0	AXIT	52
	IF(DOTMIG(1).GT.0.0)TERM=-1.0	AXIT	53
	RHUTDT = (F5*KHOAVE - T2DX*(E4-E2) - DTDR*C1	AXIT	54
	1 - DELT*DOTMIG(1)/DVAXIT + DOTPB(1,J))/PHITDT	AXIT	55
	PHIRHO = PHITDT*RHUTDT	AXIT	56
	UTDT = (F5*SIGL/DENOM - T2DX*(E4*UBG(IP1,J) - E2*UBG(IM1,J)	AXIT	57
	3 + GRAV *PCH(IP1,J) - GRAV *FCH(IM1,J))	AXIT	58
	3 + DELT*DOTMIG(1)*UBG(1,J)/DVAXIS*TERM	AXIT	59
	3 - DTDR*C1*UBG(1,JP1) - DOTPB(1,J)*UPE(1,J))/PHIRHO	AXIT	60
	IF(ABS(UTDT) .LT. 0.1) UTDT = 0.0	AXIT	61
C*		AXIT	62
C*		AXIT	63
	VTDT = 0.0	AXIT	64
	IF(PHIAVE.LT.0.999)CALL URAG(XDRAG(1,J),.FALSE.,1,J)	AXIT	65
C*		AXIT	66
	HIGN = HBG(1,1)	AXIT	67
C*		AXIT	68
	IF(DOTMIG(1) .GT. 0.00001) HIGN = HBG(1,2)	AXIT	69
	HMGTD(1,J) = GAM*(F5*(C1+B1+C2+B2+C4+B4+C5+B5)/(DENOM*GAM)	AXIT	70
	3 -T2DX*(C4*UBG(IP1,J) - C2*UBG(IM1,J))	AXIT	71
	3 - DTDR*C1*VBG(1,JP1)	AXIT	72
	3 - DELT*(DOTMIG(1)/DVAXIT*HIGN + QBAG(1,J))	AXIT	73
	3 -XDRAG(1,J)*UPB(1,J)*DELT/788.	AXIT	74
	3 + DOTPB(1,J)*HMB - BUGGER*UTDT*UTDT*PHIRHO/GAM)/PHIRHO	AXIT	75
C*		AXIT	76
	RHUTDT(1,J) = RHUTDT	AXIT	77
	UBGTD(1,J) = UTDT	AXIT	78
	VBGTD(1,J) = VTDT	AXIT	79
	RETURN	AXIT	80
	END	AXIT	81
		AXIT	82

SUBROUTINE INTER	INTER	2
COMMON/AVGDT/RHGTOT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPHAVE,	INTER	3
1 UTDI,VBGAVE,VTDT	INTER	4
COMMON/CHAM/1,J,XB,HB,NGX,NGK,IBEGE,IENDB,IPATH(60,5),AREAG(5),	CHAMI	2
3 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMI	3
3 AREAR(60),AREAX,CHAM1,CHAMP,CHAM3,IOPGAP,AREAGP(60),DAYG,	CHAMI	4
3 AREAH2,DIAMBT,BELEND,BELBEC,IPS1,IPS2,RADFS,BPIGN	CHAMI	5
COMMON/LOCK/IME,DEL1	INTER	6
COMMON/EUNS/LTDX,T2DR,T2DX,TWOTDR,LICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EUNS	2
3 DX,DR,IX,GJ,IKODT,HBP	EUNS	3
COMMON/GASCON/RB,RKG,CVO,CVH	INTER	8
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),IZC(60,5),	BAG	3
2 DUINIG(60),WBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTHBG(60,5),CUTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPTD(60,5),TZR(60),TBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
DATA GRAV/32.16/	INTER	11
C	INTER	12
RB = RB + DR	INTER	13
CALL GSPRUP(RK,RKO,K,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,	INTER	14
3 RHOBG(I,J),UBG(I,J),VBG(I,J),GAR,CP,2)	INTER	15
BUGGER = (GAP - 1.0)/TWOGJ	INTER	16
IP1 = I+1	INTER	17
IM1 = I-1	INTER	18
JP1 = J+1	INTER	19
JM1 = J-1	INTER	20
F1 = PHIBG(I,JP1)	INTER	21
F2 = PHIBG(IM1,J)	INTER	22
F3 = PHIBG(I,JM1)	INTER	23
F4 = PHIBG(IP1,J)	INTER	24
F5 = PHIBG(I,J)	INTER	25
G1 = F1*RHOBG(I,JP1)	INTER	26
G2 = F2*RHOBG(IM1,J)	INTER	27
G3 = F3*RHOBG(I,JM1)	INTER	28
G4 = F4*RHOBG(IP1,J)	INTER	29
E1 = G1*UBG(I,JP1)	INTER	30
E2 = G2*UBG(IM1,J)	INTER	31
E3 = G3*UBG(I,JM1)	INTER	32
E4 = G4*UBG(IP1,J)	INTER	33
U1 = G1*VBG(I,JP1)	INTER	34
U2 = G2*VBG(IM1,J)	INTER	35
U3 = G3*VBG(I,JM1)	INTER	36
U4 = G4*VBG(IP1,J)	INTER	37
C1 = G1*HBG(I,JP1)	INTER	38
C2 = G2*HBG(IM1,J)	INTER	39
C3 = G3*HBG(I,JM1)	INTER	40
C4 = G4*HBG(IP1,J)	INTER	41
B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1	INTER	42
B2 = BUGGER*(UBG(IM1,J)**2 + VBG(IM1,J)**2)*G2	INTER	43
B3 = BUGGER*(UBG(I,JM1)**2 + VBG(I,JM1)**2)*G3	INTER	44
B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2)*G4	INTER	45
URR = (RB + DR)/RB	INTER	46
URR = (RB - DR)/RB	INTER	47
L*	INTER	48
UENOM = F1 + F2 + F3 + F4	INTER	49

	SIGG = G1 + G2 + G3 + G4	INTER	50
	SIGU = U1 + U2 + U3 + U4	INTER	51
	SIGL = L1 + L2 + L3 + L4	INTER	52
	PHIAVE = DENOM*0.25	INTER	53
	RHOAVE = SIGG/DENOM	INTER	54
	UBGAVE = SIGL/SIGG	INTER	55
	VBGAVE = SIGU/SIGG	INTER	56
	UPBAVE = (UPB(I,JP1) + UPB(IM1,J) + UPB(I,JM1) + UPB(IP1,J))*0.25	INTER	57
	AVPHI = F5/DENOM	INTER	58
C*		INTER	59
	PHITDT = PHIBTD(I,J)	INTER	60
	RHOTDT = (F5*RHOAVE - T2DX*(E4 - L4)	INTER	61
	- T2DX*(DRP*L1 - DRM*U3)	INTER	62
	+ DOTMB(I,J))/PHITDT	INTER	63
	PHIRHO = PHITDT*RHOTDT	INTER	64
	UTDT = (AVPHI*SIGL - T2DX*(E4*UBG(IP1,J) - E2*UBG(IM1,J)	INTER	65
	+ GRAV *PCH(IP1,J) - GRAV *PCH(IM1,J))	INTER	66
	- T2DX*(DRP*L1*VBG(I,JP1) - DRM*L3*VBG(I,JM1))	INTER	67
	+ DOTMB(I,J)*UPB(I,J))/PHIRHO	INTER	68
	IF(ABS(UTDT) .LT. 0.1) UTDT = 0.0	INTER	69
C*		INTER	70
	VTDT = (AVPHI*SIGU	INTER	71
	- T2DX*(E4*VBG(IP1,J) - E2*VBG(IM1,J))	INTER	72
	- T2DX*(DRP*L1*VBG(I,JP1) + GRAV *PCH(I,JP1))	INTER	73
	- DRM*(L3*VBG(I,JM1) + GRAV *PCH(I,JM1)))	INTER	74
	+ DELT*GRAV*PCH(I,J)/RB)/PHIRHO	INTER	75
	IF(ABS(VTDT) .LT. 0.1) VTDT = 0.0	INTER	76
C*		INTER	77
C	ARRAY XDRAG IS CLEARED IN MAIN AT THE BEGINNING OF EACH TIME	INTER	78
C	INTERVAL SO IF SUBROUTINE DRAG IS NOT CALLED, XDRAG(I,J) IS 0.0.	INTER	79
	IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J), .FALSE.)	INTER	80
C*		INTER	81
C*		INTER	82
	HBBGTD(I,J) = GAM*(AVPHI*(C1+B1+C2+B2+L3+B3+C4+B4)/GAM	INTER	83
	- T2DX*(L4*UBG(IP1,J) - C2*UBG(IM1,J))	INTER	84
	- T2DX*(DRP*L1*VBG(I,JP1) - DRM*L3*VBG(I,JM1))	INTER	85
	- DELT*UBG(I,J) + DOTMB(I,J)*HMB	INTER	86
	- BUGGER*(UTDT*UTDT + VTDT*VTDT)*PHIRHO/GAM) / PHIRHO	INTER	87
C*		INTER	88
C		INTER	89
	RHOBTDT(I,J) = RHOTDT	INTER	90
	UBGTD(I,J) = UTDT	INTER	91
	VBGTD(I,J) = VTDT	INTER	92
	RETURN	INTER	93
	END	INTER	94

SUBROUTINE BNDY	BNDY	2
COMMON/AVGDT/RHODT,PHIRHO,PHIAVE,RHOAVE,UBAVE,UPBAVE,	BNDY	3
3 UTDT,VBGAVE,VTDT	BNDY	4
COMMON/CHAM/I,J,XB,RB,NGX,NGH,IBELB,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
4 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
5 AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAM1	4
6 AREAM2,DIAM1,HELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLOCK/TIME,DEL1	BNDY	6
COMMON/EGNS/DTUX,T2DX,T2DX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EGNS	2
7 DX,DR,IX,GU,TWODI,HBP	EGNS	3
COMMON/GASCON/RG,KRO,CVO,CVH	BNDY	8
COMMON/EAG/PHIBG(60,5),RHOBG(60,5),HBBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCB(60,5),IC(60,5),	BAG	3
2 DUMIG(60),GHAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBUT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBBTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPTD(60,5),TZK(60),TBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
DATA GRAV/32.16/	BNDY	11
C	BNDY	12
RH=RH+DR	BNDY	13
C	BNDY	14
CALL GSPROP(RG,KRO,R,CVO,CVH,CV,PCH(I,J),HBB(I,J),TDM,	BNDY	15
5 RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)	BNDY	16
BUGGER = (GAM - 1.0)/TWOGJ	BNDY	17
IP1 = 1+1	BNDY	18
IM1 = 1 - 1	BNDY	19
JM1 = J - 1	BNDY	20
F2 = PHIBG(IM1,J)	BNDY	21
F3 = PHIBG(1,JM1)	BNDY	22
F4 = PHIBG(IP1,J)	BNDY	23
F5 = PHIBG(1,J)	BNDY	24
G2 = F2*RHOBG(IM1,J)	BNDY	25
G3 = F3*RHOBG(1,JM1)	BNDY	26
G4 = F4*RHOBG(IP1,J)	BNDY	27
E2 = G2*UBG(IM1,J)	BNDY	28
E3 = G3*UBG(1,JM1)	BNDY	29
E4 = G4*UBG(IP1,J)	BNDY	30
J3 = G3*VBG(1,JM1)	BNDY	31
C2 = G2*HBB(IM1,J)	BNDY	32
C3 = G3*HBB(1,JM1)	BNDY	33
C4 = G4*HBB(IP1,J)	BNDY	34
G2 = BUGGER*UBG(IM1,J)**2*G2	BNDY	35
G3 = BUGGER*(UBG(1,JM1)**2 + VBG(1,JM1)**2)*G3	BNDY	36
G4 = BUGGER*UBG(IP1,J)**2*G4	BNDY	37
C*	BNDY	38
DENOM = F2 + F3 + F3 + F4	BNDY	39
SIG6 = G2 + G3 + G3 + G4	BNDY	40
SIGL = E2 + E3 + E3 + E4	BNDY	41
PHIAVE = DENOM*0.25	BNDY	42
RHOAVE = SIG6/DENOM	BNDY	43
UBGAVE = SIGL/SIG6	BNDY	44
VBGAVE = U.U	BNDY	45
UPBAVE = (UPB(IM1,J) + 2.0*UPB(1,JM1) + UPB(IP1,J))*0.25	BNDY	46
C*	BNDY	47
PHIBTD = PHIBTD(1,J)	BNDY	48
RHUTDT = (F5*RHOAVE - T2DX*(E4-E2) + DTDR*C3	BNDY	49

	3	+ DOTMB(1,J))/PHITDT	BNDY	50
		PHIKHU = PHITDT*RHOTDT	BNDY	51
C*			BNDY	52
		UTDT = (F5*SIG/DENOM - T2DX*(C4*UBG(IP1,J) - E2*UBG(IM1,J)	BNDY	53
	3	+ GRAV *PCH(IP1,J) - GRAV *FCH(IM1,J))	BNDY	54
	3	+ DTDK*E3*VBG(1,JM1)	BNDY	55
	3	+ DOTMB(1,J)*UPB(1,J))/PHIKHU	BNDY	56
		IF(ABS(UTDT) .LT. 0.1) UTDT = 0.0	BNDY	57
		VTDT=0.0	BNDY	58
C*			BNDY	59
		IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J), .FALSE.)	BNDY	60
C*			BNDY	61
C*			BNDY	62
		MBGTD(1,J) = GAM*(F5*(C2+B2+C3+C3+B3+B3+C4+B4)/(DENOM*GAM)	BNDY	63
	3	-T2DX*(C4*UBG(IP1,J) - C2*UBG(IM1,J))	BNDY	64
	3	+ DTDK*C3*VBG(1,JM1)	BNDY	65
	3	- DELT*QBAG(1,J) + DOTMB(1,J)*AMB	BNDY	66
	3	- BUGGER*UTDT*UTDT*PHIRHO/GAM)/PHIRHO	BNDY	67
C*			BNDY	68
C			BNDY	69
		RHOTDT(I,J) = RHOTDT	BNDY	70
		UBGTD(I,J) = UTDT	BNDY	71
		VBGTD(I,J) = VTDT	BNDY	72
		RETURN	BNDY	73
		END	BNDY	74

SUBROUTINE BSURFA	BSURFA	2
C 1X IS 1 HKE, 1R IS J	BSURFA	3
COMMON/AVGOT/RHOTD,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	BSURFA	4
3 UTOT,VBGAVE,VTOT	BSURFA	5
COMMON/CHAM/1,J,XB,RB,NGX,NGH,IBLGB,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
3 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAM1	4
3 AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RAQFS,BPIGN	CHAM1	5
COMMON/CLOCK/TIME,DELT	BSURFA	7
COMMON/EGNS/DUX,TZUR,TZUX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EGNS	2
3 DX,UR,NX,GJ,TWOTD,HDP	EGNS	3
COMMON/GACCON/RQ,RRO,CVU,CVH	BSURFA	9
COMMON/PAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),JZC(60,5),	BAG	3
2 DUTHI(60),QBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60),CUTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPID(60,5),TZK(60),TBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C RB=0.0	BSURFA	12
XH = XB + DX	BSURFA	13
C CALL GSPROP(RQ,RRO,R,CVU,CVH,CV,PCH,1,J),HBG(1,J),TDUM,	BSURFA	14
3 RHOBG(1,J),UBG(1,J),VBG(1,J),GAR,CP,2)	BSURFA	15
BUGGER = (GAR - 1.0)/TWOGJ	BSURFA	16
IP1 = 1+1	BSURFA	17
JP1 = J+1	BSURFA	18
F1 = PHIBG(1,JP1)	BSURFA	19
F4 = PHIBG(IP1,J)*AREAC(IP1)	BSURFA	20
F5 = PHIBG(1,J)*AREAC(1)	BSURFA	21
G1 = F1*RHOBG(1,JP1)	BSURFA	22
G4 = F4*RHOBG(IP1,J)	BSURFA	23
E4 = G4*UBG(IP1,J)	BSURFA	24
U1 = G1*VBG(1,JP1)	BSURFA	25
C1 = G1*HBG(1,JP1)	BSURFA	26
C4 = G4*HBG(IP1,J)	BSURFA	27
G1 = BUGGER*(UBG(1,JP1)**2 + VBG(1,JP1)**2)*G1	BSURFA	28
G4 = BUGGER*UBG(IP1,J)**2*G4	BSURFA	29
C* DENOM = F1 + F4	BSURFA	30
PHIAVE = DENOM/(AREAC(1) + AREAC(IP1))	BSURFA	31
RHUAVE = (G1+G4)/DENOM	BSURFA	32
UBCAVE = 0.0	BSURFA	33
VBGAVE = 0.0	BSURFA	34
UPBAVE = (UPB(1,JP1) + UPB(IP1,J))*0.5	BSURFA	35
C* PHITD = PHIBTD(1,J)	BSURFA	36
RHOTD = (F5+RHUAVE - DUX*E4 - TWOTDR*D1	BSURFA	37
3 + DELT*DOTMIG(1)/DVAXIS + CUTMP(1,J)*AREAC(1))/	BSURFA	38
3 (PHITD*AREAC(1))	BSURFA	39
PHIRHO = PHITD*RHOTD*AREAC(1)	BSURFA	40
C UTOT = 0.0	BSURFA	41
VTOT = 0.0	BSURFA	42
C* C****IF (PHIAVE .LT. 0.999) CALL DRAG(XDRAG(1,J),.FALSE.)	BSURFA	43
	BSURFA	44
	BSURFA	45
	BSURFA	46
	BSURFA	47
	BSURFA	48
	BSURFA	49

C*	HIGN = HBG(I,1)	BSURFA	50
	IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)	BSURFA	51
C		BSURFA	52
	HBOUD(I,J) = GAM*(F5*(C1+B1+C4+H4)/(DENOM*GAM)	BSURFA	53
	\$ - UTOX*C4*UBG(IP1,J) - TWOTOK*U1*HBG(I,JP1)	BSURFA	54
	\$ + DELT*(DOTMIG(I)/DVAXIS*PIGN - QBAG(I,J))	BSURFA	55
	\$ + DOTMB(I,J)*HMB*AKEAC(I))/PI*RHO	BSURFA	56
C*		BSURFA	57
C		BSURFA	58
	RHOBUD(I,J) = RHOTDI	BSURFA	59
	UBGUD(I,J) = UTCT	BSURFA	60
	VBGUD(I,J) = VTCT	BSURFA	61
	RETURN	BSURFA	62
	END	BSURFA	63
		BSURFA	64

	SUBROUTINE BSURFT	BSURFT	2
C	1X IS 1 HERE, 1K IS J	BSURFT	3
	COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	BSURFT	4
	3 UTDT,VHCAVE,VTDT	BSURFT	5
	COMMON/CHAM/1,J,XB,KB,NGX,NGR,IBLOC,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
	3 AREACH,AREAL(20),IGNIT,ONED,DIAM1,DIAM2,CIS1,UIS2,UIS3,UIS4,	CHAM1	3
	3 AREAK(60),AREARAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),UAVG,	CHAM1	4
	3 AREAH2,DIAMB1,BELBEG,BELBEG,IP1,IP2,RADES,BPIGN	CHAM1	5
	COMMON/CLUCK/TIME,DELT	BSURFT	7
	COMMON/EQNS/DIOX,I2DR,I2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
	3 UX,UR,NX,GU,IWODT,HBP	EQNS	3
	COMMON/GASCON/RG,RRG,CV0,CV	BSURFT	9
	COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
	1 VBG(60,5),UPB(60,5),PCH(60,5),TZC(60,5),	BAG	3
	2 DUTMIG(60),GBAG(60,5),XDRAG(60,5),DOTMB(60,5),UPBDT(60,5),	BAG	4
	3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
	4 VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIRP(60,5),	BAG	6
	5 PHIPTD(60,5),TZP(60),TBP(60,5)	BAG	7
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C	RH=RH+UR	BSURFT	12
C	CALL GSPROP(RG,RRG,R,CV0,CVH,CV,PCH,I,J),HBE(I,J),TDUM,	BSURFT	13
	3 RHOBG(I,J),UBG(I,J),VBG(I,J),UBA,CP,2)	BSURFT	14
	BUGGER = (GAP - 1.0)/TWOGJ	BSURFT	15
	IP1 = 1+1	BSURFT	16
	JP1 = J+1	BSURFT	17
	F1 = PHIBG(I,JP1)	BSURFT	18
	F4 = PHIBG(IP1,J)	BSURFT	19
	F5 = PHIBG(I,J)	BSURFT	20
	G1 = F1*RHOBG(I,JP1)	BSURFT	21
	G4 = F4*RHOBG(IP1,J)	BSURFT	22
	E4 = G4*UBG(IP1,J)	BSURFT	23
	U1 = G1*VBG(I,JP1)	BSURFT	24
	C1 = G1*HBG(I,JP1)	BSURFT	25
	C4 = G4*HBG(IP1,J)	BSURFT	26
	B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1	BSURFT	27
	B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2)*G4	BSURFT	28
	DENOM = F1 + F4	BSURFT	29
	PHIAVE = DENOM*0.5	BSURFT	30
	RHOAVE = (G1+G4)/DENOM	BSURFT	31
	UBGAVE = 0.0	BSURFT	32
	VBGAVE = 0.0	BSURFT	33
	UPBAVE = (UPB(I,JP1) + UPB(IP1,J))*0.5	BSURFT	34
C*	PHITDT = PHIBTD(I,J)	BSURFT	35
	RHOTDT = (F5*RHOAVE - DIOX*E4 - C1LR*D1	BSURFT	36
	3 - DELT*DUTMIG(I)/DVAXIT + CUPC(I,J))/PHITDT	BSURFT	37
	PHIRHO = PHITDT*RHOTDT	BSURFT	38
	UTDT = 0.0	BSURFT	39
	VTDT = 0.0	BSURFT	40
C*		BSURFT	41
C*	IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)	BSURFT	42
C*		BSURFT	43
	HIGN = HBG(I,1)	BSURFT	44
	IF(DUTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)	BSURFT	45
C		BSURFT	46
		BSURFT	47
		BSURFT	48
		BSURFT	49

HBGTD(I,J) = GAM*(F5*(C1+B1+C4+B4)/(LENOM*EAM)	HSURFT	50
3 - UTDX*(C4*UBG(IP1,J) - UTDH*(C1*VBG(I,JP1)	HSURFT	51
3 - DELT*(DOTMIG(I)/DVAXIT*H16A + QBAG(I,J))	HSURFT	52
3 + DOTMB(I,J)*HMB)/PHIKHO	HSURFT	53
	HSURFT	54
	HSURFT	55
KHUBTD(I,J) = RHUTDT	HSURFT	56
UBGTD(I,J) = UTDT	HSURFT	57
VBGTD(I,J) = VTDT	HSURFT	58
RETURN	HSURFT	59
END	HSURFT	60

SUBROUTINE BSURFI	BSURFI	2
COMMON/AVGDT/RHODT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	BSURFI	3
3 UTOT,VBGAVE,VTOT	BSURFI	4
COMMON/CHAM/I,J,XB,RB,NGX,NGR,IBLGE,IENDG,IPATH(60,5),AREAG(5),	CHAMI	2
3 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMI	3
3 AREAR(60),ARELAA,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMI	4
3 ARELAH2,DIAPH1,BELEND,BLLBEG,IPS1,IPS2,RADFS,BPIGN	CHAMI	5
COMMON/CLOCK/TIME,UELI	BSURFI	6
COMMON/LQNS/DILX,I2DR,I2UX,TWOTDR,DILN,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
1 UX,UR,NX,GU,TWOT,HEP	EQNS	3
COMMON/GASCON/RG,RKO,CV0,CVH	BSURFI	8
COMMON/DAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),I2C(60,5),	BAG	3
2 LUMIG(60),GDAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBOT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60,5),CUTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPTD(60,5),TZK(60,5),TBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
DATA GRAV/32.16/	BSURFI	11
L	BSURFI	12
RB=RB+UR	BSURFI	13
L*	BSURFI	14
CALL GSPROP(RG,RKO,R,CV0,CVH,CV,PC,1,J),HBE(I,J),TDUM,	BSURFI	15
3 RHOBG(I,J),UBG(I,J),VBG(I,J),DAG,CP,2)	BSURFI	16
BUGGER = (GAM - 1.0)/TWOGJ	BSURFI	17
IP1 = 1+1	BSURFI	18
JP1 = J+1	BSURFI	19
JM1 = J-1	BSURFI	20
F1 = PHIBG(I,JP1)	BSURFI	21
F3 = PHIBG(I,JM1)	BSURFI	22
F4 = PHIBG(I-1,J)	BSURFI	23
F5 = PHIBG(I,J)	BSURFI	24
G1 = F1*RHOBG(I,JP1)	BSURFI	25
G3 = F3*RHOBG(I,JM1)	BSURFI	26
G4 = F4*RHOBG(IP1,J)	BSURFI	27
L4 = G4*UBG(IP1,J)	BSURFI	28
D1 = G1*VBG(I,JP1)	BSURFI	29
D3 = G3*VBG(I,JM1)	BSURFI	30
D4 = G4*VBG(IP1,J)	BSURFI	31
G1 = G1*HBG(I,JP1)	BSURFI	32
G3 = G3*HBG(I,JM1)	BSURFI	33
G4 = G4*HBG(IP1,J)	BSURFI	34
G1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1	BSURFI	35
G3 = BUGGER*(UBG(I,JM1)**2 + VBG(I,JM1)**2)*G3	BSURFI	36
G4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2)*G4	BSURFI	37
URP = (RB + UR)/RB	BSURFI	38
URM = (RB - UR)/RB	BSURFI	39
C*	BSURFI	40
DENUM = F1 + F3 + F4 + F4	BSURFI	41
SIG6 = G1 + G3 + G4 + G4	BSURFI	42
SIGU = D1 + D3 + D4 + D4	BSURFI	43
PHIAVE = DENUM*0.25	BSURFI	44
RHCAVE = SIG6/DENUM	BSURFI	45
UBCAVE = 0.0	BSURFI	46
VBGAVE = SIGU/SIG6	BSURFI	47
UPBAVE = (UPB(I,JP1) + UPB(I,JM1) + 2.0*UPB(IP1,J))*0.25	BSURFI	48
C*	BSURFI	49

PHITDI = PHIBTD(I,J)	BSURFI	50
RHOTDI = (F5*KHOAVE - DTDX*E4 - T2DR*(URP*C1 - URM*D3)	BSURFI	51
3 + DDTMB(I,J))/PHITDI	BSURFI	52
PHIRHO = PHITDI*RHOTDI	BSURFI	53
C*	BSURFI	54
UTDI = 0.0	BSURFI	55
VTDI = (F5*SIGD/DENOM - UTDX*E4*VBG(I,I,J)	BSURFI	56
3 - T2DR*(DRP*(U1*VBG(I,JP1) + GRAV *PCH(I,JP1))	BSURFI	57
3 - DRM*(D3*VBG(I,JP1) + GRAV *PCH(I,JM1)))	BSURFI	58
3 + DELT*GRAV*PCH(I,J)/FB) / PHIRHO	BSURFI	59
IF(ABS(VTDI) .LT. 0.1) VTDI = 0.0	BSURFI	60
C*	BSURFI	61
C ARRAY XDRAG IS CLEARED IN MAIN AT THE BEGINNING OF EACH TIME	BSURFI	62
C INTERVAL SO IF SUBROUTINE DRAG IS NOT CALLED, XDRAG(I,J) IS 0.0.	BSURFI	63
C****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)	BSURFI	64
C*	BSURFI	65
MBGTD(I,J) = GAM*(F5*(C1+B1+C3+B3+C4+C4+B4+B4)/(DENOM*GAM)	BSURFI	66
3 - UTDX*C4*UBG(I,I,J)	BSURFI	67
3 - T2DR*(URP*C1*VBG(I,JP1) - DRP*C3*VBG(I,JM1))	BSURFI	68
3 - DELT*UBG(I,J) + DDTMB(I,J)*AMB	BSURFI	69
3 - BUGGER*VTDI*VTDI*PHIRHC/GAM)/PHIRHO	BSURFI	70
C*	BSURFI	71
C	BSURFI	72
RHUBTD(I,J) = RHOTDI	BSURFI	73
UBGTD(I,J) = UTDI	BSURFI	74
VBGTD(I,J) = VTDI	BSURFI	75
RETURN	BSURFI	76
END	BSURFI	77

SUBROUTINE BSURFB	BSURFB	2
COMMON/AVGDI/RHCTD1,PHIRHO,PHIAVE,PHUAVE,UBGAVE,UPHAVE,	BSURFB	3
3 UTDT,VBGAVE,VTDT	BSURFB	4
COMMON/CHAM/1,0,XB,RB,NGX,NGK,IELGB,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
3 AREACH,AREAL(60),IGNIT,ONED,DIAP1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3 AKLAR(60),AKEAAX,CHAM1,CHAM2,CHANG,OPGAP,AREAGP(60),DAYG,	CHAM1	4
3 AREAH2,DIAPBT,BELEN2,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLUCK/TIME,DELT	BSURFB	6
COMMON/EQNS/CLUX,TZUR,TZUX,TWOTUR,CLN,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
3 UX,UR,NX,00,TWOD1,HBP	EQNS	3
COMMON/GASCO/RK0,RK0,CV0,CVH	BSURFB	8
COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
1 VBG(60,5), UPB(60,5), PCH(60,5), T2C(60,5),	BAG	3
2 DUMHAG(60), GBA9(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
3 PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
4 VBGTD(60,5), THG(60,5), DOTMBG(60), CLTMP(60,5), PHIBP(60,5),	BAG	6
5 PHIBTD(60,5), TZR(60), IBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHANG,BPIGN	CHAMLOG	2
C	BSURFB	11
RH=RH0+UR	BSURFB	12
C*	BSURFB	13
CALL GSPROP(RH,RK0,R,CV0,CVH,CV,PCH,1,J),HBE(1,J),TDUM,	BSURFB	14
3 RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	BSURFB	15
BUGGLR = (GAP - 1.0)/TWOGJ	BSURFB	16
IP1 = 1+1	BSURFB	17
JM1 = J-1	BSURFB	18
F3 = PHIBG(1,JM1)	BSURFB	19
F4 = PHIBG(IP1,J)	BSURFB	20
F5 = PHIBG(1,J)	BSURFB	21
G3 = F3*RHOBG(1,JM1)	BSURFB	22
G4 = F4*RHOBG(IP1,J)	BSURFB	23
L4 = G4*UBG(IP1,J)	BSURFB	24
U3 = G3*VBG(1,JM1)	BSURFB	25
C3 = G3*HBG(1,JM1)	BSURFB	26
L4 = G4*HBG(IP1,J)	BSURFB	27
B3 = BUGGLR*VBG(1,JM1)**2*G3	BSURFB	28
B4 = BUGGLR*(UBG(IP1,J)**2 + VBG(IP1,J)**2)*G4	BSURFB	29
C*	BSURFB	30
DENUM = F3 + F4	BSURFB	31
PHIAVE = DENUM*0.5	BSURFB	32
RHUAVE = (G3+G4)/DENUM	BSURFB	33
UBGAVE = 0.0	BSURFB	34
UPHAVE = (UPB(1,JM1) + UPB(IP1,J))*0.5	BSURFB	35
VBGAVE = 0.0	BSURFB	36
C*	BSURFB	37
PHIBTD = PHIBTD(1,J)	BSURFB	38
RHIBTD = (F5*RHUAVE - DTDX*E4 + CLUX*U3	BSURFB	39
3 + DOTMB(1,J))/PHIBTD	BSURFB	40
PHIRHO = PHIBTD*RHIBTD	BSURFB	41
UTDT = 0.0	BSURFB	42
VTDT = 0.0	BSURFB	43
C*	BSURFB	44
C****IF(PHIAVE .LT. 0.999) CALL DRAG(XCRAG(1,J),.FALSE.)	BSURFB	45
C*	BSURFB	46
HBGTD(1,J) = GAM*(F5*(B3+C3+B4+C4)/(DENOM*GAM)	BSURFB	47
3 - DTDX*C4*UBG(IP1,J) + DTDZ*L3*HBG(1,JM1)	BSURFB	48
3 - DELT*GBAG(1,J) + DOTMB(1,J,*HMB))/PHIRHO	BSURFB	49

C*		BSURFB	50
C		BSURFB	51
	RHBTU(1,J) = RHOTU	BSURFB	52
	UBTU(1,J) = UTU	BSURFB	53
	VBTU(1,J) = VTU	BSURFB	54
	RETURN	BSURFB	55
	END	BSURFB	56

SUBROUTINE FSURFA	FSURFA	2
COMMON/AVGDI/RHOTDI,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,	FSURFA	3
3 UDTI,VBGAVL,VTCT	FSURFA	4
COMMON/CHAM/I,J,XB,KD,NGX,NGR,IELGE,IENDB,IPATH(60,5),AREAG(5),	CHAMI	2
3 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMI	3
3 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),DAVG,	CHAMI	4
3 AREAH2,DIAPHI,BELENB,BELLEG,IPS1,IPS2,RAOFS,BPIGN	CHAMI	5
COMMON/CLUCK/IML,DELT	FSURFA	6
COMMON/EQNS/LTDX,I2CR,I2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXI1,	EQNS	2
3 UX,UX,NX,GU,TWOUT,HBP	EQNS	3
COMMON/GASCON/KO,KRO,CVO,CVH	FSURFA	8
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),I2C(60,5),	BAG	3
2 DUTHIG(60),GBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTHBG(60,5),DOTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPTD(60,5),TZR(60,5),TBP(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C	FSURFA	11
KB=0.0	FSURFA	12
KB=KB+UX	FSURFA	13
C	FSURFA	14
CALL GSPROP(KO,KRO,K,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,	FSURFA	15
3 RHOBG(I,J),UBG(I,J),VBG(I,J),GAP,CP,2)	FSURFA	16
BUGGER = (GAP - 1.0)/TWOGJ	FSURFA	17
IM1 = I - 1	FSURFA	18
JP1 = J + 1	FSURFA	19
F1 = PHIBG(I,JP1)	FSURFA	20
F2 = PHIBG(IM1,J)*AREAC(IM1)	FSURFA	21
F5 = PHIBG(I,J)*AREAC(I)	FSURFA	22
G1 = F1*RHOBG(I,JP1)	FSURFA	23
G2 = F2*RHOBG(IM1,J)	FSURFA	24
L2 = G2*UBG(IM1,J)	FSURFA	25
U1 = G1*VBG(I,JP1)	FSURFA	26
L1 = G1*HBG(I,JP1)	FSURFA	27
G2 = G2*HBG(IM1,J)	FSURFA	28
U1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1	FSURFA	29
U2 = BUGGER*UBG(IM1,J)**2*G2	FSURFA	30
C*	FSURFA	31
DENOM = F1 + F2	FSURFA	32
PHIAVE = DENOM/(AREAC(I) + AREAC(IM1))	FSURFA	33
RHOAVE = (G1+G2)/DENOM	FSURFA	34
UBGAVE = 0.0	FSURFA	35
VBGAVE = 0.0	FSURFA	36
UPBAVE = (UPB(I,JP1) + UPB(IM1,J))*0.5	FSURFA	37
C*	FSURFA	38
PHITDI = PHITD(I,J)	FSURFA	39
RHOTDI = (F5*RHOAVE + DIDX*E2 - TWOTDR*D1	FSURFA	40
3 + DELT*DUTHIG(I)/DVAXIS + CUIP(I,J)*AREAC(I)) /	FSURFA	41
3 (PHITDI*AREAC(I))	FSURFA	42
PHIRHO = PHITDI*RHOTDI*AREAC(I)	FSURFA	43
UDTI = 0.0	FSURFA	44
VTCT = 0.0	FSURFA	45
C*	FSURFA	46
C****IF (PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)	FSURFA	47
C*	FSURFA	48
HIGN = HBG(I,1)	FSURFA	49

C	IF(DOTMIG(I).GT. 0.00001) HIGN = HBG(I,2)	FSURFA	50
	HBOID(I,J) = GAM*(15*(B1+C1+B2+C2)/(LENGH*GAM)	FSURFA	51
	+ PTDA*C2*UBG(IH1,J) - TWQIDH*L1*HBG(I,JP1)	FSURFA	52
	+ DEL1*(DOTMIG(I)/DVAXIS*HIGN - QBAG(I,J))	FSURFA	53
	+ DOTPB(I,J)*HMB*AREAC(I))/PHIRHO	FSURFA	54
C*		FSURFA	55
C		FSURFA	56
	RHOBID(I,J) = RHOTDT	FSURFA	57
	UBOID(I,J) = UTOT	FSURFA	58
	VBOID(I,J) = VTOT	FSURFA	59
	RETURN	FSURFA	60
	END	FSURFA	61
		FSURFA	62

SUBROUTINE FSURF1	FSURFT	2
COMMON/AVGDI/RHOTDI,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,	FSURFT	3
3 UTOT,VBGAVE,VTOT	FSURFT	4
COMMON/CHAM/1,J,XB,RB,NGX,NGR,IBEGE,IENDB,IPATH(60,5),AREAG(5),	CHAMI	2
2 AREACH,AREAL(60),IGNIT,UNED,DIAM1,LIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMI	3
3 AREAK(60),AREAGX,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAMI	4
4 AREAL2,DIAM1,DELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAMI	5
COMMON/CLOCK/TIME,DELT	FSURFT	6
COMMON/EQNS/DTDX,T2CR,T2DA,TWOIDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
5 DX,DX,NX,GJ,TWOIT,HBP	EQNS	3
COMMON/GASCON/R0,RK0,CV0,CVH	FSURFT	8
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),T2C(60,5),	BAG	3
2 DDTMIG(60),GDBG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBDT(60,5),RHODT(60,5),HBGDT(60,5),UBGDT(60,5),	BAG	5
4 VBGDT(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIRP(60,5),	BAG	6
5 PHIPDT(60,5),T2R(60),TBP(60,5)	BAG	7
LOGICAL IGNIT,UNED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C*	FSURFT	11
RH=RB+UK	FSURFT	12
C	FSURFT	13
CALL GSPROP(R0,RK0,R,CV0,CVH,CV,PCH(1,J),HBG(1,J),TDUM,	FSURFT	14
1 RHOBG(1,J),UBG(1,J),VBG(1,J),GAR,CP,2)	FSURFT	15
BUGGLR = (GAR - 1.0)/TWOGJ	FSURFT	16
IM1 = 1 - 1	FSURFT	17
JP1 = J + 1	FSURFT	18
F1 = PHIBG(1,JP1)	FSURFT	19
F2 = PHIBG(IM1,J)	FSURFT	20
F5 = PHIBG(1,J)	FSURFT	21
G1 = F1*RHOBU(1,JP1)	FSURFT	22
G2 = F2*RHOBU(IM1,J)	FSURFT	23
G2 = G2*UBG(IM1,J)	FSURFT	24
G1 = G1*VBG(1,JP1)	FSURFT	25
G1 = G1*HBG(1,JP1)	FSURFT	26
G2 = G2*HBG(IM1,J)	FSURFT	27
B1 = BUGGLR*(UBG(1,JP1)**2 + VBG(1,JP1)**2)*G1	FSURFT	28
B2 = BUGGLR*(UBG(IM1,J)**2 + VBG(IM1,J)**2)*G2	FSURFT	29
C*	FSURFT	30
DENOM = F1 + F2	FSURFT	31
PHIAVE = DENOM*0.5	FSURFT	32
RHOAVE = (G1+G2)/DENOM	FSURFT	33
UBGAVE = 0.0	FSURFT	34
VBGAVE = 0.0	FSURFT	35
UPBAVE = (UPB(1,JP1) + UPB(IM1,J))*0.5	FSURFT	36
C	FSURFT	37
PHITDI = PHIBDT(1,J)	FSURFT	38
RHODI = (F5*RHOAVE + DTDX*E2 - CTDR*01	FSURFT	39
5 - DELT*DDTMIG(1)/DVAXIT + DDIMP(1,J))/PHITDI	FSURFT	40
PHIRHO = PHITDI*RHODI	FSURFT	41
UTOT = 0.0	FSURFT	42
VTOT = 0.0	FSURFT	43
C	FSURFT	44
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(1,J),.FALSE.)	FSURFT	45
C*	FSURFT	46
HIGN = HBG(1,1)	FSURFT	47
IF(DDTMIG(1) .GT. 0.00001) HIGN = HBG(1,2)	FSURFT	48
C	FSURFT	49

UBG1U(I,J) = GAM*(F5*(B1+C1+B2+C2)/(LENOM*GAM)	FSURFT	50
* + U10X*L2*UBG(IM1,J) - UTDH*L1*VBG(I,JP1)	FSURFT	51
* - DELT*(DUTMIG(I)/DVAXIT*FIGN + QBAG(I,J))	FSURFT	52
* + DOTAB(I,J)*HMB)/PHIRHO	FSUREI	53
	FSURFT	54
	FSUREI	55
RHOUTU(I,J) = RHOTDT	FSURFT	56
UBG1U(I,J) = UTDT	FSUREI	57
VBG1U(I,J) = VTDT	FSURFT	58
RETURN	FSURFT	59
END	FSURFT	60

SUBROUTINE FSURFI	FSURFI	2
COMMON/AVGDI/RHODI,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	FSURFI	3
1 ULI,VBGAUL,VIDI	FSURFI	4
COMMON/CHAM/1,J,XI,RB,NGX,NGR,IBEGE,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
2 AREACH,ARELAC(60),IGNI1,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,10PGAP,AREAGP(60),UAVG,	CHAM1	4
4 AREAL,DIAM1,BELBEG,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLKCK/TIME,DELT	FSURFI	6
COMMON/EQNS/LTDX,T2DR,T2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
1 UA,UR,AX,GU,TWOUT,HEP	EQNS	3
COMMON/GASCOI/KO,KRO,CVO,CVH	FSURFI	8
COMMON/LAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCB(60,5),TZC(60,5),	BAG	3
2 DUTMIG(60),VBAG(60,5),XURAG(60,5),DUTPB(60,5),UPBOT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DUTMBG(60,5),DUTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIBTD(60,5),TZR(60,5),TBP(60,5)	BAG	7
LOGICAL IGNI1,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
DATA GRAY/52.16/	FSURFI	11
C*	FSURFI	12
RB=RB+DR	FSURFI	13
C	FSURFI	14
CALL GSPROP(KO,KRO,K,CVO,CVH,CV,PCB(1,J),HBG(1,J),TDUM,	FSURFI	15
1 RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	FSURFI	16
BUGGER = (GAP - 1.0)/TWOGJ	FSURFI	17
IM1 = 1 - 1	FSURFI	18
JP1 = J + 1	FSURFI	19
JM1 = J - 1	FSURFI	20
F1 = PHIBG(1,JP1)	FSURFI	21
F2 = PHIBG(IM1,J)	FSURFI	22
F3 = PHIBG(1,JM1)	FSURFI	23
F5 = PHIBG(1,J)	FSURFI	24
G1 = F1 * RHOBG(1,JP1)	FSURFI	25
G2 = F2 * RHOBG(IM1,J)	FSURFI	26
G3 = F3 * RHOBG(1,JM1)	FSURFI	27
L2 = G2 * UBG(IM1,J)	FSURFI	28
U1 = G1 * VBG(1,JP1)	FSURFI	29
U2 = G2 * VBG(IM1,J)	FSURFI	30
U3 = G3 * VBG(1,JM1)	FSURFI	31
C1 = G1 * HBG(1,JP1)	FSURFI	32
C2 = G2 * HBG(IM1,J)	FSURFI	33
C3 = G3 * HBG(1,JM1)	FSURFI	34
U1 = BUGGER * (UBG(1,JP1)**2 + VBG(1,JP1)**2) * G1	FSURFI	35
U2 = BUGGER * (UBG(IM1,J)**2 + VBG(IM1,J)**2) * G2	FSURFI	36
U3 = BUGGER * (UBG(1,JM1)**2 + VBG(1,JM1)**2) * G3	FSURFI	37
UMP = (RB + DR)/RB	FSURFI	38
URM = (RB - DR)/RB	FSURFI	39
C*	FSURFI	40
DENOM = F1 + F2 + F2 + F3	FSURFI	41
SIGB = G1 + G2 + G2 + G3	FSURFI	42
SIGU = U1 + L2 + U2 + U3	FSURFI	43
PHIAVE = DENOM * 0.25	FSURFI	44
RHUAVE = SIGB/DENOM	FSURFI	45
UBGAUL = 0.0	FSURFI	46
VBGAUL = SIGU/SIGB	FSURFI	47
UPBAVE = (UPB(1,JP1) + 2.0 * UPB(IM1,J) + UPB(1,JM1)) * 0.25	FSURFI	48
C*	FSURFI	49

PHITDT = PHIBTD(I,J)	FSURFI	50
RHOTDT = (F5*RHOAVE + DTDX*E2	FSURFI	51
- T2DX*(DRP*U1 - DRM*D3)	FSURFI	52
+ DOTMB(I,J))/PHITDT	FSURFI	53
PHIRHO = PHITDT*RHOTDT	FSURFI	54
C*	FSURFI	55
UTDT = 0.0	FSURFI	56
VTDT = (F5*SIGD/DENOM + DTDX*E2*VBG(I,M1,J)	FSURFI	57
- T2DX*(DRP*(D1*VBG(I,JP1) + GRAV *PCH(I,JP1))	FSURFI	58
- DRM*(D3*VBG(I,JP1) + GRAV *PCH(I,JM1)))	FSURFI	59
+ DELT*GRAV*PCH(I,J)/RB)/PHIRHO	FSURFI	60
IF(ABS(VTDT) .LT. 0.1) VTDT = 0.0	FSURFI	61
C*	FSURFI	62
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)	FSURFI	63
VBGTD(I,J) = GAM*(F5*(B1+C1+B2+C2+B2+C2+B3+C3)/(DENOM*GAM)	FSURFI	64
+ DTDX*C2*VBG(I,M1,J)	FSURFI	65
- T2DX*(DRP*C1*VBG(I,JP1) - DRM*C3*VBG(I,JM1))	FSURFI	66
- DELT*DRAG(I,J) + DOTMB(I,J)*RMB	FSURFI	67
- BUGGER*VTDT*VTDT*PHIRHC/GAM)/PHIRHO	FSURFI	68
C*	FSURFI	69
C	FSURFI	70
RHOTDT(I,J) = RHOTDT	FSURFI	71
VBGTD(I,J) = UTDT	FSURFI	72
VBGTD(I,J) = VTDT	FSURFI	73
RETURN	FSURFI	74
END	FSURFI	75

SUBROUTINE FSURFB	FSURFB	2
COMMON/AVGDT/RHODT,PHIRHO,PHIAVE,RHUAVE,UBGAVE,UPBAVE,	FSURFB	3
1 UTOT,VBGAVE,VTOT	FSURFB	4
COMMON/CHAM/1,J,XB,RB,NGX,NGK,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAM1	2
1 AREACH,AREAL(60),IGNIT,UNED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
1 AREAL(60),AREALX,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAM1	4
1 AREAL2,DIAM1,DIAM2,BLEND,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLUCK/TIME,DELT	FSURFB	6
COMMON/LGNS/DIOX,T2OK,T2OTR,DIOX,HMB,TWOGJ,DVAXIS,DVAXIT,	LGNS	2
1 DIOX,NGX,GU,TWOTR,HBP	LGNS	3
COMMON/GASCON/RG,RRG,CV0,CVH	FSURFB	6
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),L2C(60,5),	BAG	3
2 DQIMIG(60),QBAG(60,5),XDRAG(60,5),DOTMB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60),CLTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIBTL(60,5),TZK(60),TBP(60,5)	BAG	7
LOGICAL IGNIT,UNED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C* RH=RG+UR	FSURFB	11
C* CALL USPRDP(RG,RRG,R,CV0,CVH,CV,PCH(1,J),HBG(1,J),TDUM,	FSURFB	13
1 RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	FSURFB	14
BUGGER = (GAP - 1.0)/TWOGJ	FSURFB	15
IM1 = 1 - 1	FSURFB	16
JM1 = J - 1	FSURFB	17
F2 = PHIBG(IM1,J)	FSURFB	18
F3 = PHIBG(1,JM1)	FSURFB	19
F5 = PHIBG(1,J)	FSURFB	20
G2 = F2*RHOBG(IM1,J)	FSURFB	21
G3 = F3*RHOBG(1,JM1)	FSURFB	22
E2 = G2*UBG(IM1,J)	FSURFB	23
J3 = G3*VBG(1,JM1)	FSURFB	24
C2 = G2*HBG(IM1,J)	FSURFB	25
C3 = G3*HBG(1,JM1)	FSURFB	26
B2 = BUGGER*UBG(IM1,J)**2*G2	FSURFB	27
B3 = BUGGER*(UBG(1,JM1)**2 + VBG(1,JM1)**2)*G3	FSURFB	28
C* DENOM = F2 + F3	FSURFB	30
PHIAVE = DENOM*0.5	FSURFB	31
RHUAVE = (G2+G3)/DENOM	FSURFB	32
UBGAVE = 0.0	FSURFB	33
VBGAVE = 0.0	FSURFB	34
UPBAVE = (UPB(IM1,J) + UPB(1,JM1))*0.5	FSURFB	35
C* PHITDT = PHIBTD(1,J)	FSURFB	36
RHUTDT = (F5*RHUAVE + DIOX*E2 + DIOX*G3	FSURFB	37
+ DOTMB(1,J))/PHITDT	FSURFB	38
PHIRHO = PHITDT*RHUTDT	FSURFB	39
UTOT = 0.0	FSURFB	40
VTOT = 0.0	FSURFB	41
C *****IF(PHIAVE.LT. 0.999) CALL DRAG(XCRAG(I,J),.FALSE.)	FSURFB	42
HBGTD(1,J) = GAP*(F5*(B2+C2+B3+C3)/(DENOM*GAM)	FSURFB	43
+ DIOX*E2*HBG(IM1,J) + DIOX*G3*HBG(1,JM1)	FSURFB	44
- DELT*QBAG(1,J) + DOTMB(1,J)*MB)/PHIRHO	FSURFB	45
C*	FSURFB	46

C	RHOUTD(1,J) = RHOUTD	FSURFB	50
	OROUTD(1,J) = OUTD	FSURFB	51
	VROUTD(1,J) = VOUT	FSURFB	52
	RETURN	FSURFB	53
	END	FSURFB	54
		FSURFB	55

SUBROUTINE DIMIN		DIMIN	2
C	SUBROUTINE DIMIN IS THE BARREL ROUTINE EQUIVALENT TO REGRESS IN	DIMIN	3
C	THE CHAMBER, WE ASSUME ONLY IGNITED PROPELLANT IS IN THE BARREL.	DIMIN	4
C		DIMIN	5
	COMMON/BURN/ATP,CT,PEXP	DIMIN	6
	COMMON/EGNS/LTDX,T2DR,T2UX,TWOTDR,DIER,HMB,TWOGJ,DVAXIS,DVAXIT,	DIMIN	7
	3 DX,DR,NX,GJ,TWOT,HEP	EGNS	2
	COMMON/GRAIN/ XL(60,5), DU(60,5), DI(60,5), FN,	EGNS	3
	1 XLDT(60,5), DUTDT(60,5), DUTOT(60,5), XL0, D00, D10,	GRAIN	2
	2 ALB(100), DOB(100), DIB(100), UALB(100), LOOB(100), UDIB(100)	GRAIN	3
	COMMON/INPUTS/C1,C2,C3,C4,T0,T1CN,UCCNS,RHOF,PHIU,TF,CA,RH00,	GRAIN	4
	3 RD,PG,DD,GIRHOP,HW,DM,TIGNBP,QBCONS,TOTM	INPUTS	2
	COMMON/SPLINT/WHOLEC,WHOLEB	INPUTS	3
	COMMON/BARREL/ PHI(100), RHOG(100), PG(100), UG(100), UP(100),	DIMIN	11
	1 PG(100), TG(100), FRODT(100), GL(100), UCRA6(100), FRIC(100),	BARREL	2
	2 UCONV(100), OUP(100), UPHI(100), URHOG(100), UMG(100), UUG(100),	BARREL	3
	3 AMASS(100), AMOM(100), AENER(100), LAMASS(100), LAMOM(100),	BARREL	4
	4 UAENER(100)	BARREL	5
	LOGICAL WHOLEC,WHOLEB	BARREL	6
	DATA PIUF/.785398/	DIMIN	13
C		DIMIN	14
C	SUBROUTINE DIMIN CALCULATES UPHI FOR USE IN RH00H AND IN PROPMO.	DIMIN	15
C	IT UPDATES GRAIN DIMENSIONS FOR USE IN PROPMC.	DIMIN	16
C		DIMIN	17
C	ARRAY PHOUT IS CLEARED IN MAIN AT EACH TIME INTERVAL.	DIMIN	18
C		DIMIN	19
	DO 50 I=2,NX	DIMIN	20
	IF (PHI(I).GE.0.99999) GO TO 45	DIMIN	21
C		DIMIN	22
	R = ATP*PG(I)**PEXP + CT	DIMIN	23
	BURNL = R*TWOT	DIMIN	24
	UALB(I) = ALB(I) - BURNL	DIMIN	25
C		DIMIN	26
	IF (DOB(I) .LE. 3.0*DIB(I)) GO TO 20	DIMIN	27
C		DIMIN	28
C	CODING FOR CYLINDRICAL PROPELLANT GRAINS	DIMIN	29
	UOGB(I) = DOB(I) - BURNL	DIMIN	30
	UDIB(I) = DIB(I) + BURNL	DIMIN	31
	VOLL = PIUF*UALB(I)*(DOB(I)*DCB(I) - FN*DIB(I)*DIB(I))	DIMIN	32
	VNEW = PIUF*UALB(I)*(UOGB(I)*DCB(I) - FN*UDIB(I)*UDIB(I))	DIMIN	33
	GO TO 30	DIMIN	34
		DIMIN	35
C		DIMIN	36
C		DIMIN	37
C	CODING FOR SPLINTERED PROPELLANT GRAINS	DIMIN	38
20	CONTINUE	DIMIN	39
	IF (DOB(I) .LE. DIB(I)) GO TO 25	DIMIN	40
	WHOLEB = .FALSE.	DIMIN	41
	AREA = PIUF*(DOB(I)*DOB(I) - FN*UALB(I)*CIB(I))	DIMIN	42
	DIB(I) = 3.14*(DOB(I) + FN*DIB(I))	DIMIN	43
	DOB(I) = AREA	DIMIN	44
C		DIMIN	45
20	DELR = BURNL*0.5	DIMIN	46
	UOGB(I) = DOB(I) - DIB(I)*DELR	DIMIN	47
	IF (UOGB(I) .GE. 1.0E-7) GO TO 2/	DIMIN	48
	UOGB(I) = 0.0	DIMIN	49
		DIMIN	50

	UDIB(1) = 0.0	DIMIN	51
	UXLB(1) = 0.0	DIMIN	52
	UPHI(1) = 1.0	DIMIN	53
	GO TO 50	DIMIN	54
C		DIMIN	55
21	CONTINUE	DIMIN	56
	UDIB(1) = SQRT(DIB(1)*DIB(1)/DOB(1)*UDCB(1))	DIMIN	57
	VOLD = XLB(1)*DOB(1)	DIMIN	58
	VNEW = UXLB(1)*UDOB(1)	DIMIN	59
C		DIMIN	60
C		DIMIN	61
C		DIMIN	62
50	IF (VNEW .LE. 0.0) GO TO 45	DIMIN	63
	DELTAV = VOLD - VNEW	DIMIN	64
	PNDV = (1.0 - PHI(1))/VOLD	DIMIN	65
	TEMP = PNDV*DELTAV	DIMIN	66
	PRDUT(1) = TEMP*RHOP	DIMIN	67
	UPHI(1) = PHI(1) + TEMP	DIMIN	68
	GO TO 50	DIMIN	69
C		DIMIN	70
C		DIMIN	71
45	UPHI(1) = PHI(1)	DIMIN	72
C		DIMIN	73
C		DIMIN	74
50	CONTINUE	DIMIN	75
C		DIMIN	76
	RETURN	DIMIN	77
	END	DIMIN	78
C		DIMIN	79
C		DIMIN	80
C		DIMIN	81
C		DIMIN	82
C		DIMIN	83

<pre> SUBROUTINE RHOQH C C SUBROUTINE RHOQH PERFORMS THE FINITE DIFFERENCE CALCULATIONS AT C THE BARRIL GRIDS. C COMMON/AVGDT/URFOT,PHIRHO,PHIAVE,RH,AVE,UGAVE,UPAVE,OUT, VPGAVE,VTOT COMMON/BARRIL2/DUELA,AP,VP,BORER,PCHER,BORER2,DT2HU,DTUSU,XLBAR COMMON/CLOCK/TIME,DELT COMMON/EQNS/CDX,TDK,TDX,TWOTER,DICN,HMB,TWOGJ,DVAXIS,DVAXIT, DX,UR,NX,GO,TWODI,NBP COMMON/GASCON/RG,RRG,CVO,CVH COMMON/GRIDNX/LXPRIM COMMON/BARRIL/PHI(100),RHOG(100),HG(100),UG(100),UP(100), PG(100),TG(100),PRDUT(100),GL(100),UCRAG(100),FRICT(100), UCONV(100),UUP(100),UPHT(100),CRHOG(100),UHG(100),UUG(100), AMASS(100),AMOM(100),AENER(100),LAMASS(100),UAMOM(100), AENER(100) C DATA GRAV/32.16/ C C NP1 = NX-1 NP1 = NX+1 PHI(NP1) = PHI(NM1) RHOG(NP1) = RHOG(NM1) UG(NP1) = 2.0*VP - UG(NM1) AMASS(NP1) = AMASS(NM1) AMOM(NP1) = AMOM(NM1) AENER(NP1) = AENER(NM1) HG(NP1) = HG(NM1) UP(NP1) = 2*VP - UP(NM1) PG(NP1) = PG(NM1) C C *****RHOQH SHOULD NOT BE CALLED IF NX IS 1. C DO 100 I=2,NX IM1=I-1 IP1=I+1 C CALL GSPROP(RG,RRG,R,CVO,CVH,CV,PG(I),HG(I),TDUM,RHOG(I), UG(I),0.0,GAM,CP,2) BUGGER = (GAM - 1.0)/TWOGJ C WHEN I = NX - 1, GRID I + 1 SHOULD BE THE SAME DISTANCE AWAY AS GRID NX - 2 (GRID NX WILL PROBABLY BE FARTHER AWAY), SO USE LINEAR INTERPOLATION TO GET THE PROPERTIES AT I + 1. LET PHI REMAIN THE SAME. IF (I .NE. NX - 1) GO TO 30 RHUS = RHOG(I,X) UGS = UG(NX) HGS = HG(NX) PGS = PG(NX) UPS = UP(NX) RATIO = DX/UXPRIM RHOG(IP1) = RHOG(I) + RATIO*(RHOG(IP1) - RHOG(I)) UG(IP1) = UG(I) + RATIO*(UG(IP1) - UG(I)) </pre>	<pre> RHOQH 2 RHOQH 3 RHOQH 4 RHOQH 5 RHOQH 6 RHOQH 7 RHOQH 8 RHOQH 9 RHOQH 10 EQNS 2 EQNS 3 RHOQH 12 MOD0301 26 BARRIL 2 BARRIL 3 BARRIL 4 BARRIL 5 BARRIL 6 RHOQH 14 RHOQH 15 RHOQH 16 RHOQH 17 RHOQH 18 RHOQH 19 RHOQH 20 RHOQH 21 RHOQH 22 RHOQH 23 RHOQH 24 RHOQH 25 RHOQH 26 MOD1205 9 RHOQH 28 RHOQH 29 RHOQH 30 RHOQH 31 RHOQH 32 RHOQH 33 RHOQH 34 RHOQH 35 RHOQH 36 RHOQH 37 RHOQH 38 MOD0301 27 MOD0301 28 MOD0301 29 MOD0301 30 MOD0301 31 MOD0301 32 MOD0301 33 MOD0301 34 MOD0301 35 MOD0301 36 MOD0301 37 MOD0301 38 MOD0301 39 MOD0301 40 </pre>
--	--

	HG(IP1) = HG(I) + RATIO*(HG(IP1) - HG(I))	MOD0301	41
	PG(IP1) = PG(I) + RATIO*(PG(IP1) - PG(I))	MOD0301	42
	UP(IP1) = UP(I) + RATIO*(UP(IP1) - UP(I))	MOD0301	43
C	AT GRID NX DXPRIM SHOULD BE USED RATHER THAN DX, SO T2DX MUST BE	MOD0301	44
C	CHANGED.	MOD0301	45
30	CONTINUE	MOD0301	46
	IF(I.NE.NX) GO TO 40	MOD0301	47
	T2DX = T2DX	MOD0301	48
	T2DX = DELT/(2.0*DXPRIM)	MOD0301	49
C		MOD0301	50
40	CONTINUE	MOD0301	51
C		MOD0301	52
	F1 = PHI(IM1)	RHOUH	39
	F2 = PHI(IP1)	RHOUH	40
	F5 = PHI(I)	RHOUH	41
	G1 = RHOG(IP1)	RHOUH	42
	G2 = RHOG(IP1)	RHOUH	43
	G5 = RHOG(I)	RHOUH	44
	E1 = G1*UG(IM1)	RHOUH	45
	E2 = G2*UG(IP1)	RHOUH	46
	E5 = G5*UG(I)	RHOUH	47
	C1 = G1*HG(IM1)	RHOUH	48
	C2 = G2*HG(IP1)	RHOUH	49
	C5 = G5*HG(I)	RHOUH	50
	B1 = BUGGER*UG(IM1)*E1	RHOUH	51
	B2 = BUGGER*UG(IP1)*E2	RHOUH	52
	B5 = BUGGER*UG(I)*E5	RHOUH	53
C		RHOUH	54
	PHIAVE = (F1 + F5 + F5 + F2)*0.25	RHOUH	55
	RHOAVE = (G1 + G5 + G5 + G2)*0.25	RHOUH	56
	UGAVE = (UG(IM1) + UG(I) + UG(I) + UG(IP1))*0.25	RHOUH	57
	IF(UP(I).EQ.0.0.AND.PHIAVE.NE.0.)UP(I)=UGAVE	RHOUH	58
	UPAVE=(UP(IM1)+UP(I)+UP(I)+UP(IP1))*0.25	RHOUH	59
C		RHOUH	60
	UPH1T = UPH1(I)	RHOUH	61
C		RHOUH	62
	URHUT = (F5*AMASS(I)*RHOAVE	RHOUH	63
1	- T2DX*(F2*AMASS(IP1)*E2 - F1*AMASS(IM1)*E1)	RHOUH	64
2	+ PMDOT(I)*AMASS(I))/(UPH1T*AMASS(I))	RHOUH	65
C		RHOUH	66
	PHIRHU = UPH1T*URHUT	RHOUH	67
C		RHOUH	68
	GRAVA = GRAV*AMOM(I)	RHOUH	69
	UUT = (F5*AMOM(I)*(E1 + E5 + E5 + E2)*0.25	RHOUH	70
1	- T2LX*(F2*AMOM(IP1)*E2*UG(IP1) - F1*AMOM(IM1)*E1*UG(IM1)	RHOUH	71
2	+ GRAVA*PG(IP1) - GRAVA*PG(IM1))	RHOUH	72
3	- DELT*FRICT(I) + AMOM(I)*PMDOT(I)*LP(I))/	RHOUH	73
4	(PHIRHU*UAMOM(I))	RHOUH	74
C		RHOUH	75
	IF(ABS(UUT) .LT. 0.1) UUT = 0.0	RHOUH	76
	IF(PHIAVE.LT.0.999) CALL DRAG(UDRAG(I),.TRUE.,1.0)	RHOUH	77
C		RHOUH	78
	LMG(I) = GAM*(F5*AEENER(I)*(C1+B1+C5+B5+C5+B5+C2+B2)/	RHOUH	79
1	(4.0*GAM) - T2DX*(F2*AEENER(IP1)*E2*HG(IP1) -	RHOUH	80
2	F1*AEENER(IM1)*E1*HG(IP1))	RHOUH	81
3	- UCONV(I) + PMDOT(I)*AEENER(I)*HMB	RHOUH	82
		RHOUH	83

4	-UDKAB(I)*UP(I)*DELTAENER(I)/786.	RHOUH	84
4	-BUGGER*UOT*UOT*PHIRHU*UAENER(I)/GAM I/	RHOUH	85
5	(PHIRHU*UAENER(I))	RHOUH	86
		RHOUH	87
	URHUG(I) = URHUT	RHOUH	88
	UUG(I) = UUT	RHOUH	89
	THE SAVED PROPERTIES AT GRID NX SHOULD BE PUT BACK INTO THE	RHOUH	90
	APPROPRIATE ARRAYS BEFORE I IS SET TO NX.	MOD0301	53
	IF (I, NE, NX - 1) GO TO 100	MOD0301	54
	RHUG(NX) = RHOS	MOD0301	55
	UG(NX) = UGS	MOD0301	56
	HG(NX) = HGS	MOD0301	57
	PG(NX) = PGS	MOD0301	58
	UP(NX) = UPS	MOD0301	59
		MOD0301	60
		MOD0301	61
		MOD0301	62
100	CONTINUE	RHOUH	91
		RHOUH	92
	REPLACE T2OX BY ITS SAVED VALUE.	MOD0301	63
	T2OX = T2OXS	MOD0301	64
		MOD0301	65
	RETURN	RHOUH	93
		RHOUH	94
		RHOUH	95
	END	RHOUH	96

	SUBROUTINE PROPMO	PROPMO	2
C		PROPMO	3
C	SUBROUTINE PROPMO CALCULATES PROPELLANT MOTION IN THE BARREL.	PROPMO	4
C		PROPMO	5
	COMMON/CHAM/IX,IX,XB,XB,NGX,NGR,IELGE,IENDB,IPATH(60,5),AREAG(5),	CHAMIX	2
	AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,	CHAMIX	3
	AREAK(60),ARLAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),DAVG,	CHAMIX	4
	AREAH2,DIAMB1,BELBEG,BELBEG,IPF1,IPF2,RADFS,BPIGN	CHAMIX	5
	COMMON/EQNS/LTUX,T2UX,T2UX,TWOTR,UTCR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
	DX,UR,NX,GO,TWOTR,HBP	EQNS	3
	COMMON/GRAIN/ XL(60,5), DU(60,5), DI(60,5), FN,	GRAIN	2
	ALTDI(60,5), DDTOT(60,5), DITOT(60,5), XLO, DOU, DIO,	GRAIN	3
	XLB(100), DOB(100), DIB(100), UXLB(100), LDOB(100), UDIB(100)	GRAIN	4
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,UCONS,RHOF,PHIO,TF,CA,RHOO,	INPUTS	2
	NO,PO,DO,GTRHOP,HW,DM,TIGNBP,GBCONS,TOTM	INPUTS	3
	COMMON/SPLINT/WHOLEC,WHOLEB	PROPMO	10
	COMMON/LAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
	VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),	BAG	3
	DOIMIG(60), GBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
	PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
	VBGTD(60,5), TRC(60,5), DOTMBG(60), COTMP(60,5), PHIBP(60,5),	BAG	6
	PHIPID(60,5), TZK(60), TBP(60,5)	BAG	7
	COMMON/EARL/ PH1(100), RHOG(100), HG(100), UG(100), UP(100),	BARRL	2
	PG(100), TG(100), PMDT(100), GL(100), UCRAG(100), FRICT(100),	BARRL	3
	UCCNV(100), UUP(100), UPHI(100), UHOG(100), UHG(100), UUG(100),	BARRL	4
	AMASS(100), AMOM(100), AENER(100), CAMASS(100), CAMOM(100),	BARRL	5
	CAENER(100)	BARRL	6
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL WHOLEC,WHOLEB	PROPMO	14
C		PROPMO	15
C	LOAD GRID 1 WITH POROSITY AND GRAIN DIMENSIONS UPDATED BY REGRES	PROPMO	16
C	AND PROPELLANT VELOCITY UPDATED BY PRFVEL.	PROPMO	17
	IF(ONED) GO TO 10	PROPMO	16
	PHI(1) = 0.0	PROPMO	19
	ALB(1) = 0.0	PROPMO	20
	DOB(1) = 0.0	PROPMO	21
	DIB(1) = 0.0	PROPMO	22
	UP(1) = 0.0	PROPMO	23
C		PROPMO	24
	DO 5 J=1,NGR	PROPMO	25
	PHI(1) = PHI(1) + PHIBG(NGX,J)*AREAG(J)	PROPMO	26
	TEMP = (1.0 - PHIBG(NGX,J))*AREAG(J)	PROPMO	27
	ALB(1) = ALB(1) + XL(NGX,J)*TEMP	PROPMO	28
	DOB(1) = DOB(1) + DO(NGX,J)*TEMP	PROPMO	29
	DIB(1) = DIB(1) + DI(NGX,J)*TEMP	PROPMO	30
	UP(1) = UP(1) + UPB(NGX,J)*TEMP	PROPMO	31
5	CONTINUE	PROPMO	32
C		PROPMO	33
	PHI(1) = PHI(1)/AREACH	PROPMO	34
	IF(PHI(1) .GE. 0.99995) GO TO 8	MOD0620	3
	TEMP = (1.0 - PHI(1))*AREACH	PROPMO	36
	ALB(1) = ALB(1)/TEMP	PROPMO	37
	DOB(1) = DOB(1)/TEMP	PROPMO	38
	DIB(1) = DIB(1)/TEMP	PROPMO	39
	UP(1) = UP(1)/TEMP	PROPMO	40
	GO TO 15	PROPMO	41
C		PROPMO	42

C	CONTINUE	PROPMO	43
	ALB(1) = 0.0	PROPMO	44
	DOB(1) = 0.0	PROPMO	45
	UIB(1) = 0.0	PROPMO	46
	UP(1) = 0.0	PROPMO	47
	GO TO 15	PROPMO	48
C		PROPMO	49
10	PHI(1) = PHIG(NGX,1)	PROPMO	50
	ALB(1) = AL(NGX,1)	PROPMO	51
	DOB(1) = DO(NGX,1)	PROPMO	52
	UIB(1) = UI(NGX,1)	PROPMO	53
	UP(1) = UPR(NGX,1)	PROPMO	54
C		PROPMO	55
15	CONTINUE	PROPMO	56
	IF(.NOT. WHOLEC .AND. DOB(1) .LE. UIB(1)) WHOLEB = .FALSE.	PROPMO	57
C		PROPMO	58
C	ARRAY UUP IS CLEARED IN MAIN AT EACH TIME INTERVAL.	PROPMO	59
C		PROPMO	60
C	PUT UPDATED POROSITY AND GRAIN DIMENSIONS CALCULATED IN DIMIN INTO	PROPMO	61
C	ARRAYS PHI, DOB, UIB, ALB.	PROPMO	62
	DO 20 I=2,NX	PROPMO	63
	PHI(I) = UPHI(I)	PROPMO	64
	DOB(I) = UDOB(I)	PROPMO	65
	UIB(I) = UUIB(I)	PROPMO	66
	ALB(I) = UXLB(I)	PROPMO	67
20	CONTINUE	PROPMO	68
C		PROPMO	69
	DO 50 I=2,HA	PROPMO	70
C		PROPMO	71
C	UPDATE PROPELLANT VELOCITY. PUT UPDATED VALUES IN ARRAY UP.	PROPMO	72
C	USE AN AVERAGE POROSITY IN UPDATING UP.	PROPMO	73
C	GTHUP = GRAV*DELTA/RDUP	PROPMO	74
	PHIAVE = (PHI(I-1) + PHI(I) + PHI(I) + PHI(I+1))*0.25	PROPMO	75
	IF(I .EQ. NX) PHIAVE = (PHI(I-1) + PHI(I))*0.5	PROPMO	76
	IF(PHIAVE .GE. 0.99999) GO TO 30	PROPMO	77
	DELUP = GTHUP*UDRAG(I)/(1.0 - PHIAVE)	PROPMO	78
	UP(I) = UP(I) + DELUP	PROPMO	79
C		PROPMO	80
C	AMOUNT OF PROPELLANT MOVED FROM ITH GRID INTO I+1ST GRID.	PROPMO	81
30	CONTINUE	PROPMO	82
	DMPN = (1.0 - PHI(I))*DTDX*UP(I)	PROPMO	83
C		PROPMO	84
C	AMOUNT OF PROPELLANT MOVED FROM I-1ST GRID INTO ITH GRID.	PROPMO	85
C	***THIS STATEMENT WILL GET PROPELLANT INTO THE BARREL ORIGINALLY.	PROPMO	86
	DMPN1 = (1.0 - PHI(I-1))*DTDX*UP(I-1)	PROPMO	87
C		PROPMO	88
C	UPDATED POROSITY AS A RESULT OF MOVEMENT	PROPMO	89
	PHI0 = PHI(I) + DMPN - DMPN1	PROPMO	90
	IF(PHIN.LT.0.99999) GO TO 40	PROPMO	91
	UUP(I)=UG(I)	PROPMO	92
	UPHI(I) = 1.0	PROPMO	93
	UXLB(I) = ALB(I)	PROPMO	94
	UDOB(I) = DOB(I)	PROPMO	95
	UUIB(I) = UIB(I)	PROPMO	96
	GO TO 50	PROPMO	97
C		PROPMO	98
C		PROPMO	99

C	VELOCITY AND PROPELLANT DIMENSIONS ARE ADJUSTED ON THE BASIS OF	PROPMO	100
C	THE MOVEMENT	PROPMO	101
40	CONTINUE	PROPMO	102
	UUP(I) = (UP(I)*(1.0 - PHI(I)) + UP(I-1)*UMPM1	PROPMO	103
	- UP(I)*DMPN)/(1.0 - PHIN)	PROPMO	104
	UXLB(I) = (XLB(I)*(1.0 - PHI(I)) + XLB(I-1)*DMPM1	PROPMO	105
	- XLB(I)*DMPN)/(1.0 - PHIN)	PROPMO	106
	IF(WHOLEB .OR. DOB(I) .LT. 1.0E-10) UDOE(I) =	PROPMO	107
	(DOB(I)*(1.0 - PHI(I)) + DOB(I-1)*DMPM1 - DOB(I)*DMPN)/	PROPMO	108
	(1.0 - PHIN)	PROPMO	109
	IF(WHOLEB .OR. DIB(I) .LT. 1.0E-10) UDIE(I) =	PROPMO	110
	(DIB(I)*(1.0 - PHI(I)) + DIB(I-1)*DMPM1 - DIB(I)*DMPN)/	PROPMO	111
	(1.0 - PHIN)	PROPMO	112
C		PROPMO	113
	IF(UUP(I) .LT. 0.0001) UUP(I) = 0.0	PROPMO	114
	IF(UXLB(I) .LT. 1.0E-7) UXLB(I) = 0.0	PROPMO	115
	IF(UDOB(I) .LT. 1.0E-7) UDOB(I) = 0.0	PROPMO	116
	IF(UDIB(I) .LT. 1.0E-7) UDIB(I) = 0.0	PROPMO	117
C		PROPMO	118
	UPHI(I) = PHIN	PROPMO	119
C		PROPMO	120
50	CONTINUE	PROPMO	121
C		PROPMO	122
	RETURN	PROPMO	123
	END	PROPMO	124

SUBROUTINE MOTION		MOTION	2
C	COMMON/BARRL2/BOREA,XP,VP,BOREL,BOREH,BOREDE,DT2BU,DTUSQ,XLBAR	MOTION	3
	COMMON/CLOCK/TIME,DELT	MOTION	4
	COMMON/EQNS/DTUX,T2UX,T2UX,TWOTDR,DTCR,HMB,TWOGJ,DVAXIS,DVAXIT,	MOTION	5
3	UX,UX,NX,GU,TWOTDR,HMB	EQNS	2
	COMMON/GASCON/KU,RKO,CVU,CVH	EQNS	3
	COMMON/GRAIN/ XL(60,5), UD(60,5), UL(60,5), FN,	MOTION	7
1	XLID(60,5), DUTDT(60,5), DITDT(60,5), XLO, DOO, DIO,	MOD0301	66
2	XLB(100), LOB(100), DIB(100), DXLB(100), LOOB(100), UDIB(100)	MOD0301	67
	COMMON/GRIDNX/DXPRIM	MOD0301	68
	COMMON/MOLON/CON1,CON2,CON3,CON4,CONS,AREAPB,ZO,WOB,XOB,ROTK,	MOD0301	69
3	FURAX,CON6,XINT,XLO,PINT,PLO	MOTION	8
	COMMON/P/IPRINI,MOUCH,MOUGR,PR11,IDEELG(35)	MOD1111	33
	COMMON/BARRL/ PH(100), KHOG(100), HG(100), UG(100), UP(100),	P	2
1	PG(100), TG(100), PMDT(100), GL(100), UCRAG(100), FRCT(100),	BARRL	2
2	WCONV(100), UUP(100), UPHI(100), UHOG(100), UHG(100), UUG(100),	BARRL	3
3	AMASS(100), AMOM(100), AENER(100), JAMASS(100), JAMOM(100),	BARRL	4
4	JENER(100)	BARRL	5
	LOGICAL PR11,IDEBUG	BARRL	6
	LOGICAL NAGE2	PLOG	2
	DATA NAGE2/.FALSE./	MOTION	13
	DATA BPRES/2116./	MOTION	14
	DATA CSPD/1100./	MOD1111	34
	DATA DELU/0./	MOD1111	35
C		MOD1111	36
	DATA XJUL/776./	MOTION	15
C		MOTION	16
C	DETERMINE PROPULSIVE FORCE ACTING ON THE PROJECTILE	MOTION	17
	F = PG(NX)*AREAPB	MOTION	18
C		MOTION	19
C	DETERMINE FUPRIM, THE ENGRAVING FORCE AND SLIDING RESISTANCE	MOTION	20
C	XOB IS THE INITIAL PROJECTILE POSITION--POSITION WHERE ENGRAVING	MOTION	21
C	BEGINS. WOB IS THE LENGTH OF THE ENGRAVING BAND.	MOTION	22
	TEMP = XP - XOB	MOTION	23
	IF(TEMP.GT.WOB) GO TO 30	MOTION	24
	FUPRIM = ZO + CON3*TEMP	MOTION	25
	GO TO 50	MOTION	26
C		MOD1111	37
	30 IF(XP.GT.(XINT-XOB)) GO TO 40	MOTION	28
	FUPRIM=FOMAX-CON4*(TEMP-WOB)	MOD1111	38
	GO TO 50	MOD1111	39
40	CONTINUE	MOD1111	40
	FUPRIM=PINT*AREAPB-CON6*(TEMP-XINT)	MOD1111	41
45	DELL=0.2*DELU	MOD1111	42
	CSPD=CSPD + DELL	MOD1111	43
	DELP=1.4*BPRES*DELU/(CSPD-0.6*DELU)	MOD1111	44
	BPRES = BPRES + DELP	MOD1111	45
	FPRIM=BPRES*AREAPB	MOD1111	46
	IF(FUPRIM.LT.FPRIM) FUPRIM=FPRIM	MOD1111	47
50	CONTINUE	MOD1111	48
C		MOD1111	49
C	DETERMINE PROJECTILE AXIAL ACCELERATION	MOTION	32
	ACC = CON2*(F*CON1 - FUPRIM)	MOTION	33
C		MOTION	34
C	IF THERE IS NO MOVEMENT, NX AND DXPRIM REMAIN THE SAME.	MOTION	35
C	C***IF ACC IS USED ELSEWHERE, MAKE IT NONNEGATIVE BEFORE RETURNING	MOTION	36
		MOTION	37

C	IF (ACC .LE. 0.0) RETURN	MOTION	38
C		MOTION	39
C	PROJECTILE AXIAL VELOCITY	MOTION	40
	VP0 = VP	MOTION	41
	VP = VP0 + ACC*DELT	MOTION	42
	DELU=ACC*DELT	MOD1111	50
C		MOTION	43
C	PROJECTILE ANGULAR VELOCITY	MOTION	44
	OMEGA = ROTK*VP*9.5493	MOTION	45
C		MOTION	46
C	PROJECTILE AXIAL POSITION	MOTION	47
	XP = XP + VP0*DELT + ACC*COM5	MOTION	48
C		MOTION	49
C	DETERMINE THE NUMBER OF GRIDS, NX, AND THE SIZE OF THE LAST GRID	MOTION	50
C	NGC SAVES THE NUMBER OF GRIDS BEFORE PROJECTILE MOTION	MOTION	51
	NGC = NX	MOTION	52
C		MOTION	53
C	NUMBER OF GRIDS IN DECIMAL FORM AFTER PROJECTILE MOTION	MOTION	54
	ANG = XP/DX + 1.0	MOTION	55
C		MOTION	56
C	NUMBER OF TRUE GRIDS	MOTION	57
	NX = ANG	MOTION	58
C		MOTION	59
C		MOD0301	70
C		MOD0301	71
C	SAVE THE PREVIOUS DXPRI	MOD0301	72
	DXPS = DXPRI	MOD0301	73
C		MOD0301	74
C	DXPRIM IS THE WIDTH OF THE LAST GRID.	MOD0301	75
C	DXPRIM WILL BE .GE. DX AND .LT. 2*DX.	MOD0301	76
	DXPRIM = (XNG - FLOAT(NX - 1))*DX	MOD0301	77
C		MOTION	64
C		MOTION	65
C		MOTION	66
C	FILL THE ARRAY VALUES AT GRID NX	MOTION	67
C		MOTION	68
C	GET UPDATED PRESSURE AND GAM AT PREVIOUS GRID NX	MOTION	69
	IF (NXGE2) GO TO 80	MOTION	70
	IF (NX .EQ. 1) GO TO 100	MOTION	71
	NAGE2 = .TRUE.	MOTION	72
	UHG(1) = HU(1)	MOTION	73
	URHOG(1) = RHOG(1)	MOTION	74
	UUG(1) = UG(1)	MOTION	75
BY	CONTINUE	MOTION	76
	CALL GSPROP(R0,RRC,R,LV0,CVH,CV,PG(NGC),UHG(NGC),TOUR,URHOG(NGC),	MOTION	77
	LUG(NGC),U,U,GAM,CP,2)	MOTION	78
	GAM1 = GAM - 1.0	MOTION	79
	DELVP = VP - VP0	MOTION	80
	C = SQRT(GAM1*(G0*UHG(NGC) - VP0*VP0*.5))	MOTION	81
	CPRIM = C - 0.5*GAM1*DELVP	MOTION	82
C		MOTION	83
C	GAS VELOCITY AT GRID NX IS THAT OF THE PROJECTILE	MOTION	84
	UUG(NX) = VP	MOTION	85
C		MOTION	86
	TEMP = GAM*DELVP/(C + CPRIM)	MOTION	87
C		MOTION	88
C	NOTE THAT PG(NGC) HOLDS UPDATED PRESSURE AT PREVIOUS NX GRID	MOTION	89

PRES = PG(NGC)*(1.0 - TEMP)/(1.0 + TEMP)	MOTION	90
TEMP = VP*VP/TR000	MOTION	91
UUG(NX) = UPRIM*CPRIM/(GAM1*GJ) + TEMP	MOTION	92
URHOG(NX) = GAM/GAM1*PRES/(XJUL*(UUG(NX) - TEMP))	MOD0002	93
IF(NX .GT. NGC .AND. NX .EQ. 2) UUP(NX) = UUP(1)	MOTION	94
IF(NX .GT. NGC .AND. NX .GT. 2) UUP(NX) = UUP(NX - 1)	MOTION	95
IF(UUP(NX) .GT. UUG(NX)) UUP(NX) = UUG(NX)	MOTION	96
IF(NX .EQ. 1) GO TO 65	MOD0301	78
C***THE FOLLOWING MAY NOT BE NECESSARY	MOD0301	79
UAMASS(NX) = UAMASS(NX-1)	MOTION	98
UAMOM(NX) = UAMOM(NX-1)	MOD0301	80
UAEUER(NX) = UAEUER(NX-1)	MOTION	99
GO CONTINUE	MOTION	100
IF(NX .GT. NGC) GO TO 90	MOTION	101
UPHI(NX) = 1.0 - (1.0 - UPHI(NX))*(LAPS - DX*0.5)/	MOTION	102
(LXPRIM - DX*0.5)	MOTION	103
GO TO 100	MOD0301	81
CONTINUE	MOD0301	82
IF(NX .GT. 2) GO TO 95	MOD0301	83
UPHI(NX) = UPHI(1)	MOD0301	84
UUG(NX) = UUG(1)	MOD0301	85
UUB(NX) = UUB(1)	MOD0301	86
UXLB(NX) = UXLB(1)	MOD0301	87
GO TO 100	MOD0301	88
CONTINUE	MOD0301	89
UPHI(NX) = 1.0 - (1.0 - UPHI(NGC))*(LAPS - CX*0.5)/	MOD0301	90
(LXPRIM + CX*0.5)	MOD0301	91
UUG(NX) = UUG(NX - 1)	MOD0301	92
UUB(NX) = UUB(NX - 1)	MOD0301	93
UXLB(NX) = UXLB(NX - 1)	MOD0301	94
IF A NEW GRID HAS BEEN ADDED, GRID NX - 1 DOES NOT HAVE THE PROPER	MOD0301	95
VALUES EXCEPT FOR UAMASS, UAMOM, AND UAEUER (IN BNDLYR AREAS	MOD0301	96
AT GRID NX ARE SET TO THOSE OF NX - 1). IN THIS CASE NGC IS NOW	MOD0301	97
NX - 1.	MOD0301	98
FRACT = DX/(LX + DXPRIM)	MOD0301	99
URHOG(NX - 1) = URHOG(NX - 2) + FRACT*(URHOG(NX) - URHOG(NX - 2))	MOD0301	100
UUG(NX - 1) = UUG(NX - 2) + FRACT*(UUG(NX) - UUG(NX - 2))	MOD0301	101
UUB(NX - 1) = UUB(NX - 2) + FRACT*(UUB(NX) - UUB(NX - 2))	MOD0301	102
UUP(NX - 1) = UUP(NX - 2) + FRACT*(UUP(NX) - UUP(NX - 2))	MOD0301	103
UPHI(NX - 1) = UPHI(NX)	MOD0301	104
CONTINUE	MOTION	105
IF(XP .GE. XLBAR) PRI1 = .TRUE.	MOTION	106
IF(.NOT. PRI1) RETURN	MOTION	107
IF(.NOT. IDEBUG(23)) RETURN	MOTION	108
IF(XP .GE. XLBAR) WRITE(6,3000) TIME	MOTION	109
WRITE(6,4000) TIME	MOTION	110
WRITE(6,1000)	MOTION	111

PRES = PG(NX)/144.	MOTION	112
DISP = XP*12.	MOTION	113
WRITE(6,2000) ACC,VP,DISP,OMEGA,FDFRIM,PRES	MOTION	114
1000 FORMAT(///,20X,*,INTERIOR BALLISTICS OUTPUT *,//,5X,*,PROJECTILE*,	MOTION	115
1 10X,*,PROJECTILE*,10X,*,PROJECTILE*,10X,*,ROTATIONAL*,10X,	MOTION	116
2 *,PROJECTILE*,7X,*,PRESSURE AT BASE*,//,4X,*,ACCELERATION*,10X,	MOTION	117
3 *,VELOCITY*,10X,*,DISPLACEMENT*,10X,*,VELOCITY*,14X,*,DRAG*,12X,	MOTION	118
4 *,OF PROJECTILE*,)	MOTION	119
2000 FORMAT(6E20,10)	MOTION	120
3000 FORMAT(1H1,*, THE PROJECTILE HAS GONE OUT OF THE BARREL AT TIME*,	MOTION	121
3 E14.8)	MOTION	122
4000 FORMAT(1H0,*, TIME IS*,E14.8)	MOTION	123
RETURN	MOTION	124
END	MOTION	125

C	SUBROUTINE UPDATE	UPDATE	2
C		UPDATE	3
C	SUBROUTINE UPDATE UPDATES THE ARRAYS AT EACH TIME INTERVAL AND	UPDATE	4
C	PRINTS OUTPUT.	UPDATE	5
C		UPDATE	6
	COMMON/FAILEL/THICKT,PHOOP,PCOMP,ETOE,XNTUB,FAIL,MFAIL(60),	UPDATE	7
	1 THICK(60),PANEL1,TYPEL	MOD0925	516
	COMMON/BARRL2/BOREA,XP,VP,BORED,HCHER,BORED2,DT2BU,DTUSQ,XLBAR	UPDATE	9
	COMMON/CHAM/IX,IX,AX,KB,NGX,NGR,IEEGE,IEEND,IPATH(60,5),AREAG(5),	CHAMIX	2
	1 AREACH,AREAC(60),IGNI1,ONED,DIAM1,LIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
	2 AREAX(60),AREAX,CHAM1,CHAM2,CHAM3,10PGAP,AREAGP(60),DAVG,	CHAMIX	4
	3 AREAH2,DIAM2,BELEN2,BELLEG,IPS1,IPS2,RAOFS,BPIGN	CHAMIX	5
	COMMON/CLOCK/TIME,DELT	UPDATE	11
	COMMON/IGNTE/PDRAU,BPCNG,BRNOU,RGBRA,TMPBG,	MOD1002	14
	4 BEGIC(5),XLIC(5)	MOD1002	15
	COMMON/EQNS/DTUX,T2UR,T2UX,TWOTR,DILK,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
	5 UX,UR,NX,UG,TWOT,HEP	EQNS	3
	COMMON/GASCON/KO,RKO,CVO,CVH	UPDATE	13
	COMMON/GRIDX/UXPRIM	MOD0301	113
	COMMON/GASES/RUM6,KRUM6,CVUM6,CVHM6,RUBP,RRCP,CVORP,CVMBP	UPDATE	14
	COMMON/BPT/IEP(2),DEP(2),CEP(2),FIUBR,PDALST	MOD0925	517
	COMMON/PRMFLO/DOTMPM,UPRM,QPRM,PRMXP,POSHR,GSPRKS,PRMCOF	MOD0925	516
	COMMON/GRAIN/ XL(60,5), DO(60,5), LI(60,5), FN,	GRAIN	2
	1 ALTD(60,5), DOTD(60,5), DITD(60,5), XLQ, DOQ, DIO,	GRAIN	3
	2 ALB(100), DOB(100), DIB(100), UALB(100), LDOB(100), UDIB(100)	GRAIN	4
	COMMON/INPUTS/C1,C2,C3,C4,TU,TIGN,ULCHS,RHOF,PHIU,TF,CA,RHOO,	INPUTS	2
	3 HU,PU,UU,GIRKOP,HK,LM,TIGNBP,QBCONS,TOTM	INPUTS	3
	COMMON/P/PHIN1,MOUCH,MODGR,PHI1,ILBLG(35)	P	2
	COMMON/PRIMV/BPLENS,BPKAU(60,5),AGENB,BGENEP,EXPBP	PRIMV	2
	COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
	1 VBG(60,5), UPB(60,5), PCH(60,5), T2C(60,5),	BAG	3
	2 UPMIG(60), VBAF(60,5), XDRAG(60,5), DOTPB(60,5), UPMOT(60,5),	BAG	4
	3 PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
	4 VBGTD(60,5), TBG(60,5), DOTMBG(60), DOTMP(60,5), PHIBP(60,5),	BAG	6
	5 PHIBTD(60,5), T2R(60), TBP(60,5)	BAG	7
	COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),	BARRL	2
	1 PG(100), TG(100), PMDT(100), GL(100), UCRAG(100), FRICT(100),	BARRL	3
	2 UCONV(100), DUP(100), UPHI(100), UHOG(100), UHG(100), UUG(100),	BARRL	4
	3 AMASS(100), AHOM(100), AENER(100), CAMASS(100), UAMOM(100),	BARRL	5
	4 UAENER(100)	BARRL	6
	LOGICAL IGNI1,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL PRI1,IDEBUG	PLOG	2
	LOGICAL PRI2,VAR	UPDATE	23
	LOGICAL FAIL	UPDATE	24
	LOGICAL PENET	MOD0605	27
	DATA DOTMBS,DOTMPS,PMOUTS/0.0,0.0,0.0,	UPDATE	25
	DATA 124/0/	UPDATE	26
C		UPDATE	27
C		UPDATE	28
C		UPDATE	29
C		UPDATE	30
C	*****	UPDATE	31
C	UPDATE CHAMBER ARRAYS	UPDATE	32
C	*****	UPDATE	33
C		UPDATE	34
C	IF(IDEBUG(15).AND. PRI1) WRITE(6,2003) IPRINT,TIME	UPDATE	35
C		UPDATE	36

C	IF THE POROSITY CONDITION IS NOT SATISFIED, VAR WILL BE SET FALSE.	UPDATE	37
C	VAR = .TRUE.	UPDATE	38
C	DO 100 I=1,NXX	UPDATE	39
	PR12 = .FALSE.	UPDATE	40
	IF (MOD(I,MODGR) .EQ. 0 .OR. I .EQ. 1 .OR. I .EQ. NXX)	UPDATE	41
1	PR12 = .TRUE.	UPDATE	42
C		UPDATE	43
	DO 50 J=1,NGR	UPDATE	44
	PH1BG(I,J) = PH1BTD(I,J)	UPDATE	45
	PH1BP(I,J) = PH1PTD(I,J)	UPDATE	46
	RHOBG(I,J) = RHOBTD(I,J)	UPDATE	47
	UBG(I,J) = UBGTD(I,J)	UPDATE	48
	VBG(I,J) = VBGTD(I,J)	UPDATE	49
	HBG(I,J) = HBSTD(I,J)	UPDATE	50
	UPB(I,J) = UPBTD(I,J)	UPDATE	51
C		UPDATE	52
C	UPDATE TBG AND PCH BY CALLING GSPROP WITH HBG AND RHOBG	UPDATE	53
C	CALL GSPROP(RG,RKD,R,CV0,CV1,CV,PCH(I,J),HBG(I,J),	UPDATE	54
1	TBG(I,J),RHOBG(I,J),LBG(I,J),VBG(I,J),GAM,CP,2)	UPDATE	55
C		UPDATE	56
	XL(I,J) = XLTD(I,J)	UPDATE	57
	DO(I,J) = DOTD(I,J)	UPDATE	58
	DI(I,J) = DITD(I,J)	UPDATE	59
	IF (PCH(I,J) .LE. 0.0 .OR. HBG(I,J) .LE. 0.0 .OR. RHOBG(I,J) .LE. 0.0)	UPDATE	60
	*GO TO 10	UPDATE	61
C		UPDATE	62
	IF (.NOT. PR11) GO TO 40	UPDATE	63
	IF (.NOT. IDBUG(15)) GO TO 40	UPDATE	64
	IF (.NOT. PR12) GO TO 40	UPDATE	65
10	CONTINUE	UPDATE	66
	PRES = PCH(I,J)/144.	UPDATE	67
	WRITE(6,2001) I,J,PH1BG(I,J),RHOBG(I,J),UBG(I,J),VBG(I,J),	UPDATE	68
1	HBG(I,J),TBG(I,J),PRES,PH1BP(I,J),UPB(I,J),TZC(I,J),	UPDATE	69
2	UBAG(I,J),XDRAG(I,J),COTPC(I,J),DOTMP(I,J),TBP(I,J),	UPDATE	70
3	XL(I,J),DO(I,J),DI(I,J)	UPDATE	71
40	CONTINUE	UPDATE	72
C		UPDATE	73
50	CONTINUE	UPDATE	74
C		UPDATE	75
C	POROSITY TEST	UPDATE	76
	IF (UNED) GO TO 100	UPDATE	77
	IF (.NOT. VAR) GO TO 100	UPDATE	78
	IF (1 .LE. IBLEB) GO TO 100	UPDATE	79
	IF (I .GT. IENDB) GO TO 100	UPDATE	80
	IF (PH1BP(1,1) .LT. 0.999) VAR = .FALSE.	UPDATE	81
100	CONTINUE	UPDATE	82
C		UPDATE	83
	NAMELIST/DOT/DOTM15,DOIMBG,TZR	UPDATE	84
	IF (UNED) GO TO 110	UPDATE	85
	IF (IDBUG(16) .AND. PR11) WRITE(6,DCT)	UPDATE	86
110	CONTINUE	UPDATE	87
C		UPDATE	88
C		UPDATE	89
C		UPDATE	90
C	*****	UPDATE	91
C	DETERMINE WHETHER THE CHAMBER SHOULD BE MADE 1-DIMENSIONAL	UPDATE	92
C	*****	UPDATE	93

C	IF (UNED) GO TO 150	UPDATE	94
	IF (.NOT. VAR) GO TO 150	UPDATE	95
	IF (.NOT. IGNIT) GO TO 150	UPDATE	96
	UNED = .TRUE.	UPDATE	97
	CALL UNEDIM	UPDATE	98
	IF (IDEBUG(17)) WRITE(6,2006) TIME	UPDATE	99
	IF (IDEBUG(17)) WRITE(6,2009) IPKJAI	UPDATE	100
	IF (.NOT. IDEBUG(18)) GO TO 150	UPDATE	101
	WRITE(6,2007)	UPDATE	102
	DO 150 I=1,NXX	UPDATE	103
	PRES = PCN(I,1)/144.	UPDATE	104
	WRITE(6,2008) I,PHI(I,1),RHOG(I,1),UG(I,1),HIG(I,1),	UPDATE	105
	TBG(I,1),PRES,UPB(I,1),XL(I,1),DO(I,1),DI(I,1)	UPDATE	106
100	CONTINUE	UPDATE	107
C		UPDATE	108
150	CONTINUE	UPDATE	109
C		UPDATE	110
C		UPDATE	111
C		UPDATE	112
C	*****	UPDATE	113
C	UPDATE BARKEL ARRAYS	UPDATE	114
C	*****	UPDATE	115
C		UPDATE	116
	IF (.NOT. PRI1) GO TO 170	UPDATE	117
	IF (.NOT. IDEBUG(19)) GO TO 170	UPDATE	118
	WRITE(6,2004)	UPDATE	119
	I = 1	UPDATE	120
	PRES = PG(I)/144.	UPDATE	121
	WRITE(6,2005) I,PHI(I),RHOG(I),UG(I),HIG(I),TG(I),PRES,UP(I),	UPDATE	122
	UL(I),UDRAG(I),PMDOT(I),APASS(I),AMOM(I),AENER(I),	UPDATE	123
	XLB(I),DUB(I),DIB(I)	UPDATE	124
170	CONTINUE	UPDATE	125
C		UPDATE	126
	IF (NXX .LT. 2) GO TO 220	UPDATE	127
	DO 200 I=2,NXX	UPDATE	128
	PHI(I) = UPHI(I)	UPDATE	129
	RHOG(I) = URHOG(I)	UPDATE	130
	UG(I) = UUG(I)	UPDATE	131
	HIG(I) = UHIG(I)	UPDATE	132
	UP(I) = UUP(I)	UPDATE	133
C		UPDATE	134
C	UPDATE TG AND PG BY CALLING GSPROP WITH HG AND RHOG	UPDATE	135
	CALL GSPROP(RH,RHO,K,CVO,CVH,CV,PG(I),HG(I),TG(I),RHOG(I),	UPDATE	136
	UG(I),Q,Q,GAM,CP,2)	UPDATE	137
C		UPDATE	138
	XLB(I) = UXLB(I)	UPDATE	139
	DUB(I) = UDUB(I)	UPDATE	140
	DIB(I) = UDIB(I)	UPDATE	141
C		UPDATE	142
	AMASS(I) = UAMASS(I)	UPDATE	143
	AMOM(I) = UAMOM(I)	UPDATE	144
	AENER(I) = UAENER(I)	UPDATE	145
	IF (PG(I).LE.0.0.OR.HG(I).LE.0.0.OR.RHOG(I).LE.0.0) GO TO 180	UPDATE	146
C		UPDATE	147
	IF (.NOT. PRI1) GO TO 200	UPDATE	148
	IF (.NOT. IDEBUG(19)) GO TO 200	UPDATE	149
180	CONTINUE	UPDATE	150

	PRES = PC(1)/144.	UPDATE	151
	WRITE(6,2005) I,PHI(1),RHOG(I),CG(I),HG(I),TG(I),PRES,	UPDATE	152
	UP(I),QL(I),CDRAG(I),PMDOT(I),PMASS(I),AMOM(I),ALNEK(I),	UPDATE	153
	1 XLB(I),DUF(I),UIB(I)	UPDATE	154
200	CONTINUE	UPDATE	155
220	CONTINUE	UPDATE	156
C		UPDATE	157
C	*****	UPDATE	158
C	PRINT OUT CERTAIN PRESSURES	UPDATE	159
C	*****	UPDATE	160
C		UPDATE	161
	IF(.NOT. PHI1) GO TO 240	UPDATE	162
	IF(.NOT. IDEHOG(22)) GO TO 240	UPDATE	163
	P1 = PC(1)/144.	UPDATE	164
	WRITE(6,2014) P1	UPDATE	165
C		UPDATE	166
240	CONTINUE	UPDATE	167
C		UPDATE	168
C	*****	UPDATE	169
C	COMPUTE MASS OF GAS IN THE SYSTEM AND THE MASS OF BLACK POWDER	UPDATE	170
C	AND PROPELLANT IN THE SYSTEM	UPDATE	171
C	*****	UPDATE	172
C		UPDATE	173
C	CALCULATIONS WILL BE DONE EACH TIME INTERVAL AND DENSITIES WILL BE	UPDATE	174
C	ADJUSTED IF MASS IS LOST.	UPDATE	175
C		UPDATE	176
C	LOGIC IS NOT WRITTEN FOR THE CASE WHERE CHAM1 IS TRUE	UPDATE	177
	IF(.NOT. CHAM1) GO TO 250	UPDATE	178
	WRITE(6,2010)	UPDATE	179
	GO TO 500	UPDATE	180
C		UPDATE	181
250	CONTINUE	UPDATE	182
	IF(ONEU) GO TO 300	UPDATE	183
	VOL = DVAXIS*0.5	UPDATE	184
	GASMAS = (PHIBG(1,1) + PHIBP(1,1) - 1.0)*RHOBG(1,1)*VOL	UPDATE	185
	PROMAS = ((1.0 - PHIBG(1,1))*RHOP + (1.0 - PHIBP(1,1))*BPDENS)*	UPDATE	186
	VOL	UPDATE	187
	VOL = DVAXIS	UPDATE	188
	VOLROP = VOL*RHOP	UPDATE	189
	VOLBPD = VOL*BPDENS	UPDATE	190
	DO 260 I=2,NGX	UPDATE	191
	GASMAS = GASMAS + (PHIBG(I,1) + PHIBP(I,1) - 1.0)*RHOBG(I,1)*	UPDATE	192
	VOL	UPDATE	193
	PROMAS = PROMAS + (1.0 - PHIBG(I,1))*VOLROP +	UPDATE	194
	(1.0 - PHIBP(I,1))*VOLBPD	UPDATE	195
260	CONTINUE	UPDATE	196
C		UPDATE	197
	VOL = AREAK(1)*DX*0.5	UPDATE	198
	GASMAS = GASMAS + (PHIBG(1,2) + PHIBP(1,2) - 1.0)*RHOBG(1,2)*VOL	UPDATE	199
	PROMAS = PROMAS + ((1.0 - PHIBG(1,2))*RHOP +	UPDATE	200
	(1.0 - PHIBP(1,2))*BPDENS)*VOL	UPDATE	201
	DO 270 I=2,NGX	UPDATE	202
	VOL = AREAK(I)*DX	UPDATE	203
	GASMAS = GASMAS + (PHIBG(I,2) + PHIBP(I,2) - 1.0)*RHOBG(I,2)*	UPDATE	204
	VOL	UPDATE	205
	PROMAS = PROMAS + ((1.0 - PHIBG(I,2))*RHOP +	UPDATE	206
	(1.0 - PHIBP(I,2))*BPDENS)*VOL	UPDATE	207

270	CONTINUE	UPDATE	208
	IF (CHAM2) GO TO 320	UPDATE	209
C		UPDATE	210
C	THERE IS NO PROPELLANT AND NO BLACK POWDER IN RADIAL ROW 3 (WHEN	UPDATE	211
C	CHAM2 IS TRUE)	UPDATE	212
	VOL = AREAGP(1)*DX*0.5	UPDATE	213
	GASMAS = GASMAS + RHOBG(1,3)*VOL	UPDATE	214
	DO 280 I=2,NX	UPDATE	215
	VOL = AREAGP(I)*DX	UPDATE	216
	GASMAS = GASMAS + RHOBG(I,3)*VOL	UPDATE	217
280	CONTINUE	UPDATE	218
	GO TO 320	UPDATE	219
C		UPDATE	220
C	CALCULATIONS WHEN CHAMBER IS ONE DIMENSIONAL	UPDATE	221
300	CONTINUE	UPDATE	222
	VOL = AREAC(1)*DX*0.5	UPDATE	223
	GASMAS = (PHIBG(1,1) + PHIBP(1,1) - 1.0)*RHCBG(1,1)*VOL	UPDATE	224
	PRUMAS = ((1.0 - PHIBG(1,1))*RHOP + (1.0 - PHIBP(1,1))*BPDENS)*	UPDATE	225
	VOL	UPDATE	226
	DO 310 I=2,NX	UPDATE	227
	VOL = AREAC(I)*DX	UPDATE	228
	GASMAS = GASMAS + (PHIBG(I,1) + PHIBP(I,1) - 1.0)*RHCBG(I,1)*	UPDATE	229
	VOL	UPDATE	230
	PRUMAS = PRUMAS + ((1.0 - PHIBG(I,1))*RHOP +	UPDATE	231
	(1.0 - PHIBP(I,1))*BPDENS)*VOL	UPDATE	232
310	CONTINUE	UPDATE	233
C		UPDATE	234
C	BARREL CALCULATIONS	UPDATE	235
320	CONTINUE	UPDATE	236
	IF (NX .EQ. 1) GO TO 360	UPDATE	237
	VOL = BOREA*DX	UPDATE	238
	VOLROP = VOL*RHOP	UPDATE	239
	IF (NX .EQ. 2) GO TO 340	UPDATE	240
	NX1 = NX - 1	UPDATE	241
	DO 330 I=2,NX1	UPDATE	242
	GASMAS = GASMAS + PHI(I)*RHOG(I)*VOL	UPDATE	243
	PRUMAS = PRUMAS + (1.0 - PHI(I))*VOLROP	UPDATE	244
330	CONTINUE	UPDATE	245
C		UPDATE	246
340	CONTINUE	UPDATE	247
	VOL = BOREA*(DXPRIM - 0.5*DX)	MOD0301	114
	VOLROP = VOL*RHOP	MOD0301	115
	GASMAS = GASMAS + PHI(NX)*RHOG(NX)*VOL	MOD0301	116
	PRUMAS = PRUMAS + (1.0 - PHI(NX))*VOLROP	MOD0301	117
C		UPDATE	250
360	IF (PHI1 .AND. IDEBUG(20)) WRITE(6,2011) GASMAS,PRUMAS	UPDATE	251
C		UPDATE	252
C	ADJUST DENSITIES	UPDATE	253
	ACTGAS = TOTM - PRUMAS	UPDATE	254
	ADJUST = ACTGAS/GASMAS	UPDATE	255
	DO 370 J=1,NCR	UPDATE	256
	DO 370 I=1,NX	UPDATE	257
	RHOBG(I,J) = RHOBG(I,J)*ADJUST	UPDATE	258
370	CONTINUE	UPDATE	259
	DO 380 I=2,NX	UPDATE	260
	RHOG(I) = RHOG(I)*ADJUST	UPDATE	261
380	CONTINUE	UPDATE	262

C	IF (PR11 .AND. IDEBUG(20)) WRITE(6,2013) ACTEAS,ADJUST	UPDATE	263
C		UPDATE	264
C		UPDATE	265
C		UPDATE	266
C	*****	UPDATE	267
C	KEEP A RUNNING SUM OF DOTMB,DOTMP, AND PMDOT TERMS AND	UPDATE	268
C	DETERMINE GAS CONSTANTS ON THE BASIS OF THESE	UPDATE	269
C	*****	UPDATE	270
C		UPDATE	271
C		UPDATE	272
C	IF CHAM1 IS TRUE THIS LOGIC IS SKIPPED.	UPDATE	273
C	400 CONTINUE	UPDATE	274
C	IF (ONEU) GO TO 420	UPDATE	275
C	DOTMBS = DOTMBS + (DOTMB(1,1)*DVAXIS + DOTME(1,2)*AREAR(1)*DX)*0.5	UPDATE	276
C	DOTMPS = DOTMPS + (DOTMP(1,1)*DVAXIS + DOTMF(1,2)*AREAR(1)*DX)*0.5	UPDATE	277
C	DO 410 I=2,NGX	UPDATE	278
C	DOTMBS = DOTMBS + DOTMB(I,1)*CVAXIS + DCTMB(I,2)*AREAR(I)*DX	UPDATE	279
C	DOTMPS = DOTMPS + DOTMP(I,1)*CVAXIS + DCTMP(I,2)*AREAR(I)*DX	UPDATE	280
C	410 CONTINUE	UPDATE	281
C	GO TO 440	UPDATE	282
C		UPDATE	283
C	CALCULATIONS WHEN THE CHAMBER IS ONE DIMENSIONAL	UPDATE	284
C	420 CONTINUE	UPDATE	285
C	DOTMBS = DOTMBS + DOTMB(1,1)*ANEAC(1)*DX*0.5	UPDATE	286
C	DOTMPS = DOTMPS + DOTMP(1,1)*ANEAC(1)*DX*0.5	UPDATE	287
C	DO 430 I=2,NGX	UPDATE	288
C	DOTMBS = DOTMBS + DOTMB(I,1)*AREAL(I)*DX	UPDATE	289
C	DOTMPS = DOTMPS + DOTMP(I,1)*AREAL(I)*DX	UPDATE	290
C	430 CONTINUE	UPDATE	291
C		UPDATE	292
C		UPDATE	293
C	BARREL CALCULATIONS	UPDATE	294
C	440 CONTINUE	UPDATE	295
C	IF (NX .EQ. 1) GO TO 470	UPDATE	296
C	VOL = BOREA*DX	UPDATE	297
C	IF (NX .EQ. 2) GO TO 460	UPDATE	298
C	NX1 = NX - 1	UPDATE	299
C	DO 450 I=2,NX1	UPDATE	300
C	PMDOTS = PMDOTS + PMDOT(I)*VOL	UPDATE	301
C		MOD0301	118
C	460 PMDOTS = PMDOTS + PMDOT(NX)*BOREA*(UJ-RIM - DX*0.5)	MOD0301	119
C		UPDATE	303
C	470 CONTINUE	UPDATE	304
C	PROPS = DOTMBS + PMDOTS	UPDATE	305
C	DENOM = PROPS + DOTMPS	UPDATE	306
C	IF (DENOM .LT. 0.0001) GO TO 480	UPDATE	307
C	FRACM6 = PROPS/DENOM	UPDATE	308
C	FRACBP = DOTMPS/DENOM	UPDATE	309
C	RU = FRACM6*RMOM6 + FRACBP*ROBP	UPDATE	310
C	RRU = FRACM6*RRMOM6 + FRACBP*RRUBP	UPDATE	311
C	CVU = FRACM6*CVOM6 + FRACBP*CVOBP	UPDATE	312
C	CVH = FRACM6*CVHM6 + FRACBP*CVHBP	UPDATE	313
C		UPDATE	314
C	480 CONTINUE	UPDATE	315
C	IF (PR11 .AND. IDEBUG(21)) WRITE(6,2012) DOTMBS,DOTMPS,PMDOTS,RU,	UPDATE	316
C	RRU, CVU, CVH	UPDATE	317
C	*****	UPDATE	318

C	GET READY FOR THE NEXT TIME INTERVAL	UPDATE	319
C	*****	UPDATE	320
C	GOO CONTINUE	UPDATE	321
C		UPDATE	322
C	CLEAR ARRAYS DOTMIG AND DOTMBG FOR SUBROUTINE MFLOW, ARRAY DOTM	UPDATE	323
C	FOR PRIMEX, ARRAY GQAG FOR PRPFIX, ARRAY DOTPH FOR REGRES, AND	UPDATE	324
C	ARRAY UPHDI FOR PRPVEL.	UPDATE	325
	CALL CLEAR(DOTMIG(1),QBAG(60.5))	UPDATE	326
	CALL CLEAR(DOTMH(1,1),UPHDI(60.5))	UPDATE	327
	CALL CLEAR(DOTMBG(1),DOTMP(60.5))	UPDATE	328
C		UPDATE	329
C		UPDATE	330
C		UPDATE	331
C	CLEAR ARRAY PPDUT FOR SUBROUTINE DIMA, ARRAY QL FOR BNDLYR,	UPDATE	332
C	ARRAY UQKAG FOR RHOM, AND ARRAY ULP FOR PROPMO,	UPDATE	333
	CALL CLEAR(PPDUT(1), UQP(100))	UPDATE	334
	P11=PCM(1,1)/144.0	UPDATE	335
	P12=PCM(1,2)/144.0	UPDATE	336
	PN1=PCM(NGX,1)/144.0	UPDATE	337
	PN2=PCM(NGX,2)/144.0	UPDATE	338
	PGN=PG(NX)/144.0	UPDATE	339
	APPF=AP*12.0	UPDATE	340
	124=124+1	UPDATE	341
	1124=MOD(124,10)	UPDATE	342
	IF(10EBUG(24).AND.1124.EQ.0)WRITE(6,2015)IPRINT,TIME,P11,	UPDATE	343
	1 P12,P11,PN2,PGN,APPF,VP	UPDATE	344
	IF(10EBUG(34).AND.TIME.LT.0.005,WRITE(6,2016)	MOD0605	28
	* TIME,DOTMPH,QPMP,PRMXP,PUSHR	MOD0604	6
C		UPDATE	345
C	RETURN	UPDATE	346
C		UPDATE	347
C		UPDATE	348
C		UPDATE	349
C	*****	UPDATE	350
C		UPDATE	351
	2001 FORMAT(/,215,7(2X,E14.8),/,10X,7(2X,E14.8),/,	UPDATE	352
	5 10X,4(2X,E14.8))	UPDATE	353
	2002 FORMAT(//,' IPRINT = ',16,	UPDATE	354
	1 /,' TIME =',E14.8,/,3X,'I',4X,'J',6X,'PHIBG',11X,	UPDATE	355
	1 'RHUBG',12X,'UBG',13X,'VBG',13X,'HCU',13X,'TBG',13X,'PCH',	UPDATE	356
	2 /,15X,'PHIBP',11X,'UPB',13X,'TZC',15X,'QBAG',11X,'XDRAG',11X,	UPDATE	357
	3 'DOTMB',11X,'DOTMP',/,16X,'TBP',14X,'XL',14X,'DO',14X,'DI')	UPDATE	358
	2004 FORMAT(//,3X,'I',5X,'PHI',13X,'RHU',13X,'LG',14X,'HG',14X,	UPDATE	359
	1 '10',14X,'PG',14X,'UP',/,29X,'GL',12X,'QKAG',11X,'PPDUT',	UPDATE	360
	2 11X,'AMASS',12X,'AMOM',11X,'ALEN',	UPDATE	361
	3 /,26X,'XLE',13X,'DOB',13X,'DIE')	UPDATE	362
	2005 FORMAT(//,15, 7(2X,E14.8),/, 21X, 6(2X,E14.8), /,21X, 3(2X,E14.8))	UPDATE	363
	2006 FORMAT('1 AT TIME ',E13.7,' THE CHAMBER WAS MADE 1-DIMENSIONAL	UPDATE	364
	SL')	UPDATE	365
	2007 FORMAT(//,3X,'I',6X,'PHIBG',11X,'RHUBG',12X,'UBG',13X,'HBG',	UPDATE	366
	5 13X,'TBG',13X,'PCH',13X,'UPB',/,26X,'XL',14X,'DO',14X,'DI')	UPDATE	367
	2008 FORMAT(/,15,7(2X,E14.8),/,21X,3(2X,E14.8))	UPDATE	368
	2009 FORMAT(//,' IPRINT = ',15)	UPDATE	369
	2010 FORMAT(' LOGIC FOR SUMMING GAS MASS AND PROPELLANT MASS HAS NOT B	UPDATE	370
	SEEN WRITTEN FOR CHAM1 TRUE')	UPDATE	371
	2011 FORMAT(' THE MASS OF GAS IN THE SYSTEM IS',F10.4,/,	UPDATE	372
	5 ' THE MASS OF PROPELLANT AND BLACK POWDER IN THE SYSTEM IS',	UPDATE	373

3 F10.4)	UPDATE	374
2012 FORMAT(//,' DUTMBS = ',F10.4,' EQUIPFS = ',F10.4,' PMDOTS = ',	UPDATE	375
3 F10.4,/,', FOR THE NEXT TIME INTERVAL RO = ',E10.4,' KRO = ',	UPDATE	376
3 E10.4,' CVO = ',E10.4,' CVM = ',E10.4,/,)	UPDATE	377
2013 FORMAT(' ACTUAL GAS IN THE SYSTEM IS ', F10.4,	UPDATE	378
3 ' ADJUSTING FRACTION IS ',F10.4,/,)	UPDATE	379
2014 FORMAT(1H0,'PRESSURE AT GRID (1,1) IS ', E20.10)	UPDATE	380
2015 FORMAT(2X, 15, 3X, 0(E13.6,2X))	UPDATE	381
2016 FORMAT(//,2X,5HTIME=,E14.8,3X,7HDCIMPR=,E14.5,	MOD0611	6
3 3X,5HUMPR=,E14.5,3X,6HPMXP=,E14.5,3X,6HPSHR=,E14.5)	MOD0611	7
LNU	UPDATE	382

C	SUBROUTINE JREFIF	ONEDIM	2
	COMMON/BANK2/BOREA,AP,VP,BORED,HORH,BORED2,DT2BU,DTUS0,XLRAR	ONEDIM	3
	COMMON/CHAM/IX,IR,XE,KB,NGX,NGR,IBLGB,IENDB,IPATH(60,5),AREAG(5),	ONEDIM	4
1	AREACH,AREAL(60),IGNIT,ONED,DIAM1,LIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	2
2	AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMIX	3
3	AREAH2,DIABT,BLEND,BELBEG,IPSI,IP32,RADFS,BPIGN	CHAMIX	4
	COMMON/EQNS/DIOX,IZOR,IZOX,TWOIDR,IGL,HMB,THOGJ,DVAXIS,DVAXIT,	CHAMIX	5
4	UX,UK,NX,GO,TWOBT,HEP	EQNS	2
	COMMON/GASCOR/KQ,RKU,CVO,CVH	EQNS	3
	COMMON/GRAIN/ XL(60,5), DO(60,5), OI(60,5), FN,	ONEDIM	7
1	XLTD(60,5), DOTD(60,5), ITDT(60,5), XLO, DOU, DIO,	GRAIN	2
2	ALB(100), LOB(100), DIB(100), UALB(100), LOOB(100), UDIB(100)	GRAIN	3
	COMMON/INPUTS/C1,C2,C3,C4,TQ,TIGN,GCNS,RHOF,PHIQ,TF,CA,RHUQ,	GRAIN	4
3	HU,P0,U0,GTTHOP,HK,DM,TIGNBP,QBCNS,TOTM	INPUTS	2
	COMMON/PRIMV/BPLENS,BPKAU(60,5),AGLEP,BGENEP,EXPBP	INPUTS	3
	COMMON/BAG/PHIBG(60,5), KHORG(60,5), HBG(60,5), UBG(60,5),	PRIMV	2
1	VBG(60,5), UFB(60,5), PCH(60,5), T2C(60,5),	BAG	2
2	UIMIG(60), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	3
3	PHIBTD(60,5), KHUBTD(60,5), HBGLD(60,5), UBGTD(60,5),	BAG	4
4	VBGTD(60,5), THG(60,5), DOTMBG(60), DOTMP(60,5), PHIBP(60,5),	BAG	5
5	PHIPTD(60,5), IZK(60), IZP(60,5)	BAG	6
	LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,CPIGN	BAG	7
		CHAMLOG	2
C	THIS SUBROUTINE PUTS THE CORRECT CHAMBER AREAS INTO ARRAY AREAC.	ONEDIM	13
C	PUTS VOLUME-WEIGHTED AVERAGES OF THE GAS ARRAYS INTO THE ARRAY	ONEDIM	14
C	ELEMENTS AT GRID 1. CLEARS THE GAS ARRAYS AT THE OTHER GRIDS.	ONEDIM	15
C	AND SETS NGR TO 1.	ONEDIM	16
C		ONEDIM	17
	DATA PUDF/.785398/	ONEDIM	18
C		ONEDIM	19
C		ONEDIM	20
C		ONEDIM	21
C	*****	ONEDIM	22
C	FILL ARRAY AREAC	ONEDIM	23
C	*****	ONEDIM	24
C		ONEDIM	25
	IF(CHAM3) GO TO 30	ONEDIM	26
	DO 20 I=1,NGX	ONEDIM	27
	AREAC(I) = AREAR(I) + AREAX	ONEDIM	28
20	CONTINUE	ONEDIM	29
	GO TO 50	ONEDIM	30
C		ONEDIM	31
30	CONTINUE	ONEDIM	32
	DO 40 I=1,NGX	ONEDIM	33
	AREAC(I) = AREAR(I) + AREAX + AREAGP(I)	ONEDIM	34
40	CONTINUE	ONEDIM	35
C		ONEDIM	36
50	AREAC(NGX+1) = BOREA	ONEDIM	37
C		ONEDIM	38
C		ONEDIM	39
C	*****	ONEDIM	40
C	FILL EACH CHAMBER ARRAY AT (I,1) WITH THE VOLUME-WEIGHTED AVERAGE	ONEDIM	41
C	OF THE ARRAY VALUES AT 1.	ONEDIM	42
C	*****	ONEDIM	43
C		ONEDIM	44
	IF(CHAM2) GO TO 400	ONEDIM	45
		ONEDIM	46

	IF (CHAR3) GO TO 500	ONEDIM	47
	DO 300 I=1,NGX	ONEDIM	48
C		ONEDIM	49
C	NAME T4C GREATER THAN TIGN SO THAT RLGRES CALCULATIONS ARE DONE.	ONEDIM	50
	T4C(I,1) = TIGN + 1.0	ONEDIM	51
C		ONEDIM	52
	RHUBG(I,1) = RHUBG(I,1)*AREAG(1)	ONEDIM	53
	HBG(I,1) = HBG(I,1)*AREAG(1)	ONEDIM	54
	UBG(I,1) = UBG(I,1)*AREAG(1)	ONEDIM	55
	PCH(I,1) = PCH(I,1)*AREAG(1)	ONEDIM	56
	TEMP = (1.0 - PHIBG(I,1))*AREAG(1)	ONEDIM	57
	UPB(I,1) = UPB(I,1)*TEMP	ONEDIM	58
	XL(I,1) = XL(I,1)*TEMP	ONEDIM	59
	UO(I,1) = UO(I,1)*TEMP	ONEDIM	60
	UI(I,1) = UI(I,1)*TEMP	ONEDIM	61
C	THIS STATEMENT SHOULD COME AFTER THE CALCULATIONS OF TEMP,	ONEDIM	62
	PHIBG(I,1) = PHIBG(I,1)*AREAG(1)	ONEDIM	63
	BPRAD(I,1) = BPRAD(I,1)*(1.0 - PHIBP(I,1))*AREAG(1)	ONEDIM	64
	PHIBP(I,1) = PHIBP(I,1)*AREAG(1)	ONEDIM	65
C		ONEDIM	66
	DO 250 J=2,NGR	ONEDIM	67
	RHUBG(I,1) = RHUBG(I,1) + RHUBG(I,J)*AREAG(J)	ONEDIM	68
	HBG(I,1) = HBG(I,1) + HBG(I,J)*AREAG(J)	ONEDIM	69
	UBG(I,1) = UBG(I,1) + UBG(I,J)*AREAG(J)	ONEDIM	70
	PCH(I,1) = PCH(I,1) + PCH(I,J)*AREAG(J)	ONEDIM	71
	PHIBG(I,1) = PHIBG(I,1) + PHIBG(I,J)*AREAG(J)	ONEDIM	72
	TEMP = (1.0 - PHIBG(I,J))*AREAG(J)	ONEDIM	73
	UPB(I,1) = UPB(I,1) + UPB(I,J)*TEMP	ONEDIM	74
	XL(I,1) = XL(I,1) + XL(I,J)*TEMP	ONEDIM	75
	DO(I,1) = DO(I,1) + DO(I,J)*TEMP	ONEDIM	76
	DI(I,1) = DI(I,1) + DI(I,J)*TEMP	ONEDIM	77
	BPRAD(I,1) = BPRAD(I,1) + BPRAD(I,J)*(1.0 - PHIBP(I,J))*	ONEDIM	78
	AREAG(J)	ONEDIM	79
	PHIBP(I,1) = PHIBP(I,1) + PHIBP(I,J)*AREAG(J)	ONEDIM	80
250	CONTINUE	ONEDIM	81
C		ONEDIM	82
	RHUBG(I,1) = RHUBG(I,1)/AREACH	ONEDIM	83
	HBG(I,1) = HBG(I,1)/AREACH	ONEDIM	84
	UBG(I,1) = UBG(I,1)/AREACH	ONEDIM	85
	PCH(I,1) = PCH(I,1)/AREACH	ONEDIM	86
	PHIBP(I,1) = PHIBP(I,1)/AREACH	ONEDIM	87
	TEMP2 = (1.0 - PHIBP(I,1))*AREACH	ONEDIM	88
	IF (TEMP2 .LT. .000001) GO TO 255	MOD1205	10
	BPRAD(I,1) = BPRAD(I,1)/TEMP2	ONEDIM	90
	GO TO 258	ONEDIM	91
C		ONEDIM	92
255	CONTINUE	ONEDIM	93
	BPRAD(I,1) = 0.0	ONEDIM	94
C		ONEDIM	95
258	CONTINUE	ONEDIM	96
	PHIBG(I,1) = PHIBG(I,1)/AREACH	ONEDIM	97
	TEMP = (1.0 - PHIBG(I,1))*AREACH	ONEDIM	98
	IF (TEMP .LT. .000001) GO TO 260	MOD1205	11
	UPB(I,1) = UPB(I,1)/TEMP	ONEDIM	100
	XL(I,1) = XL(I,1)/TEMP	ONEDIM	101
	UO(I,1) = UO(I,1)/TEMP	ONEDIM	102
	UI(I,1) = UI(I,1)/TEMP	ONEDIM	103

GO TO 300	ONEDIM	104
C	ONEDIM	105
200 UPB(I,1) = 0.0	ONEDIM	106
XL(I,1) = 0.0	ONEDIM	107
UU(I,1) = 0.0	ONEDIM	108
DI(I,1) = 0.0	ONEDIM	109
C	ONEDIM	110
300 CONTINUE	ONEDIM	111
GO TO 600	ONEDIM	112
C	ONEDIM	113
C	ONEDIM	114
C	ONEDIM	115
C CALCULATIONS FOR CHAP2 TRUE	ONEDIM	116
400 CONTINUE	ONEDIM	117
GO 450 I=1,NGX	ONEDIM	118
140(I,1) = TIGN + 1.0	ONEDIM	119
PIEMP1 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAX	ONEDIM	120
PIEMP2 = (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)	ONEDIM	121
BPRAD(I,1) = BPRAD(I,1)*(1.0 - PHIBP(I,1))*AREAX +	ONEDIM	122
BPRAD(I,2)*(1.0 - PHIBP(I,2))*AREAR(I)	ONEDIM	123
PHIBP(I,1) = (PHIBP(I,1)*AREAX + PHIBP(I,2)*AREAR(I))/	ONEDIM	124
AREAC(I)	ONEDIM	125
TEMP = (1.0 - PHIBP(I,1))*AREAC(I)	ONEDIM	126
IF (TEMP .LT. .000001) GO TO 410	MOD1205	12
BPRAD(I,1) = BPRAD(I,1)/TEMP	ONEDIM	128
GO TO 420	ONEDIM	129
C	ONEDIM	130
410 CONTINUE	ONEDIM	131
BPRAD(I,1) = 0.0	ONEDIM	132
C	ONEDIM	133
420 CONTINUE	ONEDIM	134
TEMP1 = (1.0 - PHIBG(I,1))*AREAX	ONEDIM	135
TEMP2 = (1.0 - PHIBG(I,2))*AREAR(I)	ONEDIM	136
PHIBG(I,1) = (PHIBG(I,1)*AREAX + PHIBG(I,2)*AREAR(I))/	ONEDIM	137
AREAC(I)	ONEDIM	138
TEMP3 = (1.0 - PHIBG(I,1))*AREAC(I)	ONEDIM	139
IF (TEMP3 .LT. .000001) GO TO 430	MOD1205	13
UPB(I,1) = (UPB(I,1)*TEMP1 + UPB(I,2)*TEMP2)/TEMP3	ONEDIM	141
XL(I,1) = (XL(I,1)*TEMP1 + XL(I,2)*TEMP2)/TEMP3	ONEDIM	142
UU(I,1) = (UU(I,1)*TEMP1 + UU(I,2)*TEMP2)/TEMP3	ONEDIM	143
DI(I,1) = (DI(I,1)*TEMP1 + DI(I,2)*TEMP2)/TEMP3	ONEDIM	144
IF (XL(I,1) .LE. 0. .OR. UU(I,1) .LE. 0. .OR. DI(I,1) .LE.	MOD1205	14
* 0.) PHIBG(I,1) = 1.0	MOD1205	15
GO TO 440	ONEDIM	145
430 UPB(I,1) = 0.0	ONEDIM	146
XL(I,1) = 0.0	ONEDIM	147
UU(I,1) = 0.0	ONEDIM	148
DI(I,1) = 0.0	ONEDIM	149
IF (XL(I,1) .LE. 0. .OR. UU(I,1) .LE. 0. .OR. DI(I,1) .LE.	MOD1205	16
* 0.) PHIBG(I,1) = 1.0	MOD1205	17
C	ONEDIM	150
440 CONTINUE	ONEDIM	151
PIEMP3 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAC(I)	ONEDIM	152
C	ONEDIM	153
C***PIEMP3 SHOULD NEVER BE 0.0	ONEDIM	154
RHUSAV = (RHUGB(I,1)*PIEMP1 + RHUGB(I,2)*PIEMP2)/PIEMP3	ONEDIM	155
RHB(I,1) = (RHB(I,1)*PIEMP1 + RHB(I,2)*PIEMP2)/PIEMP3	ONEDIM	156

	HBG(I,2)*PIEMP2*RHOBG(I,2))/(PIEMP3*RHOSAV)	ONEDIM	157
	UBG(I,1) = (UBG(I,1)*PIEMP1+RHOBG(I,1) +	ONEDIM	158
	UBG(I,2)*PIEMP2*RHOBG(I,2))/(PIEMP3*RHOSAV)	ONEDIM	159
	RHOBG(I,1) = RHOSAV	ONEDIM	160
C		ONEDIM	161
C	CALL GSPROP TO UPDATE PCM	ONEDIM	162
	CALL GSPROP(KO,KRO,K,CVO,CVH,CV,FCH(I,1),HBG(I,1),TDOM,	ONEDIM	163
	RHOBG(I,1),UBG(I,1),0.0,GAM(CP,2)	ONEDIM	164
450	CONTINUE	ONEDIM	165
	GO TO 600	ONEDIM	166
C		ONEDIM	167
C		ONEDIM	168
C		ONEDIM	169
C	CALCULATIONS FOR CHAM3 TRUE	ONEDIM	170
500	CONTINUE	ONEDIM	171
	DO 550 I=1,NGX	ONEDIM	172
	TL(I,1) = TIGN + 1.0	ONEDIM	173
C		ONEDIM	174
C	SAVE TOTAL POROSITY TIMES AREA	ONEDIM	175
	PIEMP1 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAX	ONEDIM	176
	PIEMP2 = (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)	ONEDIM	177
	PIEMP3 = AREAGP(I)	ONEDIM	178
C		ONEDIM	179
C	NOTE THAT (1.0 - PHIBP(I,3))*AREAGP(I) IS 0.0	ONEDIM	180
	BPRAD(I,1) = BPRAD(I,1)*(1.0 - PHIBP(I,1))*AREAX +	ONEDIM	181
	BPRAD(I,2)*(1.0 - PHIBP(I,2))*AREAR(I)	ONEDIM	182
	PHIBP(I,1) = (PHIBP(I,1)*AREAX + PHIBP(I,2)*AREAR(I) +	ONEDIM	183
	PHIBP(I,3)*AREAGP(I))/AREAC(I)	ONEDIM	184
	TEMP = (1.0 - PHIBP(I,1))*AREAC(I)	ONEDIM	185
	IF(TEMP .LT. .000001) GO TO 510	MOD1205	186
	BPRAD(I,1) = BPRAD(I,1)/TEMP	ONEDIM	187
	GO TO 520	ONEDIM	188
C		ONEDIM	189
510	CONTINUE	ONEDIM	190
	BPRAD(I,1) = 0.0	ONEDIM	191
C		ONEDIM	192
520	CONTINUE	ONEDIM	193
	TEMP1 = (1.0 - PHIBG(I,1))*AREAX	ONEDIM	194
	TEMP2 = (1.0 - PHIBG(I,2))*AREAR(I)	ONEDIM	195
C		ONEDIM	196
C	NOTE THAT (1.0 - PHIBG(I,3))*AREAGP(I) IS 0.0	ONEDIM	197
	PHIBG(I,1) = (PHIBG(I,1)*AREAX + PHIBG(I,2)*AREAR(I) +	ONEDIM	198
	PHIBG(I,3)*AREAGP(I))/AREAC(I)	ONEDIM	199
	TEMP4 = (1.0 - PHIBG(I,1))*AREAC(I)	ONEDIM	200
	IF(TEMP4 .LT. .000001) GO TO 530	MOD1205	201
	UPB(I,1) = (UPB(I,1)*TEMP1 + UPB(I,2)*TEMP2)/TEMP4	ONEDIM	202
	XL(I,1) = (XL(I,1)*TEMP1 + XL(I,2)*TEMP2)/TEMP4	ONEDIM	203
	DU(I,1) = (DU(I,1)*TEMP1 + DU(I,2)*TEMP2)/TEMP4	ONEDIM	204
	DI(I,1) = (DI(I,1)*TEMP1 + DI(I,2)*TEMP2)/TEMP4	ONEDIM	205
	GO TO 540	ONEDIM	206
530	UPB(I,1) = 0.0	ONEDIM	207
	XL(I,1) = 0.0	ONEDIM	208
	DU(I,1) = 0.0	ONEDIM	209
	DI(I,1) = 0.0	ONEDIM	210
C		ONEDIM	211
540	CONTINUE	ONEDIM	212
	PIEMP4 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAC(I)	ONEDIM	213

C		ONEDIM	214
C****PTEMP4 SHOULD NOT BE 0.0		ONEDIM	215
	RHUSAV = (RHOBG(1,1)*PTEMP1 + RHOBG(1,2)*PTEMP2 +	ONEDIM	216
1	RHOBG(1,3)*PTEMP3)/PTEMP4	ONEDIM	217
	HBG(I,1) = (HBG(I,1)*PTEMP1 + RHOBG(I,1) +	ONEDIM	218
1	HBG(I,2)*PTEMP2 + RHOBG(I,2) + FOG(I,3)*PTEMP3 + RHOBG(I,3))/	ONEDIM	219
1	(PTEMP4 + RHUSAV)	ONEDIM	220
	UBG(I,1) = (UBG(I,1)*PTEMP1 + RHOBG(I,1) +	ONEDIM	221
1	UBG(I,2)*PTEMP2 + RHOBG(I,2) + UBG(I,3)*PTEMP3 + RHOBG(I,3))/	ONEDIM	222
1	(PTEMP4 + RHUSAV)	ONEDIM	223
	RHOBG(I,1) = RHUSAV	ONEDIM	224
C		ONEDIM	225
C	CALL GSPROP TO UPDATE PRESSURE	ONEDIM	226
	CALL GSPROP(RD,RHO,R,CV0,CVH,CV,FCH(I,1),HBG(I,1),TDUM,	ONEDIM	227
1	RHOBG(I,1),UBG(I,1),0.0,GAM,Cr,2)	ONEDIM	228
550	CONTINUE	ONEDIM	229
C		ONEDIM	230
C*****		ONEDIM	231
C	CLEAR ARRAY VBG AND OTHER CHAMBER ARRAYS AT GRID NOT ON THE AXIS	ONEDIM	232
C*****		ONEDIM	233
C		ONEDIM	234
600	CONTINUE	ONEDIM	235
	CALL CLEAR(VBG(1,1),VBG(60,5))	ONEDIM	236
	CALL CLEAR(RHOBG(1,2),RHOBG(60,5))	ONEDIM	237
	CALL CLEAR(PHIBG(1,2),PHIBG(60,5))	ONEDIM	238
	CALL CLEAR(HBG(1,2),HBG(60,5))	ONEDIM	239
	CALL CLEAR(UBG(1,2),UBG(60,5))	ONEDIM	240
	CALL CLEAR(UFB(1,2),UFB(60,5))	ONEDIM	241
	CALL CLEAR(PCH(1,2),PCH(60,5))	ONEDIM	242
	CALL CLEAR(TBG(1,2),TBG(60,5))	ONEDIM	243
	CALL CLEAR(XL(1,2),XL(60,5))	ONEDIM	244
	CALL CLEAR(DV(1,2),DV(60,5))	ONEDIM	245
	CALL CLEAR(DI(1,2),DI(60,5))	ONEDIM	246
	CALL CLEAR(PHIBP(1,2),PHIBP(60,5))	ONEDIM	247
	CALL CLEAR(BPRAD(1,2),BPRAD(60,5))	ONEDIM	248
C		ONEDIM	249
C		ONEDIM	250
C		ONEDIM	251
	NGK = 1	ONEDIM	252
C		ONEDIM	253
	RETURN	ONEDIM	254
C		ONEDIM	255
	END	ONEDIM	256

	SUBROUTINE NEWDX	NEWDX	2
C		NEWDX	3
C	SUBROUTINE NEWDX IS CALLED WHEN THE BARREL GETS A 21ST GRID.	NEWDX	4
C	THE GRID SIZE DX IS DOUBLED, THE BARREL IS CLT DOWN TO 11 GRIDS.	NEWDX	5
C	AND THE NUMBER OF CHAMBER GRIDS IS HALVED.	NEWDX	6
C	A NEW TIME INTERVAL IS ALSO CALCULATED.	NEWDX	7
C		NEWDX	8
	COMMON/BAKRL2/BURLA,XP,VP,BORER,BORERH,BORERD,DT2BU,DTUS9,XLBAR	NEWDX	9
	COMMON/CAKEAS/ARROW1,ARROW2,ARROW3,ARIOT	MOD0301	120
	COMMON/GR1DXX/UXPRIM	MOD0301	121
	COMMON/CHAM/IX,IX,XB,NGX,NBX,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAMIX	2
1	AREACH,AREAL(60),IGN1,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
2	AREAK(60),AREALX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAMIX	4
3	AREAH2,DIAMB1,BELBEG,BELBEG,IPS1,IPS2,KADFS,BPIGN	CHAMIX	5
	COMMON/CLOCK/TIME,DELT	NEWDX	11
	COMMON/EQNS/D1UX,T2UR,T2UX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
1	UX,UR,NX,GJ,TWOT,HBP	EQNS	3
	COMMON/GRAIN/ XL(60,5), DO(60,5), DI(60,5), FN,	GRAIN	2
1	XLTDI(60,5), DOTDI(60,5), DITDI(60,5), XLO, DOO, DIO,	GRAIN	3
2	XLH(100), LOB(100), DIB(100), UXLB(100), LOOB(100), UDIB(100)	GRAIN	4
	COMMON/HOLEA/RACHOL(85),NRROWH,NHOLES(85),XCL(85),AREAH(60)	NEWDX	14
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QLCS,RHOF,PHIU,TF,CA,RHU0,	INPUTS	2
1	H0,P0,U0,GTRHOP,HV,UM,IIGNBP,QBCONS,TOTM	INPUTS	3
	COMMON/MOCON/CON1,CON2,CON3,CON4,CON5,AREAPE,Z0,WOB,XUB,ROTK,	NEWDX	16
1	FUMAX,CUN6,XINT,XLU,PINT,PLO	MOD1111	51
	COMMON/P/1PRINT,MOUCH,MOUGR,PR11,IDEBUG(35)	P	2
	COMMON/BAG/PHIEG(60,5), RHORG(60,5), RBG(60,5), UBG(60,5),	BAG	2
1	VHG(60,5), UPB(60,5), PCH(60,5), T2C(60,5),	BAG	3
2	UUMIG(60), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),	BAG	4
3	PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
4	VBGTD(60,5), TBG(60,5), DOTMBG(60), DOTMP(60,5), PHIRP(60,5),	BAG	6
5	PHIPID(60,5), T2R(60), TBP(60,5)	BAG	7
	COMMON/BAHRL/ PHI(100), RHUG(100), HG(100), UG(100), UP(100),	BARRL	2
1	PG(100), TG(100), PMDUT(100), GL(100), UCRAG(100), FRICT(100),	BARRL	3
2	UCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),	BARRL	4
3	AMASS(100), AMOM(100), AENER(100), LAMASS(100), LAMOM(100),	BARRL	5
4	LAENER(100)	BARRL	6
	LOGICAL IGN1,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL PR11,IDEBUG	PLOG	2
	DATA GRAV/32.16/	NEWDX	23
C		NEWDX	24
	IX = 11	NEWDX	25
C		NEWDX	26
C	PUT THE BARREL ARRAY VALUES AT THE UUC-NUMBERED GRIDS INTO GRIDS	NEWDX	27
C	1 THROUGH 11.	NEWDX	28
	J=1	NEWDX	29
	DO 10 I=2,11	NEWDX	30
	J=J+2	NEWDX	31
	UPHI(I) = UPHI(J)	NEWDX	32
	URHUG(I) = URHUG(J)	NEWDX	33
	UUG(I) = UUG(J)	NEWDX	34
	UHG(I) = UHG(J)	NEWDX	35
	UUP(I) = UUP(J)	NEWDX	36
	UDRAG(I)=UDRAG(J)	NEWDX	37
	UXLB(I) = UXLB(J)	NEWDX	38
	UDOB(I) = UDOB(J)	NEWDX	39
	UDIB(I) = UDIB(J)	NEWDX	40

UAMASS(I) = UAMASS(J)	NEWDX	41
UAMOM(I) = UAMOM(J)	NEWDX	42
UAENER(I) = UAENER(J)	NEWDX	43
10 CONTINUE	NEWDX	44
C	NEWDX	45
C REDUCE THE NUMBER OF GRIDS IN THE CHAMBER	NEWDX	46
C PUT THE TOTAL HOLE AREA INTO THE NEW GRIDS.	NEWDX	47
IF(NGX.EQ.1) GO TO 50	NEWDX	48
AREAH(1) = AREAH(1) + AREAH(2)	NEWDX	49
AREAH2 = AREAH2 + AREAH2	NEWDX	50
NGX = (NGX + 1)/2	NEWDX	51
NGP1 = NGX + 1	NEWDX	52
C	NEWDX	53
C PUT THE CHAMBER ARRAY VALUES AT THE OLD-NUMBERED GRIDS INTO THE	NEWDX	54
C NEW CHAMBER GRIDS.	NEWDX	55
IF(NGX.EQ.1) GO TO 50	NEWDX	56
J=1	NEWDX	57
DO 40 I=2,NGX	NEWDX	58
J=J+2	NEWDX	59
AREAH(I) = AREAH(J) + AREAH(J+1)	NEWDX	60
DO 30 K=1,NGR	NEWDX	61
PHIBTD(I,K) = PHIBTD(J,K)	NEWDX	62
RHOBTD(I,K) = RHOBTD(J,K)	NEWDX	63
UBGTD(I,K) = UBGTD(J,K)	NEWDX	64
VBGTD(I,K) = VBGTD(J,K)	NEWDX	65
HBGTD(I,K) = HBGTD(J,K)	NEWDX	66
UPBTD(I,K) = UPBTD(J,K)	NEWDX	67
XLTD(I,K) = XLTD(J,K)	NEWDX	68
DOTTD(I,K) = DOTTD(J,K)	NEWDX	69
LITD(I,K) = LITD(J,K)	NEWDX	70
30 CONTINUE	NEWDX	71
AREAK(I) = AREAK(J)	NEWDX	72
AREAC(I) = AREAC(J)	NEWDX	73
IF(CHAM3) AREAGP(I) = AREAGP(J)	MOD0301	122
40 CONTINUE	NEWDX	74
C PUT THE VOLUME LOST IN THE BARREL INTO THE CHAMBER.	MOD0301	123
VLOST = 0.5*DX*HOLEA	MOD0301	124
IF(UNED) GO TO 46	MOD0301	125
IF(.NOT. CHAM2) GO TO 44	MOD0301	126
SUM = AREAR(NGX) + AREAA	MOD0301	127
ARROW2 = ARROW2 + (AREAK(NGX)/SUM)*VLOST	MOD0301	128
ARROW1 = ARROW1 + (AREAA/SUM)*VLOST	MOD0301	129
AKTUT = ARROW1 + ARROW2	MOD0301	130
GO TO 50	MOD0301	131
C	MOD0301	132
44 CONTINUE	MOD0301	133
IF(.NOT. CHAM3) GO TO 50	MOD0301	134
SUM = AREAR(NGX) + AREAGP(NGX) + AREAA	MOD0301	135
ARROW1 = ARROW1 + (AREAA/SUM)*VLOST	MOD0301	136
ARROW2 = ARROW2 + (AREAK(NGX)/SUM)*VLOST	MOD0301	137
ARROW3 = ARROW3 + (AREAGP(NGX)/SUM)*VLOST	MOD0301	138
AKTUT = ARROW1 + ARROW2 + ARROW3	MOD0301	139
GO TO 50	MOD0301	140
C	MOD0301	141
46 CONTINUE	MOD0301	142
AKTUT = AKTUT + VLOST	MOD0301	143
C	MOD0301	144

C		MOD0301	145
C		NEWDX	75
C	CHANGE DX, DELT, AND CONSTANTS DEPENDING ON THEM.	NEWDX	76
50	CONTINUE	NEWDX	77
	DXPRIM = DXPRIM + DX	MOD0301	146
	DX = 2.0*DX	NEWDX	78
	DELT = 2.0*DELT	NEWDX	79
	FWDDI = 2.0*DELT	NEWDX	80
	DTDX = DELT/UX	NEWDX	81
	DTDR = DELT/UR	NEWDX	82
	DTDR = 0.5*DTDR	NEWDX	83
	DTDX = 0.5*DTDX	NEWDX	84
	FWOTDR = 2.0*DTDR	NEWDX	85
C		NEWDX	86
C		NEWDX	87
	CONS = 0.5*DELT*DELT	NEWDX	88
	DTDSG = DELT*BORED*BORED	NEWDX	89
	DTZBG = -0.5*DELT/BORED	NEWDX	90
	GTNRCP = GRAV*DELT/RHCP	NEWDX	91
C		MOD0301	147
C	CALCULATE THE CHAMBER VOLUME. IF I ₁ HAS CHANGED ADJUST THE AREAS.	MOD0301	148
	IF (I ₁ EQ. 1) GO TO 150	MOD0301	149
	ARR1 = (FLOAT(NGX) - 0.5)*DX*AREAA	MOD0301	150
C		MOD0301	151
	ARR2 = AREA(1)*DX*0.5	MOD0301	152
	IF (NGX .EQ. 1) GO TO 80	MOD0301	153
	DO 70 I = 2,NGX	MOD0301	154
	ARR2 = ARR2 + AREA(I)*DX	MOD0301	155
70	CONTINUE	MOD0301	156
C		MOD0301	157
80	CONTINUE	MOD0301	158
	IF (CHAM2) GO TO 100	MOD0301	159
C		MOD0301	160
	ARR3 = AREA(1)*DX*0.5	MOD0301	161
	IF (NGX .EQ. 1) GO TO 100	MOD0301	162
	DO 90 I = 2,NGX	MOD0301	163
	ARR3 = ARR3 + AREA(I)*DX	MOD0301	164
90	CONTINUE	MOD0301	165
C		MOD0301	166
100	CONTINUE	MOD0301	167
C		MOD0301	168
C	ADJUST AREAA AND ARR1 AREAC	MOD0301	169
	ADJUST = ARRCW1/ARR1	MOD0301	170
	AREAA = AREAA*ADJUST	MOD0301	171
	DO 105 I = 1,NGX	MOD0301	172
	AREAC(I) = AREAC(I)*ADJUST	MOD0301	173
105	CONTINUE	MOD0301	174
C		MOD0301	175
C	ADJUST ARR2 AREAR	MOD0301	176
	ADJUST = ARRCW2/ARR2	MOD0301	177
	DO 110 I = 1,NGX	MOD0301	178
	AREAR(I) = AREAR(I)*ADJUST	MOD0301	179
110	CONTINUE	MOD0301	180
	IF (CHAM2) GO TO 200	MOD0301	181
C		MOD0301	182
C	ADJUST ARR3 AREAGP	MOD0301	183
	ADJUST = ARRCW3/ARR3	MOD0301	184

DO 100 I = 1,NGX	MOD0301	185
AREAGP(I) = AREAGP(I)*ADJUST	MOD0301	186
120 CONTINUE	MOD0301	187
GO TO 200	MOD0301	188
C	MOD0301	189
C	MOD0301	190
C LOGIC WHEN ONED IS TRUE	MOD0301	191
130 CONTINUE	MOD0301	192
ART = AREAL(1)*LX*0.5	MOD0301	193
IF(INGX.EQ.1) GO TO 170	MOD0301	194
DO 100 I = 2,NGX	MOD0301	195
ART = ART + AREAL(I)*LX	MOD0301	196
140 CONTINUE	MOD0301	197
C	MOD0301	198
170 CONTINUE	MOD0301	199
ADJUST = ARTOT/ART	MOD0301	200
DO 100 I = 1,NGX	MOD0301	201
AREAL(I) = AREAL(I)*ADJUST	MOD0301	202
150 CONTINUE	MOD0301	203
C	MOD0301	204
C	MOD0301	205
200 CONTINUE	MOD0301	206
C	NEWDX	92
C FIX AREAS AND VOLUMES	NEWDX	93
IF(.NOT.CHAM1) GO TO 210	MOD0301	207
UVAXIS = 3.141593*UR*UR*UX	NEWDX	95
UVAXIS = 1.25*UVAXIS	NEWDX	96
GO TO 220	MOD0301	208
C	NEWDX	98
210 CONTINUE	MOD0301	209
AREAG(2) = AREAG(NGX)	NEWDX	100
IF(CHAM3) AREAG(3) = AREAG(NGX)	NEWDX	101
AREACH = AREAG(1) + AREAG(2)	NEWDX	102
IF(CHAM3) AREACH = AREACH + AREAG(3)	NEWDX	103
UVAXIS = AREACH*UX	NEWDX	104
C	NEWDX	105
220 CONTINUE	MOD0301	210
IF(ONED) AREAC(NGP1) = BUREA	NEWDX	107
AREAR(NGP1) = BUREA - AREACH	NEWDX	108
IF(CHAM3) AREAR(NGP1) = AREAR(NGP1) - AREAG(NGP1)	NEWDX	109
IF(IDEBUG(12)) WRITE(6,2000) IPRIAT,NGX	NEWDX	110
NAMLIST/NEWCHK,UX,DELT,IWOT,UTDX,UTLR,T2DF,T2DX,TWOTDR,	NEWDX	111
CON5,UTDSG,UT2BU,GTTRUP,UVAXIS,CVAXIS,T,AREACH	NEWDX	112
IF(IDEBUG(13)) WRITE(6,NEWCHK)	NEWDX	113
IF(.NOT. IDEBUG(14)) RETURN	NEWDX	114
WRITE(6,2006)	NEWDX	115
WRITE(6,2003) (AREAGP(I),I=1,NGP1)	NEWDX	116
WRITE(6,2002)	NEWDX	117
WRITE(6,2003) (AREAR(I),I=1,NGP1)	NEWDX	118
WRITE(6,2004)	NEWDX	119
WRITE(6,2003) (AREAL(I),I=1,NGP1)	NEWDX	120
WRITE(6,2005) (AREAG(I),I=1,NGX)	NEWDX	121
2000 FORMAT(11 NEWDX CALLED, IPRIAT =',15,', NGX =',15)	NEWDX	122
2002 FORMAT(177, ' ARRAY AREAR',77)	NEWDX	123
2003 FORMAT(19X,10F11.7,7)	NEWDX	124
2004 FORMAT(177, ' ARRAY AREAL',77)	NEWDX	125
2005 FORMAT(177, ' ARRAY AREAG',77,20X,5F11.7)	NEWDX	126

2000 FORMAT(///,'* ARRAY AREAGP',//)
RETURN
END

NEWDX 127
NEWDX 128
NEWDX 129

SUBROUTINE AXIT2	AXIT2	2
C SUBROUTINE AXIT2 IS CALLED FOR GRICS IN THE 2ND RADIAL ROW WHEN	AXIT2	3
C THE CHAMBER HAS 2 SEPARATE ONE DIMENSIONAL RCWS.	AXIT2	4
COMMON/AVOUT/RHO(1),PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	AXIT2	5
1 UIDI,VBCAVAL,VIDI	AXIT2	6
COMMON/CHAM/1,J,XR,R0,NGX,NGR,IBLGB,IENDB,IPATH(60,5),AREAG(5),	CHAM1	2
2 AREACH,AREAC(60),IGN1,UNED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,LOPGAP,AREAGP(60),UAVG,	CHAM1	4
4 AREAR2,DIAMBT,BELIND,BELBFG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLOCK/TIME,DELT	AXIT2	8
COMMON/EQNS/DTDX,TZDR,TZDX,TWOIDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
5 DX,OR,IX,G0,TWOIDT,HBP	EQNS	3
COMMON/GASCOR/R0,RK0,CV0,CVH	AXIT2	10
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCH(60,5),IZC(60,5),	BAG	3
2 DOUTIG(60),GBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBTD(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60,5),CCTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIBPL(60,5),TZK(60),TBP(60,5)	BAG	7
LOGICAL IGN1,UNED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
DATA GRAV/32.16/	AXIT2	13
C	AXIT2	14
C	AXIT2	15
CALL GSRPOP(R0,RK0,R,CV0,CVH,CV,PCN(1,J),HBG(1,J),TDOM,	AXIT2	16
2 RHOBG(1,J),UBG(1,J),VBG(1,J),GAR,CP,2)	AXIT2	17
BUGGER = (GAR - 1.0)/TWOGJ	AXIT2	18
IP1 = 1+1	AXIT2	19
IM1 = 1 - 1	AXIT2	20
C	AXIT2	21
C	AXIT2	22
C IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST	AXIT2	23
C POROSITY OF THE PROPELLANT.	AXIT2	24
F5 = PHIBG(1,J)	AXIT2	25
A5 = AREAR(1)	AXIT2	26
G2 = RHOBG(IM1,J)	AXIT2	27
G4 = RHOBG(IP1,J)	AXIT2	28
G5 = RHOBG(1,J)	AXIT2	29
E2 = G2*UBG(IM1,J)	AXIT2	30
E4 = G4*UBG(IP1,J)	AXIT2	31
E5 = G5*UBG(1,J)	AXIT2	32
H2 = PHIBG(IM1,J)*AREAR(IM1)*E2	AXIT2	33
H4 = PHIBG(IP1,J)*AREAR(IP1)*E4	AXIT2	34
G2 = G2 * HBG(IM1,J)	AXIT2	35
G4 = G4 * HBG(IP1,J)	AXIT2	36
G5 = G5 * HBG(1,J)	AXIT2	37
G2 = BUGGER * UBG(IM1,J) * E2	AXIT2	38
G4 = BUGGER * UBG(IP1,J) * E4	AXIT2	39
G5 = BUGGER * UBG(1,J) * E5	AXIT2	40
VOL = AREAR(1)*LX	AXIT2	41
C	AXIT2	42
PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25	AXIT2	43
RHCAVE = (G2 + G5 + G5 + G4)*0.25	AXIT2	44
UBCAVE = (UBG(IM1,J) + UBG(1,J) + UBG(1,J) + UBG(IP1,J))*0.25	AXIT2	45
VBCAVAL = 0.0	AXIT2	46
UPBAVE = (UPB(IM1,J) + UPB(1,J) + UPB(1,J) + UPB(IP1,J))*0.25	AXIT2	47
C	AXIT2	48
PHIBTD = PHIBTD(1,J) + PHIBTD(1,J) - 1.0	AXIT2	49

	RHOTDT = (F5*RHOAVE - T2DX*(H4 - H2)/A5	AXIT2	50
1	- DELT*(DOTMIG(I) - DOTMBG(I))/VOL	AXIT2	51
2	+ DOTMB(I,J) + DOTMP(I,J))/PHITDT	AXIT2	52
	PHIRHO = PHITDT*RHOTDT	AXIT2	53
	TERMI=0.0	AXIT2	54
	IF(DOTMIG(I).GT.0.0)TERMI=1.0	AXIT2	55
	TERM2=0.0	AXIT2	56
	IF(DOTMBG(I),L1,0.0)TERM2=1.0	AXIT2	57
C		AXIT2	58
	UTDT = (F5*(E2 + E3 + E5 + E4)*0.25	AXIT2	59
1	- T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,L))	AXIT2	60
2	+ GRAV*AREAK(I)*PCH(IP1,L))	AXIT2	61
3	- GRAV*AREAK(I)*PCH(IM1,L))/A5	AXIT2	62
4	+ -DELT*(DOTMIG(I)*UBG(I,J)*TERMI-(DOTMBG(I)*UBG(I,J)*TERM2))/VOL	AXIT2	63
	+ DOTMB(I,J)*UPB(I,J))/PHIRHO	AXIT2	64
	IF (ABS(UTDT).LT. 0.1)UTDT = 0.0	AXIT2	65
C		AXIT2	66
	VTDI = 0.0	AXIT2	67
C		AXIT2	68
	IF(PHIAVE.LT.0.999)CALL URAG(XURAG(I,L),.FALSE.,I,J)	AXIT2	69
C		AXIT2	70
	HIGN = MBG(I,1)	AXIT2	71
	IF (DOTMIG(I) .GT. 0.00001) HIGN = MBG(I,2)	AXIT2	72
	MBGN = MBG(I,2)	AXIT2	73
	IF(DOTMBG(I) .GT. 0.00001) MBGN = MBG(I,3)	AXIT2	74
C		AXIT2	75
	MBGTD(I,J) = GAM*(F5*(C2+B2+C5+C5+B5+B5+C4+B4)/(4.0*GAM)	AXIT2	76
1	- T2DX*(H4*MBG(IP1,J) - H2*MBG(IM1,L))/A5	AXIT2	77
2	+ DELT*(DOTMBG(I)*MBGN - DOTMIG(I)*HIGN)/VOL	AXIT2	78
3	- QBAG(I,J) + DOTMB(I,J)*HMB + DOTMP(I,J)*HBP	AXIT2	79
4	-XURAG(I,J)*UPB(I,J)*DELT/788.	AXIT2	80
	- BUGGER*PHIRHO*UTDT*UTDT/GAM)/PHIRHO	AXIT2	81
C		AXIT2	82
	RHOBTD(I,J) = RHOTDT	AXIT2	83
	URGTD(I,J) = UTDT	AXIT2	84
	VBGTD(I,J) = VTDI	AXIT2	85
	RETURN	AXIT2	86
	END	AXIT2	87

	SUBROUTINE AXIT3	AXIT3	2
C	SUBROUTINE AXIT3 IS CALLED FOR GRIDS IN THE 2ND RADIAL ROW WHEN	AXIT3	3
C	THE CHAMBER HAS 3 SEPARATE ONE DIMENSIONAL ROWS.	AXIT3	4
	COMMON/AVGDT/KRO(1),PHIRHO,PHIAVE,KRCAVE,UBCAVE,UPBAVE,	AXIT3	5
	1 UDT,VBCAVE,VITDT	AXIT3	6
	COMMON/CHAN/I,J,XB,RB,RGX,NGK,IBLUB,IENDB,IPATH(60,5),AREAG(5),	CHAMI	2
	2 AREACH,AREAC(60),ICWIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMI	3
	3 AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,IOGAP,AREAGP(60),UAVG,	CHAMI	4
	4 AREAH,DIAMH,RELENC,BLEBG,IPS1,IPS2,RAOFS,BPIGN	CHAMI	5
	COMMON/CLOCK/TIPL,DELI	AXIT3	8
	COMMON/EGNS/LTDX,TZK,TZUX,TWOTOP,DICK,HMB,TWOGJ,OVAXIS,OVAXIT,	EGNS	2
	5 DX,DK,IX,OX,TWOT,HEP	EGNS	3
	COMMON/GASCOR/KO,KRG,CVO,CVH	AXIT3	10
	COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
	1 VBG(60,5),UPB(60,5),PCB(60,5),TZC(60,5),	BAG	3
	2 DUTMBG(60),UBAG(60,5),DURAG(60,5),DUTPB(60,5),UPBDT(60,5),	BAG	4
	3 PHIBD(60,5),RHBD(60,5),HBGD(60,5),UBGD(60,5),	BAG	5
	4 VBGD(60,5),IBG(60,5),DUTMBG(60),DUTPB(60,5),PHIBP(60,5),	BAG	6
	5 PHIPD(60,5),TZK(60),TBP(60,5)	BAG	7
	LOGICAL IGNI,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	DATA GRAV/32.167	AXIT3	13
C		AXIT3	14
C	CALL USPROP(KO,KRG,CVO,CVH,CV,PC(1,J),HBE(1,J),TDUM,	AXIT3	15
	1 RHOBG(1,J),UBG(1,J),VRG(1,J),GAP,CP,2)	AXIT3	16
	BUGGER = IGAM = 1.0/1WGGJ	AXIT3	17
C		AXIT3	18
C	IN THIS SUBROUTINE PHIBG REPRESENTS IFL TOTAL POROSITY, NOT JUST	AXIT3	19
C	POROSITY OF THE PROPELLANT.	AXIT3	20
	IP1 = 1+1	AXIT3	21
	IM1 = 1 - 1	AXIT3	22
	F5 = PHIBG(1,J)	AXIT3	23
	A5 = AREAGP(1)	AXIT3	24
	G2 = RHOBG(IP1,J)	AXIT3	25
	G4 = RHOBG(IP1,J)	AXIT3	26
	G5 = RHOBG(1,J)	AXIT3	27
	E2 = G2*UBG(IM1,J)	AXIT3	28
	E4 = G4*UBG(IP1,J)	AXIT3	29
	E5 = G5*UBG(1,J)	AXIT3	30
	H2 = PHIBG(IM1,J)*AREAGP(IM1)*E2	AXIT3	31
	H4 = PHIBG(IP1,J)*AREAGP(IP1)*E4	AXIT3	32
	L2 = G2*HBG(IM1,J)	AXIT3	33
	L4 = G4*HBG(IP1,J)	AXIT3	34
	L5 = G5*HBG(1,J)	AXIT3	35
	G2 = BUGGER*UBG(IM1,J)*E2	AXIT3	36
	G4 = BUGGER*UBG(IP1,J)*E4	AXIT3	37
	G5 = BUGGER*UBG(1,J)*E5	AXIT3	38
C		AXIT3	39
C	VOL = AREAGP(1)*DX	AXIT3	40
C		AXIT3	41
C		AXIT3	42
	PHITD1 = PHIBD(1,J) + PHIPD(1,J) - 1.0	AXIT3	43
	RHITD1 = (F5*(G2 + G5 + G5 + G4)*0.2	AXIT3	44
1	- TZDX*(H4 - H2)/A5	AXIT3	45
2	- DELI*DUTMBG(1)/VOL)/PHITD1	AXIT3	46
C	PHIRHO = PHITD1*KRODT	AXIT3	47
C		AXIT3	48
		AXIT3	49

UTDT = (F5*(L2 + F5 + L5 + E4)*0.25	AXIT3	50
1 - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J))	AXIT3	51
2 + GRAV*A5*PCH(IP1,J) - GRAV*A5*PCH(IM1,J))/A5)/PHIRHO	AXIT3	52
IF(ABS(UTDT) .LT. 0.1) UTDT = 0.0	AXIT3	53
C	AXIT3	54
HBGN = HBG(I,2)	AXIT3	55
IF(DOTHBG(I) .GT. 0.00001) HBGN = HBG(I,3)	AXIT3	56
HBGUT(I,J) = GAM*(F5*(L2+b2+C4+B4+L5+B5+C5+B5)/(4.0*GAM)	AXIT3	57
3 -T2DX*(H4*HBG(IP1,J) - H2*HBG(IM1,J))/A5	AXIT3	58
4 - DEL1*DOTHBG(I)*HBGN/ VOL	AXIT3	59
5 - BUGGER*UTDT*UTDT*PHIRHO/GAM)/PHIRFO	AXIT3	60
C	AXIT3	61
KHOUT(I,J) = KHOT(I)	AXIT3	62
UBOUT(I,J) = UTDT	AXIT3	63
RETURN	AXIT3	64
END	AXIT3	65

	SUBROUTINE BSURA2	BSURA2	2
C		BSURA2	3
C	AREA FACTORS ARE NOT NEEDED IN THIS SUBROUTINE BECAUSE THEY WOULD	BSURA2	4
C	ALL CANCEL.	BSURA2	5
C		BSURA2	6
	COMMON/PROP/LOZ,LEMPM, UPRN	BSURA2	7
	COMMON/AVGDT/RHUTDT,PHIKHU,FIUAVE,RI,CAVE,UBCAVE,UPBAVE.	BSURA2	8
	1 Q1Q1,VGBAVE,V1Q1	BSURA2	9
	COMMON/CHAM/1,J,XB,RB,NGX,NGR,IBEGE,IENDH,IPATH(60,5),AREAG(5).	CHAM1	2
	1 AREACH,AREAL(60),ICN11,ONED,DIAP1,LIAM2,CIS1,DIS2,DIS3,DIS4.	CHAM1	3
	1 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,IUPGAP,AREAGP(60),DAVG.	CHAM1	4
	1 AREAH2,DIAP1,BELEND,BELBEG,IPS1,IFG2,RADFS,BPIGN	CHAM1	5
	COMMON/CLOCK/TIME,DELT	BSURA2	11
	COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QLCBS,RHOF,PHIQ,TF,CA,RHOO.	MOD0925	519
	1 H0,P0,U0,GTRHOP,H0,DA,TIGNBP,ORCONS,TOTM	MOD0925	520
	COMMON/FAILED/THICK1,FH0UP,PCOMP,BICE,XNTUB,FAIL,MFAIL(60).	MOD0925	521
	1 THICK(60),PENET,TYPECL	MOD0925	522
	COMMON/LQNS/DTEX,TZUK,TZUX,TWOTDR,DICK,HMB,TWOGJ,DVAXIS,DVAXIT.	EQNS	2
	1 DX,DX,NX,G0,TWOT,HEP	EQNS	3
	COMMON/GASCON/R0,RR0,CV0,CVH	BSURA2	13
	COMMON/BAG/PHIBG(60,5), RHOBG(60,5), RBG(60,5), UBG(60,5).	BAG	2
	1 VBG(60,5), UFB(60,5), PCH(60,5), T2C(60,5).	BAG	3
	2 DUTMIG(60), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5).	BAG	4
	3 PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5).	BAG	5
	4 VBGTD(60,5), TBG(60,5), DUTMBG(60), DUTMP(60,5), PHIBP(60,5).	BAG	6
	5 PHIPTD(60,5), T2K(60), TBP(60,5)	BAG	7
	LOGICAL IGR1,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
	LOGICAL PENET	MOD0925	523
C		MOD0925	524
	CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(1,J),HBE(I,J),TDUM,	BSURA2	17
	1 RHOBG(1,J),UBG(1,J),VPG(1,J),GAP,CP,2)	BSURA2	18
	BUGGER = (GAP - 1.0)/TDUM	BSURA2	19
	IP1 = 1+1	BSURA2	20
C		BSURA2	21
C	IN THIS SUBROUTINE PHIEG REPRESENTS THE TOTAL POROSITY, NOT JUST	BSURA2	22
C	POROSITY OF THE PROPELLANT.	BSURA2	23
	F4 = PHIBG(IP1,J)	BSURA2	24
	F5 = PHIBG(1,J)	BSURA2	25
	G4 = F4*RHOBG(IP1,J)	BSURA2	26
	G5 = F5*RHOBG(1,J)	BSURA2	27
	E4 = G4*UBG(IP1,J)	BSURA2	28
	E5 = 0.0	BSURA2	29
	C4 = G4*HBG(IP1,J)	BSURA2	30
	C5 = G5*HBG(1,J)	BSURA2	31
	B4 = G4*BUGGER*UBG(IP1,J)**2	BSURA2	32
	B5 = 0.0	BSURA2	33
	DENOM = F4 + F5	BSURA2	34
	PHIAVE = DENOM*0.5	BSURA2	35
	RHUAVE = (G4 + G5)/DENOM	BSURA2	36
	UBGAVE = 0.0	BSURA2	37
	UPBAVE = (UPB(IP1,J) + UPB(1,J))*0.5	BSURA2	38
C		BSURA2	39
	PHITDT = PHIBTD(1,J) + PHIPTD(1,J) - 1.0	BSURA2	40
	RHUTDT=(F5*RHUAVE-DTEX*E4 + DELT + (DUTMIG(1) + DOTMPM)/	BSURA2	41
1	DVAXIS*2.0 + DUTMB(1,J) + DUTIP(1,J))/PHITDT	BSURA2	42
	PHIKHU = PHITDT*RHUTDT	BSURA2	43
C		BSURA2	44

UTDT = 0.0	BSURA2	45
VTDI = 0.0	BSURA2	46
C	BSURA2	47
HIGN = HBG(1,1)	BSURA2	48
C	BSURA2	49
IF (DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)	BSURA2	50
HGBID(1,J) = GAM*(F5*(C4+B4+C5+B5)/(LENOM*GAM)	BSURA2	51
1 - DTUX*(C4*UBG(IP1,J)	BSURA2	52
2 + DELT * ((DOTMIG(I) * HIGN + DOTMP * HBP) * 2.0/	BSURA2	53
3 DVAXIS - UBAG(1,J)) + DOTMB(1,J) * FMB	BSURA2	54
4 + DOTMP(1,J) * HBP)/PHIRHC	BSURA2	55
C	BSURA2	56
UTDT=2.0* DELT * DOTMPM * UPRM/RHCTDI/DVAXIS	BSURA2	57
RHBTU(1,J) = RHDTU1	BSURA2	58
UBGTD(1,J) = UTDT	BSURA2	59
VBGTD(1,J) = VTDI	BSURA2	60
RETURN	BSURA2	61
END	BSURA2	62

SUBROUTINE HSURT2	HSURT2	2
COMMON/AVG/1/RHOTD1,PHIRHO,PHIAVE,RHOAVE,UBCAVE,UPBAVE,	HSURT2	3
1 UDT,VBCAVE,VTOT	HSURT2	4
COMMON/CHAM/1,J,XB,RB,NGX,NGR,IELE,IEEND,IPATH(60,5),AREAG(5),	CHAM1	2
2 AREACH,AREAC(60),IGN1,ONED,DIAM1,CIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
3 AREAR(60),AREAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,	CHAM1	4
4 AREAH2,DIAMB1,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAM1	5
COMMON/CLOCK/TIME,DEL1	HSURT2	6
COMMON/EQNS/DUX,TZK,TZUX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
5 DX,DX,NX,GO,TWOT1,HBP	EQNS	3
COMMON/GASCON/RD,KRO,CV0,CVH	HSURT2	8
COMMON/BAG/PHIBG(60,5),RHOE(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPB(60,5),PCB(60,5),TZC(60,5),	BAG	3
2 DOTMIG(60),QBAG(60,5),XDRAG(60,5),DOTPB(60,5),UPBDT(60,5),	BAG	4
3 PHIBTD(60,5),RHOBT(60,5),HBGTD(60,5),UBGTD(60,5),	BAG	5
4 VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPTD(60,5),TZK(60),TBP(60,5)	BAG	7
LOGICAL IGN1,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
CALL GSPROP(RD,KRO,K,CV0,CVH,CV,PCN(1,J),HBE(1,J),TDUM,	HSURT2	11
1 RHOE(1,J),UBG(1,J),VBC(1,J),GAP,CP,2)	HSURT2	12
BUGGER = (GAP - 1.0)/TWOGJ	HSURT2	13
IP1 = 1+1	HSURT2	14
	HSURT2	15
IN THIS SUBROUTINE PHIBG REPRESENTS IFL TOTAL POROSITY, NOT JUST	HSURT2	16
POROSITY OF THE PROPELLANT.	HSURT2	17
F4 = PHIBG(IP1,J)*AREAR(IP1)	HSURT2	18
F5 = PHIBG(1,J)*AREAR(1)	HSURT2	19
G4 = F4*RHOE(1,J)	HSURT2	20
G5 = F5*RHOE(1,J)	HSURT2	21
E4 = G4*UBG(IP1,J)	HSURT2	22
E5 = 0.0	HSURT2	23
G4 = G4*HBG(IP1,J)	HSURT2	24
G5 = G5*HBG(1,J)	HSURT2	25
H4 = G4*BUGGER*(HBC(IP1,J)**2	HSURT2	26
H5 = 0.0	HSURT2	27
	HSURT2	28
DENOM = F4 + F5	HSURT2	29
PHIAVE = DENOM/(AREAR(IP1) + AREAR(1))	HSURT2	30
RHOAVE = (G4 + G5)/DENOM	HSURT2	31
UBCAVE = 0.0	HSURT2	32
UPBAVE = (UPB(IP1,J) + UPB(1,J))*0.5	HSURT2	33
	HSURT2	34
PHITDI = (PHIBTD(1,J) + PHIPTD(1,J) - 1.0)*AREAR(1)	HSURT2	35
	HSURT2	36
RHOIDI = (F5*RHOAVE - DUX*E4 - DUX*(DOTMIG(1) - DOTMBG(1))	HSURT2	37
1 + (DOTPB(1,J) + DOTMP(1,J))*AREAR(1))/PHITDI	HSURT2	38
	HSURT2	39
PHIRHO = PHITDI*RHOIDI	HSURT2	40
	HSURT2	41
UDT = 0.0	HSURT2	42
VTOT = 0.0	HSURT2	43
	HSURT2	44
HIGN = HBG(1,1)	HSURT2	45
IF(DOTMIG(1).GT.0.00001) HIGN = HBG(1,2)	HSURT2	46
HIGN = HBG(1,2)	HSURT2	47
IF(DOTMBG(1).GT.0.00001) HIGN = HBG(1,3)	HSURT2	48
	HSURT2	49

C			BSURT2	50
		HRGID(I,J) = GAM*(F5*(C4+B4+C5+B5)/(LENOM*GAM)	BSURT2	51
1		= DUX*(C4+UBG(IP1,J) + DCTMG(I)*HIGN - DOTMBG(I)*HBGN)	BSURT2	52
2		+ AREAR(I)*(DOTMB(I,J)*FIB + DCTMP(I,J)*HBP	BSURT2	53
3		= DELT*CBAG(I,J)))/PHINNU	BSURT2	54
		KHODID(I,J) = KHOTDI	BSURT2	55
		URGID(I,J) = UTDI	BSURT2	56
		VBGID(I,J) = VTDI	BSURT2	57
		RETURN	BSURT2	58
		END	BSURT2	59

SUBROUTINE BSURT3	BSURT3	2
COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHCAVE,UBCAVE,UPBAVE,	BSURT3	3
3 UDT,VBCAVE,VTDT	BSURT3	4
COMMON/CHAM/1 J,XB,RB,NGX,NGR,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAM1	2
4 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAM1	3
5 AREAK(60),AREAX,CHAM1,CHAM2,CHAM3,OPGAP,AREAGP(60),DAVG,	CHAM1	4
6 AREANG,DIAM1,BELEND,ELLEG,IPS1,IPS2,RAUFS,BPIGN	CHAM1	5
COMMON/CLKR/TIME,DEL1	BSURT3	6
COMMON/EQNS/DUX,T2DR,T2DX,TWOTDR,DICR,HMB,TWOGJ,DVAXIS,DVAXIT,	EQNS	2
7 DUX,DR,NX,GU,TWOT,HBP	EQNS	3
COMMON/GASCON/KO,KRO,CVO,CVH	BSURT3	6
COMMON/BAG/PHIBG(60,5),RHOBG(60,5),HBG(60,5),UBG(60,5),	BAG	2
1 VBG(60,5),UPH(60,5),PCH(60,5),T2C(60,5),	BAG	3
2 DUTMIG(60),UBAG(60,5),XDRAG(60,5),DOTMBG(60,5),UPBDT(60,5),	BAG	4
3 PHITD(60,5),RHOTD(60,5),HMGD(60,5),UBGD(60,5),	HAG	5
4 VBGD(60,5),TEG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),	BAG	6
5 PHIPD(60,5),T2P(60,5),T3P(60,5)	BAG	7
LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN	CHAMLOG	2
C CALL GSPROP(KO,KRO,K,CVO,CVH,CV,PCH,1,J),HBE(1,J),TDUM,	BSURT3	11
6 RHOBG(1,J),UBG(1,J),VBG(1,J),GAP,CP,2)	BSURT3	12
BUGGER = (GAP - 1.0)/TWOGJ	BSURT3	13
IP1 = 1+1	BSURT3	14
C	BSURT3	15
C IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST	BSURT3	16
C POROSITY OF THE PROPELLANT,	BSURT3	17
F4 = PHIBG(IP1,J)*AREAGP(IP1)	BSURT3	18
F5 = PHIBG(1,J)*AREAGP(1)	BSURT3	19
G4 = F4*RHOBG(IP1,J)	BSURT3	20
G5 = F5*RHOBG(1,J)	BSURT3	21
L4 = G4*UBG(IP1,J)	BSURT3	22
L5 = G5*UBG(1,J)	BSURT3	23
C4 = G4*HBG(IP1,J)	BSURT3	24
C5 = G5*HBG(1,J)	BSURT3	25
U4 = G4*BUGGER*UBG(IP1,J)**2	BSURT3	26
U5 = G5*UBG(1,J)	BSURT3	27
C	BSURT3	28
UENOM = F4 + F5	BSURT3	29
C PHIAVE = UENOM/(AREAGP(IP1) + AREAGP(1))	BSURT3	30
RHOAVE = (G4 + G5)/UENOM	BSURT3	31
C UBCAVE = 0.0	BSURT3	32
C UPBAVE = (UPB(IP1,J) + UPB(1,J))*0.5	BSURT3	33
C	BSURT3	34
PHITDT = (PHITD(1,J) + PHITD(IP1,J) - 1.0)*AREAGP(1)	BSURT3	35
C	BSURT3	36
RHOTDT = (F5*RHOAVE - DUX*E4 - CLDX*DOTMBG(1))/PHITDT	BSURT3	37
C	BSURT3	38
PHIRHO = PHITDT*RHOTDT	BSURT3	39
C	BSURT3	40
UTDT = 0.0	BSURT3	41
C	BSURT3	42
HMBN = HMB(1,2)	BSURT3	43
IF(DUTMBG(1).GT.0.00001) HMBN = HMB(1,3)	BSURT3	44
C	BSURT3	45
HMGTD(1,J) = GAM*(F5*(C4+B4+C5+B5)/(UENOM*GAM)	BSURT3	46
- DUX*(C4*UBG(IP1,J) + DOTMBG(1)*HMBN))/PHIRHO	BSURT3	47
1 RHOTDT(1,J) = RHOTDT	BSURT3	48
	BSURT3	49

UBGTD(1,3) = UGDT
RETURN
END

BSURT3 50
BSURT3 51
BSURT3 52

Line	Code	Statement	Variable	Value
1		SUBROUTINE BNDLYR	BNDLYR	2
2		COMMON/BARREL/ICREA,XP,VP,BORED,HCKEN,BOREDE,DT2BU,DTUSU	BNDLYR	3
3		COMMON/CONS/DTGX,DTGR,DTGX,TWOIDR,D,CH,HMB,TWOGJ,DVAXIS,DVAXIT,	BNDLYR	4
4		DX,DR,NX,GJ,TWOI,HBP	CONS	2
5		COMMON/CASCOR/KO,KRO,CVO,CVH	LUNS	3
6		COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,CLCKS,RHOF,PHIU,TF,CA,RHOO,	BNDLYR	6
7		HU,PG,DU,GRHOP,HM,LP,TIGNBP,QBCONS,TOTM	INPUTS	2
8		COMMON/BARREL/PHI(100),RHOG(100),HG(100),UG(100),UP(100),	INPUTS	3
9		PG(100),TG(100),FNDUT(100),GL(100),UCRAG(100),FRIC(100),	HARRL	2
10		UCONV(100),UOP(100),UPH(100),URHOG(100),UHG(100),UUG(100),	BARRL	3
11		AMASS(100),AMOM(100),AEN(100),LAMASS(100),UAMOM(100),	HARRL	4
12		UAENER(100)	HARRL	5
13		REAL MACHSU	BARRL	6
14		DATA PIUF/.785398/	BNDLYR	9
15	C	SUBROUTINE BNDLYR CALCULATES THE THICKNESSES OF THE BOUNDARY LAYERS	BNDLYR	10
16	C	FORMED AS PROPELLANT GAS FLOWS THROUGH THE BARREL.	BNDLYR	11
17	C	AMASS IS THE AREA AVAILABLE FOR MASS FLOW	BNDLYR	12
18	C	AMOM IS THE AREA OUTSIDE THE MOMENTUM LAYER	BNDLYR	13
19	C	AENER IS THE AREA OUTSIDE THE ENERGY BOUNDARY LAYER	BNDLYR	14
20	C	IT ALSO CALCULATES GL, THE HEAT TRANSFERRED FROM THE GRID TO OR	BNDLYR	15
21	C	FROM THE BARREL.	BNDLYR	16
22	C	ARRAY GL CONTAINS G.J'S AT THE BEGINNING OF EACH TIME INTERVAL.	BNDLYR	17
23	C		BNDLYR	18
24	C		BNDLYR	19
25	C	NM1 = IX - 1	BNDLYR	20
26	C		BNDLYR	21
27	C	SPIX = 0.0	BNDLYR	22
28	C	DO GO I=2,NM1	BNDLYR	23
29	C	U = (UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25	BNDLYR	24
30	C	IF(U.LT. 10.) U = 10.	BNDLYR	25
31	C	USE = U+U	BNDLYR	26
32	C		BNDLYR	27
33	C	STATIC ENTHALPY	BNDLYR	28
34	C	HSTAT = (HG(I)+HG(I)+HG(I-1)+HG(I+1))*0.25-.2E-4*USE	BNDLYR	29
35	C	CALL GSPROP(KO,KRO,K,CVO,CVH,CV,PG(I),HSTAT,TSTAT,RHODUM,	BNDLYR	30
36	C	U,0,U,0,GAM,CF,3)	BNDLYR	31
37	C	GAM1 = GAM - 1.0	BNDLYR	32
38	C	GAM3 = (3.0+GAM - 1.0)/(2.0+GAM1)	BNDLYR	33
39	C		BNDLYR	34
40	C	VISCOSITY	BNDLYR	35
41	C	VIS = C1*ISTAT*SQRT(ISTAT)/(ISTAT + C2)	BNDLYR	36
42	C		BNDLYR	37
43	C	REYNOLDS NUMBER/FOOT	BNDLYR	38
44	C	RE = (RHOG(I)+RHOG(I)+RHOG(I-1)+RHOG(I+1))*0.25*U/VIS	BNDLYR	39
45	C		BNDLYR	40
46	C	FACH NUMBER SQUARE1	BNDLYR	41
47	C	MACHSU = USG/(32.16+GAM*R*ISTAT)	BNDLYR	42
48	C		BNDLYR	43
49	C	FACH NUMBER WEIGHTING FUNCTION	BNDLYR	44
50	C	PRESX = FACHSU*MACHSU/((1.0 + 0.5*GAM1+FACHSU)**GAM3)	BNDLYR	45
51	C	PDX = PRESX*DX	BNDLYR	46
52	C	SPIX = SPIX + PDX	BNDLYR	47
53	C		BNDLYR	48
54	C	EQUIVALENT FLAT PLATE LENGTH	BNDLYR	49
55	C	LAF = SPIX/PRESX	BNDLYR	50
56	C		BNDLYR	51
57	C		BNDLYR	52

C	REYNOLD'S NUMBER	BNDLYR	53
	REX = RE*EXF	BNDLYR	54
	GFUN1 = (1.0 + 0.25*GAM1*MACHSQ)**(-0.7)	BNDLYR	55
	GFUN2 = (1.0 + 2.0*GAM1*MACHSQ)**0.44	BNDLYR	56
	IF (REX .LT. 5.0E6) GO TO 30	BNDLYR	57
	REF = EXF/REX**0.1667	BNDLYR	58
	DELTA = 0.23*REF	BNDLYR	59
C		BNDLYR	60
	IF (DELTA .LT. BOREX) GO TO 20	BNDLYR	61
	ITEMP = 1	BNDLYR	62
	GO TO 150	BNDLYR	63
C		BNDLYR	64
C	DISPLACEMENT BOUNDARY LAYER THICKNESS	BNDLYR	65
20	CONTINUE	BNDLYR	66
	DELSTR = 0.028*GFUN2*REF	BNDLYR	67
C		BNDLYR	68
C	MOMENTUM BOUNDARY LAYER THICKNESS	BNDLYR	69
	THEIA = 0.022*GFUN1*REF	BNDLYR	70
	GO TO 40	BNDLYR	71
C		BNDLYR	72
30	REF = EXF/REX**0.2	BNDLYR	73
	DELTA = 0.37*REF	BNDLYR	74
C		BNDLYR	75
	IF (DELTA .LT. BOREX) GO TO 35	BNDLYR	76
	ITEMP = 1	BNDLYR	77
	GO TO 150	BNDLYR	78
C		BNDLYR	79
35	CONTINUE	BNDLYR	80
	DELSTR = 0.046*GFUN2*REF	BNDLYR	81
	THEIA = 0.036*GFUN1*REF	BNDLYR	82
C		BNDLYR	83
C	ENERGY BOUNDARY LAYER THICKNESS	BNDLYR	84
40	DELST = 1.269*DELSTR/(DELSTR/THEIA - 0.379)	BNDLYR	85
C		BNDLYR	86
C	MASS FLOW AREA	BNDLYR	87
	UAMASS(I) = P1DF*(BORED - 2.0*DELSTR)**2	BNDLYR	88
C		BNDLYR	89
C	MOMENTUM FLOW AREA	BNDLYR	90
	UAPOM(I) = P1DF*(BORED - 2.0*THEIA)**2	BNDLYR	91
C		BNDLYR	92
C	ENERGY FLOW AREA	BNDLYR	93
	UAENER(I) = P1DF*(BORED - 2.0*DELST)**2	BNDLYR	94
C		BNDLYR	95
C	AVERAGE (REFERENCE) ENTHALPY IN BOUNDARY LAYER	BNDLYR	96
	HSTAR = 0.5*(HG(I) + HW) - 6.12E-6*USQ	BNDLYR	97
	CALL GSPROP(RH,RRH,R,CV0,CVH,CV,PC(I),HSTAR,TSTAR,RHOSIR,	BNDLYR	98
	* 0.0,0.0,GAM,CP,3)	BNDLYR	99
	CALL GSPROP(RH,RRH,R,CV0,CVH,CV,PC(I),HSTAR,TSTAR,RHOSIR,	BNDLYR	100
	* T00M,RHOG(I),0.0,0.0,GAM,CP,2)	BNDLYR	101
	GAM1 = GAM - 1.0	BNDLYR	102
	MACHSQ = USQ/(32.16*GAM*R*TSTAR)	BNDLYR	103
	VISTR = (1*TSTAR*SQR(1/TSTAR)/(TSTAR + C2)	BNDLYR	104
	RESTRX = RHOSIR*U/VISTR*EXF	BNDLYR	105
C		BNDLYR	106
C	COMPRESSIBLE SKIN FRICTION COEFFICIENT	BNDLYR	107
	CFR = (1.0 + GAM1*GAM1*MACHSQ)**(-0.6)	BNDLYR	108
	CUU1 = 0.0366*VISTR/EXF * RESTRX**0.8 * (HG(I) - HW)*CFR	BNDLYR	109

C		BNDLYR	110
C	HEAT FLUX	BNDLYR	111
	QL(1) = 3.141593*BORED*DX*QDOT	BNDLYR	112
C		BNDLYR	113
C	CONTINUE	BNDLYR	114
	GO TO 200	BNDLYR	115
C		BNDLYR	116
C		BNDLYR	117
C		BNDLYR	118
C		BNDLYR	119
100	CONTINUE	BNDLYR	120
	DO 110 I=2,NP1	BNDLYR	121
	UAMASS(1) = BUREA	BNDLYR	122
	UAMOM(1) = BUREA	BNDLYR	123
	UAEKER(1) = BUREA	BNDLYR	124
110	CONTINUE	BNDLYR	125
	GO TO 200	BNDLYR	126
C		BNDLYR	127
C		BNDLYR	128
150	CONTINUE	BNDLYR	129
	I = ITEMP	BNDLYR	130
	I1 = I - 1	BNDLYR	131
	UAMASS(I) = UAMASS(I1)	BNDLYR	132
	UAMOM(I) = UAMOM(I1)	BNDLYR	133
	UAEKER(I) = UAEKER(I1)	BNDLYR	134
C		BNDLYR	135
	KED = KE*BORED	BNDLYR	136
	EPSLON = 0.0005*ALOG(KED) - 0.00556	BNDLYR	137
	IF(EPSLON .LT. 0.0001) EPSLON = 0.0	BNDLYR	138
C		BNDLYR	139
	LAMBDA = 0.3164/KED**0.25 + EPSLON	BNDLYR	140
C		BNDLYR	141
	RHOU = (RHOG(I) + RHOG(I) + RHOG(I-1) + RHOG(I+1))*0.25*U	BNDLYR	142
	LAMU = LAMBDA*U	BNDLYR	143
C		BNDLYR	144
C	BURED8 IS BURED/8.	BNDLYR	145
	FRAC1(I) = LAMU*RHOU*BURED8	BNDLYR	146
C		BNDLYR	147
C	DT2BU IS -0.5*DELT/BURED	BNDLYR	148
C	DT1SG IS DELT*BURED*BURED	BNDLYR	149
	GCORV(I) = P10F*((HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 - HW)	BNDLYR	150
	1 * (1.0 - EXP(LAMU*DT2BU))*RHOU*DTDSQ	BNDLYR	151
C		BNDLYR	152
C		BNDLYR	153
	I2 = I+1	BNDLYR	154
C		BNDLYR	155
DO 160 I=2,NP1		BNDLYR	156
	UAMASS(I) = UAMASS(I1)	BNDLYR	157
	UAMOM(I) = UAMOM(I1)	BNDLYR	158
	UAEKER(I) = UAEKER(I1)	BNDLYR	159
C		BNDLYR	160
	U=(UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25	BNDLYR	161
	IF(U .LT. 10.) U = 10.	BNDLYR	162
	USG = U*U	BNDLYR	163
	HSTAT = (HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 - .2E-4*USG	BNDLYR	164
	CALL GSPROP(RD,RRC,R,CV0,CVH,CV,P0(I),HSTAT,ISTAT,RHODUM,	BNDLYR	165
		BNDLYR	166

	0.0*0.0*0.0*AM(CP,3)	BNDLYR	167
	VIS = (C1*ISTAT*SQRT(ISTAT)/(TS1A1 + C2)	BNDLYR	168
	RHOU = (RHOG(I) + RHOG(I) + RHOG(I-1) + RHOG(I+1))*0.25*U	BNDLYR	169
	RED = RHOU/VIS*BORED	BNDLYR	170
	EPSLON = 0.0005*ALOG(RED) - 0.0056	BNDLYR	171
	IF(EPSLON .LT. 0.0001) EPSLON = 0.0	BNDLYR	172
	LAMDA = 0.3164/RED**0.25 + EPSLON	BNDLYR	173
	LAMU = LAMDA*U	BNDLYR	174
	FRACT(I) = LAMU*RHOU*BORED8	BNDLYR	175
	GLONV(I) = FIDF*((HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 - HW)	BNDLYR	176
	* (1.0 - EXP(LAMU*UT2BD))*RHOU*OTDSQ	BNDLYR	177
160	CONTINUE	BNDLYR	178
C		BNDLYR	179
C		BNDLYR	180
200	CONTINUE	BNDLYR	181
	DO 220 I=2,NM1	BNDLYR	182
	UAMASS(I) = (4.0*AMASS(I) + UAMASS(I))/5.0	BNDLYR	183
	UAMUM(I) = (4.0*AMUM(I) + UAMUM(I))/5.0	BNDLYR	184
	UAENER(I) = (4.0*AEENER(I) + UAENER(I))/5.0	BNDLYR	185
220	CONTINUE	BNDLYR	186
C		BNDLYR	187
C	LET AREAS AT THE BASE OF THE PROJECTILE BE THE SAME AS THOSE AT THE	BNDLYR	188
C	GRID IMMEDIATELY PRECEDING.	BNDLYR	189
	UAMASS(NX) = UAMASS(NM1)	BNDLYR	190
	UAMUM(NX) = UAMUM(NM1)	BNDLYR	191
	UAENER(NX) = UAENER(NM1)	BNDLYR	192
	RETURN	BNDLYR	193
C		BNDLYR	194
	END	BNDLYR	195
		BNDLYR	196
		BNDLYR	197

SUBROUTINE TUBFAL	TUBFAL	2
COMMON/CLOCK/TIME,DELT	TUBFAL	3
COMMON/EGNS/DTDX,TZUR,TZUX,TWOTR,DIER,HMB,TWOGJ,DVAXIS,DVAXIT,	EGNS	2
1 UX,UR,UX,UG,TWOT,HBP	EGNS	3
COMMON/HPT/TEP(2),DEF(2),CEP(2),FTUBR,PLALST	MOD0925	528
COMMON/PRMFLC/LOIMPR,UPRR,OPRR,PRFAP,FDSHR,CSPRKS,PRMCOF	MOD0925	529
COMMON/GRAIN/ XL(60,5), DU(60,5), LI(60,5), FN,	GRAIN	2
1 XLDT(60,5), DDTDT(60,5), DITDT(60,5), XLQ, DQV, DIO,	GRAIN	3
2 ALR(100), LOB(100), DIB(100), UXLB(100), LUOB(100), UDIB(100)	GRAIN	4
COMMON/FAILED/THICKT,PHOUP,PCOMP,ETUB,XATUB,FAIL,MFAIL(60),	TUBFAL	7
1 THICK(60),PLENET,TIFCL	MOD0925	530
COMMON/BAG/PHIBG(60,5), KHOBG(60,5), HBG(60,5), UBG(60,5),	BAG	2
1 VBG(60,5), UFB(60,5), PCN(60,5), TZC(60,5),	BAG	3
2 DUTMG(60), GUBAG(60,5), XUBAG(60,5), DOTPB(60,5), UPBUT(60,5),	BAG	4
3 PHIBTD(60,5), KHIBTD(60,5), HBGTD(60,5), UBGTD(60,5),	BAG	5
4 VBGTD(60,5), UBG(60,5), DOTMG(60), LUTMP(60,5), PHIBP(60,5),	BAG	6
5 PHIBTD(60,5), TZC(60), TBP(60,5)	BAG	7
COMMON/HOLEA/RADHOL(85),NRKOH,NHOLE(85),XCL(85),AREAH(60),	HOLEA	2
1 AH(60),FRACT(60)	HOLEA	3
COMMON/CHAM/IX,IX,XB,KH,NGX,NGR,IBEG,IBEND,IPATH(60,5),AREAG(5),	CHAMIX	2
1 AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMIX	3
2 AREAX(60),AREAX,CHAM1,CHAM2,CHAM3,IUPGAP,AREAGP(60),DAVG,	CHAMIX	4
3 AREAH2,DIAM1,BELBEG,BELBEG,IPS1,IPS2,RADFS,BPIGN	CHAMIX	5
COMMON/INPUTS/L1,C2,C3,C4,T0,TIGN,EGNS,RHOF,PHIU,TF,CA,KHOO,	INPUTS	2
1 H0,P0,U0,GIRHOP,H0,GM,IGNBP,GRCONS,TOTM	INPUTS	3
C THIS SUBROUTINE CALCULATES FAILURE OF THE CENTER CORE IGNITER TUBE	TUBFAL	13
C DUE TO COMPRESSIVE OR TENSILE (HOUP) PRESSURE LOADING.	TUBFAL	14
C INCLUDED IS STRENGTH DEGRADATION DUE TO TUBE COMBUSTION	TUBFAL	15
C MFAIL(1) - TUBE STATUS INDEX	TUBFAL	16
C	TUBFAL	17
C 0 TUBE HAS NOT FAILED	TUBFAL	18
C	TUBFAL	19
C 1 TUBE HAS FAILED IN TENSION	TUBFAL	20
C AND A PSEUDO HOLE HAS BEEN ADDED	TUBFAL	21
C TO SIMULATE FORMATION OF A CRACK	TUBFAL	22
C	TUBFAL	23
C 2 TUBE HAS FAILED IN COMPRESSION.	TUBFAL	24
C A PSEUDO HOLE HAS BEEN ADDED.	TUBFAL	25
C AND PROPELLANT HAS MOVED INTO THE CENTER CORE.	TUBFAL	26
C	TUBFAL	27
C PCOMP - COMPRESSIVE STRENGTH OF IGNITER TUBE	TUBFAL	28
C PHOUP - TENSILE OR HOUP STRENGTH OF IGNITER TUBE	TUBFAL	29
C THICKT - INITIAL IGNITER TUBE THICKNESS	TUBFAL	30
LOGICAL PENEI	MOD0605	31
LOGICAL FAIL	TUBFAL	31
FAIL = .TRUE.	TUBFAL	32
DO 100 I=INEGH,IBEND	TUBFAL	33
IF(MFAIL(I),EQ,2) GO TO 100	TUBFAL	34
IF(PHOUP,EG,0.0.AND.PCOMP,EG,0.0) GO TO 20	TUBFAL	35
FAIL = .FALSE.	TUBFAL	36
C IF THE TUBE HAS ALREADY FAILED IN COMPRESSION,	TUBFAL	37
C IT IS ASSUMED TO BE NONEXISTENT IN THAT GRID,	TUBFAL	38
DELP=PCN(I,1)-PCN(I,2)	TUBFAL	39
C DELP IS POSITIVE FOR HOUP STRESS.	TUBFAL	40
C COMPUTE CURRENT TUBE STRENGTH AFTER IT HAS BEEN DEGRADED DUE TO BURNI	TUBFAL	41
DELT1 = 0.0	TUBFAL	42
DELT2 = 0.0	TUBFAL	43

C IF PROPELLANT IGNITION HAS OCCURRED,	TURFAL	44
C ASSUME THE ADJACENT TUBE SURFACE IS ALSO IGNITED	TURFAL	45
IF (TBP(I,1) .GE. TIGNBP .OR. TZC(I,1) .GE. TIGN) DELTR1 = RTUB *	TURFAL	46
*PCH(I,1)**XNTUB*DEL1	TURFAL	47
IF (TZC(I,2) .GE. TIGN .OR. TZH(I) .GE. TIGN) DELTR2 = BTUB *	TURFAL	48
*PCH(I,2)**XNTUB*DEL1	TURFAL	49
THICK(I) = THICK(I) - DELTR1 - DELTR2	TURFAL	50
KATIO = THICK(I) / THICK1	TURFAL	51
PHOOP1 = PHOOP * KATIO	TURFAL	52
PCOMP1 = PCOMP * KATIO	TURFAL	53
IF (DELP .LT. PHOOP1 .AND. DELP .GT. -PCOMP1 .AND. MFAIL(1) .EQ. 0	TURFAL	54
*) GO TO 100	TURFAL	55
20 IF (MFAIL(1) .EQ. 0) AREA1(I) = 3.1416*.75*NTAMRT*SQRT(DX1)**2.	TURFAL	56
IF (MFAIL(1) .EQ. 0 .AND. I .EQ. 1) AREA1(I) = AREA1(I) / 4.0	TURFAL	57
IF (PHOOP .EQ. 0.0 .AND. PCOMP .EQ. 0.0) GO TO 40	TURFAL	58
MFAIL(1) = 1	TURFAL	59
IF (DELP .GT. -PCOMP1) GO TO 100	TURFAL	60
40 MFAIL(1) = 2	TURFAL	61
F12 = 2.0 - PHIRG(I,2) - PHIRP(I,2)	TURFAL	62
F11 = 2.0 - PHIRG(I,1) - PHIRP(I,1)	TURFAL	63
AREA1 = AREA1(I) + AREA1(I)	TURFAL	64
FIRGN2 = 1.0 - ((F12*AREAR(I) + F11*AREAC(I) - (1.0 - PHIRP(I,2))*AREA1) /	TURFAL	65
*AREA1)	TURFAL	66
FIRGN1 = 1.0 - ((F12*AREAR(I) + F11*AREAC(I) - (1.0 - PHIRP(I,1))*AREA1) /	TURFAL	67
*AREA1)	TURFAL	68
IF (FIRGN1 .LE. 1.0 .AND. FIRGN2 .LE. 1.0) GO TO 50	TURFAL	69
FIRGN2 = PHIRG(I,2)	TURFAL	70
FIRGN1 = PHIRG(I,1)	TURFAL	71
GO TO 100	TURFAL	72
50 CONTINUE	TURFAL	73
RAF1 = (PHIRG(I,1) + PHIRP(I,1) - 1.0) * RHOBG(I,1) * AREAC(I)	TURFAL	74
RAF2 = (PHIRG(I,2) + PHIRP(I,2) - 1.0) * RHOBG(I,2) * AREAR(I)	TURFAL	75
AFN1 = (FIRGN1 + PHIRP(I,1) - 1.0) * AREAC(I)	TURFAL	76
AFN2 = (FIRGN2 + PHIRP(I,2) - 1.0) * AREAR(I)	TURFAL	77
RHOBGN = (RAF2 + RAF1) / (AFN1 + AFN2)	TURFAL	78
UBGN = (RAF2 * URG(I,2) + RAF1 * URG(I,1)) /	TURFAL	79
1 (RHOBGN * (AFN1 + AFN2))	TURFAL	80
HGBN = (RAF2 * HRG(I,2) + RAF1 * HRG(I,1)) /	TURFAL	81
1 (RHOBGN * (AFN1 + AFN2))	TURFAL	82
APH12 = AREAR(I) * (FIRGN2 - PHIRG(I,2))	TURFAL	83
APH11 = AREAC(I) * (1.0 - PHIRG(I,1))	TURFAL	84
APHIN1 = AREAC(I) * (1.0 - FIRGN1)	TURFAL	85
UPN1 = (UPR(I,2) * APH12 + UPR(I,1) * APH11) / APHIN1	TURFAL	86
XLN1 = (XL(I,2) * APH12 + XL(I,1) * APH11) / APHIN1	TURFAL	87
DON1 = (DO(I,2) * APH12 + DO(I,1) * APH11) / APHIN1	TURFAL	88
GIN1 = (GI(I,2) * APH12 + GI(I,1) * APH11) / APHIN1	TURFAL	89
PHIRG(I,1) = FIRGN1	TURFAL	90
PHIRG(I,2) = FIRGN2	TURFAL	91
RHOBG(I,1) = RHOBGN	TURFAL	92
RHOBG(I,2) = RHOBGN	TURFAL	93
URG(I,1) = URGN	TURFAL	94
URG(I,2) = URGN	TURFAL	95
HRG(I,1) = HGBN	TURFAL	96
HRG(I,2) = HGBN	TURFAL	97
UPR(I,1) = UPN1	TURFAL	98
XL(I,1) = XLN1	TURFAL	99
DO(I,1) = DON1	TURFAL	100

```

DI(1,1) = DIM1
ALIDI(1,1)=XLM1
DOTDT(1,1)=DOM1
DIYDT(1,1)=DIN1
OPBDT(1,1)=OPN1
IZC(1,1)-IZC(1,2)
100 CONTINUE
RETURN
END

```

```

TURFAL 101
TURFAL 102
TURFAL 103
TURFAL 104
TURFAL 105
TURFAL 106
TURFAL 107
TURFAL 108
TURFAL 109

```

SUBROUTINE RPINIT	RPINIT	2
COMMON/IGNTF/POHAD,HPCHG,HRMOUT,PGHRA,TPPRG,	MOD0925	531
1 REGTC(5),XLTC(5)	MOD0925	532
COMMON/RPT/TEP(2),DEP(2),CEP(2),FTUHR,PDALST	MOD0925	533
COMMON/CHAP/IX,TR,XH,RR,NGX,NGR,TEFGR,IFENDR,IPATH(60,5),AREAG(5),	CHAMTX	2
1 AREACH,AREAC(60),TGNIT,ONED,DIAM1,DIAM2,CIS1,DIS2,DIS3,DIS4,	CHAMTX	3
2 AREAR(60),AREARX,CHAM1,CHAM2,CHAM3,LOPGAP,AREAGP(60),DAVG,	CHAMTX	4
3 AREAH2,DIAM1,HELLEND,HELLEFG,IPS1,IPS2,RADES,BPIGN	CHAMTX	5
COMMON/PRIMV/HIEMNS,HEKAD(60,5),AGKRP,PGFNEP,EXPBP	PRIMV	2
COMMON/CONS/TDX,TZDR,TZUX,TWOTDR,OTDR,HMR,TWOGJ,DVAXIS,DVAXIT,	CONS	2
1 DX,DXNX,GJ,TWOOT,HRP	CONS	3
COMMON/RAG/PIRG(60,5),RHORG(60,5),RAG(60,5),UBG(60,5),	RAG	2
1 VBG(60,5),UPR(60,5),PCH(60,5),TZC(60,5),	RAG	3
2 DUTMIG(60),OHAC(60,5),XOPAC(60,5),DOTPB(60,5),UPBOT(60,5),	RAG	4
3 PHITD(60,5),PHOTD(60,5),MRGTD(60,5),URGTD(60,5),	RAG	5
4 VBGTD(60,5),TRG(60,5),DOTMRG(60),DOTPF(60,5),PHIRP(60,5),	RAG	6
5 PHITD(60,5),TZK(60),TBP(60,5)	RAG	7
C SUBROUTINE RPINIT CALCULATES INITIAL POROSITY	RPINIT	8
C VALUES AT EACH GRID THAT CONTAINS BLACK	RPINIT	9
C POWDER. THE SUBROUTINE READS IN THE FOLLOWING PARAMETERS	RPINIT	10
C TEP(2)- THICKNESS OF ENDPADS 1 + 2 (INCHES)	RPINIT	11
C DEP(2)- DIAMETER OF ENDPADS 1 + 2 (INCHES)	RPINIT	12
C CEP(2)- CHARGE WEIGHT OF ENDPADS 1 + 2 (OZ.)	RPINIT	13
C ITUBE- NUMBER OF TUBE CHARGES	RPINIT	14
C REGTC(5)- LOCATION OF THE TUBE CHARGES (INCHES)	RPINIT	15
C XLTC(5)- LENGTH OF THE TUBE CHARGES (INCHES)	RPINIT	16
C CHTC(5)- CHARGE WEIGHT OF THE TUBE CHARGES (OZ.)	RPINIT	17
C	RPINIT	18
COMMON/P/ IPRINT,MODCH,MODGR,PRT1,IERHUG(35)	RPINIT	19
COMMON/CFRACT/ FRACT1,FRACT2	RPINIT	20
COMMON/TUBE/IENTC(5),ITUBE,IREGTC(5),INBP	MOD1115	31
C	RPINIT	21
DIMENSION FRACT(5)	RPINIT	22
DIMENSION ENTC(5),	MOD0925	534
* CHTC(5),FRACT(2,60)	MOD1115	32
LOGICAL IPAD(2),IERHUG(35)	RPINIT	25
NAMLIST/BPCHK/ FRUFG,FPDEND,RPOBFR,RPDEND,IREGFP,IFNDFP,IREGFP,	RPINIT	26
* (FNDHP,REGTC,FNDTC,HEL,BEG,BELEND,IRFGR,IFENDR,DX,	RPINIT	27
* IREGTC,IENTC,TWOGR,TWOGR	RPINIT	28
C	RPINIT	29
C INITIALIZE PAD THICKNESS ARRAY	RPINIT	30
C	RPINIT	31
TEP(1) = 0.0	RPINIT	32
TEP(2) = 0.0	RPINIT	33
C	RPINIT	34
C INITIALIZE FRACT ARRAY	RPINIT	35
C	RPINIT	36
CALL CLEAR(FRACT(1,1), FRACT(2,60))	RPINIT	37
C	RPINIT	38
C INITIALIZE IPAD	RPINIT	39
C	RPINIT	40
IPAD(1) = .FALSE.	RPINIT	41
IPAD(2) = .FALSE.	RPINIT	42
C	RPINIT	43
NAMLIST/HPTNP/TEP,DEP,CEP,REGTC,XLTC,CHTC,ITUBE,FTUHR,PDALST	MOD0604	11
C	RPINIT	45
C READ INPUTS AND CALCULATE CONSTANTS	RPINIT	46

```

C      READ (5,RPINP)
      PDALST=144.*PDALST
      IF (IDLRHG(2)) WRITE(6,RPINP)

C      DO 20 I=1,2
      TEP(I) = TEP(I) / 12.0
      DEP(I) = DEP(I) / 12.0
      CEP(I) = CEP(I) / 16.0
20    CONTINUE

C      IF (ITUBE .EQ. 0) GO TO 501
      DO 21 I=1,ITUBE
      BEGIC(I) = BEGIC(I) / 12.0
      XLIC(I) = XLIC(I) / 12.0
      CHIC(I) = CHIC(I) / 16.0
21    CONTINUE

C      IFAD(1) IS SET .TRUE. IF TEP(1) IS GREATER THAN 0.0.
C      IFAD(2) IS SET .TRUE. IF TEP(2) IS GREATER THAN 0.0.
C      DETERMINE IF% THE FRONT PAD, THE BACK PAD
C      OR BOTH BLACK POWDER PADS ARE PRESENT
C      AND IF (TEP(1) .GT. 0.0) IFAD(1) = .TRUE.
C      IF (TEP(2) .GT. 0.0) IFAD(2) = .TRUE.
      BPCNG=CEP(1)+CEP(2)+CHIC(1)+CHIC(2)+CHIC(3)+CHIC(4)+CHIC(5)

C      CALCULATE RADIUS FOR RADIAL GRID NUMBER 1.
      TWODR = (AREAC(IEGRH) + AREAR(IEGRH))/3.1415927
      TWODR = SQRT(TWODR)

C      TWODRB = (AREAC(IENRH) + AREAR(IENRH))/3.1415927
      TWODRB = SQRT(TWODRB)
      IF (IFAD(1)) GO TO 500
      IF (IFAD(2)) GO TO 510
      GO TO 516

C      CALCULATE POSITION OF FRONT PAD, THE GRIDS IT LIES IN
C      AND THE FRACTIONS OF THE GRIDS IT LIES IN
C      IF THE FRONT PAD EXTENDS BEYOND RADIAL
C      GRID NUMBER 2 - REDISTRIBUTE IT IN RADIAL
C      GRIDS 1 AND 2 BY INCREASING TEP(1).
C      500 REP1 = DEP(1) / 2.0
      IF (REP1 .LE. TWODR) GO TO 505
      TEP(1) = REP1*REP1*TEP(1)/(TWODR+TWODR)
      REP1 = TWODR
      505 FPUREG = BELREG - TEP(1)

C      DETERMINE WHICH GRIDS THE PAD IS IN
C      IF (FPUREG .GE. .5*DX) GO TO 506
      ISLGFP = 1
      GO TO 509

```

```

RPINIT  47
RPINIT  48
MON0606 13
RPINIT  49
RPINIT  50
RPINIT  51
RPINIT  52
RPINIT  53
RPINIT  54
RPINIT  55
RPINIT  56
RPINIT  57
RPINIT  58
RPINIT  59
RPINIT  60
RPINIT  61
RPINIT  62
RPINIT  63
RPINIT  64
RPINIT  65
RPINIT  66
RPINIT  67
RPINIT  68
RPINIT  69
RPINIT  70
RPINIT  71
MON1002 16
RPINIT  72
RPINIT  73
RPINIT  74
RPINIT  75
RPINIT  76
RPINIT  77
RPINIT  78
RPINIT  79
RPINIT  80
RPINIT  81
RPINIT  82
RPINIT  83
RPINIT  84
RPINIT  85
RPINIT  86
RPINIT  87
RPINIT  88
RPINIT  89
RPINIT  90
RPINIT  91
RPINIT  92
RPINIT  93
RPINIT  94
RPINIT  95
RPINIT  96
RPINIT  97
RPINIT  98
RPINIT  99
RPINIT 100
RPINIT 101

```

506	IREGFP = FPDREG / DX + .5	RPINIT	102
	IREGFP = IREGFP + 1	RPINIT	103
509	IENDFP = IREGFP	RPINIT	104
C	FRACT IS AN ARRAY WHICH WILL CONTAIN THE FRACTIONS OF GRIDS THE PAD D	RPINIT	105
	K = IREGFP	RPINIT	106
	IF (K .EQ. 1) GO TO 600	RPINIT	107
	IF (K.LT. IENDFP) GO TO 601	RPINIT	108
	FRACT(1,IENDFP) = TEP(1)/DX	RPINIT	109
	GO TO 603	RPINIT	110
601	FRACT(1,K) = ((FLOAT(IREGFP) - .5)*DX - FPDREG)/DX	RPINIT	111
	GO TO 604	RPINIT	112
604	IF (K.LT. IENDFP) GO TO 602	RPINIT	113
	FRACT(1,IENDFP) = TEP(1)/(1.5*DX)	RPINIT	114
	GO TO 603	RPINIT	115
602	FRACT(1,IREGFP) = ((FLOAT(IREGFP) - .5)*DX - FPDREG)/(1.5*DX)	RPINIT	116
604	K = K + 1	RPINIT	117
	DO 512 I=K,IENDFP	RPINIT	118
	IF (I .EQ. IENDFP) GO TO 605	RPINIT	119
	FRACT(1,I) = 1.0	RPINIT	120
512	CONTINUE	RPINIT	121
605	FRACT(1,IENDFP) = 1.0 - FRACT1	RPINIT	122
603	IF (1 .NOT. IPAD(2)) GO TO 515	RPINIT	123
C		RPINIT	124
C	DO CALCULATIONS FOR BACK PAD	RPINIT	125
C		RPINIT	126
C	IF THE BACK PAD EXTENDS BEYOND RADIAL GRID	RPINIT	127
C	NUMBER 2 - REDISTRIBUTE IT IN RADIAL GRIDS	RPINIT	128
C	1 + 2 BY INCREASING TEP(2).	RPINIT	129
C		RPINIT	130
510	REP2 = REP(2) / 2.0	RPINIT	131
	IF (REP2 .LE. TWOGRB) GO TO 511	RPINIT	132
	TEP(2) = REP2*REP2*TEP(2)/(TWOGRB+1*CORR1	RPINIT	133
	REP2 = TWOGRB	RPINIT	134
C		RPINIT	135
C		RPINIT	136
511	IREGFP = IENDB	RPINIT	137
	IREGFP = IREGFP	RPINIT	138
	IREGFP = IREGFP + IEP(2)	RPINIT	139
	IENDFP = IREGFP/DX+.5	RPINIT	140
	IENDFP = IENDFP + 1	RPINIT	141
	FRACT(1,IREGFP) = 1.0 - FRACT2	RPINIT	142
	IF (IREGFP .EQ. IENDFP) GO TO 515	RPINIT	143
	KB = IREGFP + 1	RPINIT	144
	DO 514 I = KB,IENDFP	RPINIT	145
	IF (I .EQ. IENDFP) GO TO 521	RPINIT	146
	FRACT(1,I) = 1.0	RPINIT	147
514	CONTINUE	RPINIT	148
521	FRACT(1,IENDFP) = IENDFP - (FLOAT(IREGFP) - 1.5)*DX/DX	RPINIT	149
515	IF (1 .NOT. IPAD(1)) GO TO 608	RPINIT	150
C		RPINIT	151
C	CALCULATE PROPER VALUES FOR FRACT(2,I)	RPINIT	152
C		RPINIT	153
	TEMPVL = REP1*REP1*3.141593	RPINIT	154
	IF (IREGFP .GT. 1) GO TO 606	RPINIT	155
	VOL1 = (TEMPVL - AREAC(1))*DX*.5*FRAC(1,1)	RPINIT	156
	FRACT(2,1) = VOL1/(AREAR(1)*DX*.5)	RPINIT	157
	GO TO 607	RPINIT	158

606 VOL1 = (TEMPV - AREAC(IEGEP)) * DX * FRAC(1, IREGEP)	RPINIT	154
FRAC(2, IREGEP) = VOL1 / (AREAR(IREGEP) * DX)	RPINIT	155
607 K = IREGEP + 1	RPINIT	156
IF (K .GT. IENDFP) GO TO 608	RPINIT	157
DO 609 I = K, IENDFP	RPINIT	158
VOL1 = (TEMPV - AREAC(I)) * DX * FRAC(1, I)	RPINIT	159
FRAC(2, I) = VOL1 / (AREAR(I) * DX)	RPINIT	160
IF (I .EQ. IENDFP) GO TO 608	RPINIT	161
609 CONTINUE	RPINIT	162
608 IF (.NOT. IPAD(2)) GO TO 516	RPINIT	163
TEMPV = RLP2 * REP2 * 3.141593	RPINIT	164
VOL2 = (TEMPV - AREAC(IEGBP)) * DX * FRAC(1, IEGBP)	RPINIT	165
FRAC(2, IEGBP) = VOL2 / (AREAR(IEGBP) * DX)	RPINIT	166
KB = IEGBP + 1	RPINIT	167
IF (KB .GT. IENDBP) GO TO 516	RPINIT	168
DO 700 I = KB, IENDBP	RPINIT	169
VOL2 = (TEMPV - AREAC(I)) * DX * FRAC(1, I)	RPINIT	170
FRAC(2, I) = VOL2 / (AREAR(I) * DX)	RPINIT	171
700 CONTINUE	RPINIT	172
C COMPUTE LOCATION OF TUBE CHARGES AND THE PROPER FRACTIONS	RPINIT	173
C FOR THE GRIDS THEY OCCUPY	RPINIT	174
C	RPINIT	175
516 DTC = DIAMBT	RPINIT	176
C	RPINIT	177
C CHECK FOR EXTENSION OF TUBE CHARGES	RPINIT	178
C BEYOND THE LIMITS OF THE WELL TUBE	RPINIT	179
C	RPINIT	180
IF (ITUBE .EQ. 0) GO TO 800	RPINIT	181
IF (BEGTC(1) .LT. BELEBG) GO TO 120	RPINIT	182
LEUTC(ITUBE) = BEGTC(ITUBE) + XTTC(ITUBE)	RPINIT	183
IF (LEUTC(ITUBE) .GT. RELEND) GO TO 121	RPINIT	184
C	RPINIT	185
C DETERMINE GRIDS IN WHICH EACH TUBE CHARGE BEGINS AND ENDS.	RPINIT	186
C	RPINIT	187
DO 100 I = 1, ITUBE	RPINIT	188
LEUTC(I) = BEGTC(I) + XTTC(I)	RPINIT	189
IF (BEGTC(I) .GE. .5 * DX) GO TO 805	RPINIT	190
IBEGTC(I) = 1	RPINIT	191
GO TO 806	RPINIT	192
805 IBEGTC(I) = BEGTC(I) / DX + .5	RPINIT	193
IBEGTC(I) = IBEGTC(I) + 1	RPINIT	194
806 LEUTC(I) = LEUTC(I) / DX + .5	RPINIT	195
LEUTC(I) = LEUTC(I) + 1	RPINIT	196
C	RPINIT	197
C CALCULATE FRACTIONS OF BEGINNING AND END GRIDS	RPINIT	198
C WHICH EACH TUBE CHARGE OCCUPIES	RPINIT	199
C	RPINIT	200
KE = LEUTC(1)	RPINIT	201
IF (I .EQ. 1) GO TO 804	RPINIT	202
KK = LEUTC(I - 1)	RPINIT	203
IF (LEUTC(I - 1) .EQ. TRFGTC(I)) TFRAC(1, I) = FRAC(1, KK)	RPINIT	204
804 FRAC(1, KE) = (LEUTC(I) - (FLOAT(KE) - 1.5) * DX) / DX	RPINIT	205
K = IBEGTC(I)	RPINIT	206
IF (K .EQ. 1) FFRAC(1, K) = ((FLOAT(K) - .5) * DX - BEGTC(I)) / .5 * DX	RPINIT	207
FFRAC(1, K) = ((FLOAT(K) - .5) * DX - BEGTC(I)) / DX	RPINIT	208
100 CONTINUE	RPINIT	209
	RPINIT	210
	RPINIT	211
	RPINIT	212
	RPINIT	213
	RPINIT	214
	RPINIT	215

C	APINIT	216
C CALCULATE FRACTIONS FOR WHOLE GRIDS COLLECTED	APINIT	217
C	APINIT	218
DO 101 I=1,ITUBE	APINIT	219
KNEG=IBEGTC(I) +1	APINIT	220
KEND = IENDTC(I) - 1	APINIT	221
IF (KNEG .LT. KEND) GO TO 103	APINIT	222
FRACT(1,KNEG) = 1.0	APINIT	223
GO TO 101	APINIT	224
103 DO 102 J= KNEG,KEND	APINIT	225
FRACT(1,J) = 1.0	APINIT	226
102 CONTINUE	APINIT	227
101 CONTINUE	APINIT	228
C	APINIT	229
C USING THE FRACTIONS STORED IN ARRAY FRACT,	APINIT	230
C CALCULATE THE POROSITIES FOR EACH GRID	APINIT	231
C	APINIT	232
C FRONT PAD	APINIT	233
NOB IF (.NOT. IPAD(1)) GO TO 140	APINIT	234
C	APINIT	235
DO 110 I=IBEGFP, IENDFP	APINIT	236
DO 111 J=1,2	APINIT	237
VOLF = 3.141593*REF1*REF1*TFP(1)	APINIT	238
PHIBP(1,J) = 1.0 - (CEP(1)*FRACT(J,1))/VOLF/BPUENS	APINIT	239
111 CONTINUE	APINIT	240
110 CONTINUE	APINIT	241
C	APINIT	242
C BACK PAD	APINIT	243
140 IF (.NOT. IPAD(2)) GO TO 150	APINIT	244
C	APINIT	245
DO 112 I = IBEGBP, IENDBP	APINIT	246
DO 113 J = 1,2	APINIT	247
VOLB = 3.141593*REF2*REF2*TFP(2)	APINIT	248
PHIBP(1,J) = 1.0 - (CEP(2)*FRACT(J,1))/VOLB/BPUENS	APINIT	249
113 CONTINUE	APINIT	250
112 CONTINUE	APINIT	251
C	APINIT	252
C TUBE CHARGES	APINIT	253
C	APINIT	254
150 IF (ITUBE .EQ. 0) GO TO 802	APINIT	255
RRT = DIAMBT/2.0	APINIT	256
RRTSC = RRT * PRT	APINIT	257
DO 115 J=1,ITUBE	APINIT	258
KNEG = IBEGTC(J)	APINIT	259
KEND = IENDTC(J)	APINIT	260
IF (IHEGFP .EQ. 1 .AND. IENDEP .EQ. 1) GO TO 810	APINIT	261
FRACT3 = 1.0 - FRACT1	APINIT	262
GO TO 811	APINIT	263
810 FRACT3 = TFP(1)/(1.5*GX)	APINIT	264
811 IF (IHEGRP .EQ. IENDRP) GO TO 812	APINIT	265
FRACT4 = 1.0 - FRACT2	APINIT	266
GO TO 813	APINIT	267
812 FRACT4 = TFP(2)/HX	APINIT	268
813 K = KNEG	APINIT	269
N = KEND	APINIT	270
VOLC = PHISQ*XLTC(J)*3.141593	APINIT	271
DO 116 J=KNEG,KEND	APINIT	272

PHIBP(1,1) = 1.0 - (CHIC(1)*FRACT(1,1)/VOLC/PPDENS	RPINIT	273	
NAMELIST/BILLT/ VOLC,TUBVOL,CHTC,TERACT,BPDENS,FRED1	RPINIT	274	
NAMELIST/BILLH/ VOLC,VOLH,CEP,FRACT2,FRACT4,CHTC,PPDENS,FRED2	RPINIT	275	
NAMELIST/BILLF/ VOLC,VOLF,CEP,FRACT1,FRACT3,CHTC,BPDENS,FRED1,CHK	RPINIT	276	
CHK = PHIBP(1,1)	RPINIT	277	
FRED1 = FRACT(1,1)	RPINIT	278	
FRED2 = FRACT(1,16)	RPINIT	279	
FRED1 = FRACT(1,10)	RPINIT	280	
IF(IENDFP .EQ. KREF) PHIBP(M,1) = 1.0 - ((VOLC*CEP(1))*	FRACT3)	RPINIT	281
+CHTC(J)*VOLC*FRACT(1,KREF)/VOLC*		RPINIT	282
VOLC/PPDENS)		RPINIT	283
IF(IDEBUG(33) .AND. IENDFP .EQ. KREF) WRITE(6,BILLF)		RPINIT	284
IF(IHGBP .EQ. KEND) PHIBP(M,1) = 1.0 - ((VOLC*CEP(2))*	FRACT4)	RPINIT	285
+CHTC(J)*VOLC*FRACT(1,KEND)/VOLC*		RPINIT	286
VOLC/PPDENS)		RPINIT	287
IF(IDEBUG(33) .AND. IHGBP .EQ. KEND) WRITE(6,BILLH)		RPINIT	288
IF(J .EQ. 1) GO TO 116		RPINIT	289
TUBVOL = RATSQ*XLTC(J-1)*3.141593		RPINIT	290
L = IBEGTC(J)		RPINIT	291
IF(IENDTC(J-1) .EQ. IBEGTC(J)) PHIBP(L,1) = 1.0 - ((VOLC*TERACT(J)*		RPINIT	292
CHTC(J-1) + TUBVOL*FRACT(1,1)*CHTC(J))/(TUBVOL*VOLC/PPDENS)		RPINIT	293
IF(IDEBUG(33) .AND. IENDTC(J-1) .EQ. IBEGTC(J)) WRITE(6,BILLT)		RPINIT	294
116 CONTINUE		RPINIT	295
115 CONTINUE		RPINIT	296
802 IF(IDEBUG(30)) WRITE(6,BPCHK)		RPINIT	297
RETURN		RPINIT	298
120 WRITE(6,1000)		RPINIT	299
STOP		RPINIT	300
121 WRITE(6,1001)		RPINIT	301
STOP		RPINIT	302
1000 FORMAT(1H1,'TUBE CHARGE = 1 EXTENDS OUTSIDE THE HELI TUBE')		RPINIT	303
1001 FORMAT(1H1,'TUBE CHARGE = 5 EXTENDS OUTSIDE THE HELI TUBE')		RPINIT	304
END		RPINIT	305

APPENDIX C

REQUIRED INPUT AND OUTPUT

A. REQUIRED INPUT

The input quantities for the computer code are read in with five separate statements. The first quantity, IDEBUG, is read in with an L format where card spaces 1 through 35 are marked with either a T or an F to denote various output displays. The other four quantities are read in with a NAMELIST format. The first of these, called MØDS, contains two items that are used to regulate the time and space intervals of the primary program output. The second group, called CHINP (for chamber input), contains those inputs required to perform all calculations in the gun chamber and is called from subroutine CHSET. The third group, called BPINP (for black powder input), reads in all input required to load black powder into the center core ignition system and is called from subroutine BPINIT. The last group, called BARINP (for barrel input), contains those additional inputs required to perform the barrel calculations and is read in from subroutine BARSET. This appendix gives a description of all the required input quantities and a value used to represent the XM185 gun with the XM123E2 propelling charge and projectile.

OUTPUT SELECTION - IDEBUG

IDEBUG A logical variable array with a dimension of 35 that is used to specify which output is to be displayed for a given computer run. If IDEBUG(K) is TRUE, the Kth block of output will be displayed (see B. OUTPUT).

OUTPUT CONTROL - NAMELIST MØDS

MØDCH The number of time step intervals between normal data printouts. A value of 100 has proved satisfactory for most computer runs where a fairly detailed history of all flow parameters is desired.

MØDGR The number of I intervals between print locations. A value of 4 will result in data printout for I = 1, 4, 8, 12, 16, and 20, for all values of J.

CHAMBER INPUTS - NAMELIST CHINP

AGEN The "A" term in the propellant burning rate equation

$$\dot{r} = (AT_o + B) p^E + CT_o$$

In this expression, p is in psi, T_o in $^{\circ}R$, and $\dot{r}$ is in in/sec. Temperature dependence was not used for M6 propellant, and this term was set equal to zero.

AGENBP Similar to AGEN but for black powder. This term was set equal to .248.

ALPHA The thermal diffusivity of the propellant. This quantity is assumed not to differ significantly between the various propellants. A value of 1.0×10^{-6} ft²/sec was assumed for the checkout calculations.

ALPHBP The thermal diffusivity for black powder. This quantity is assumed to be 1.0×10^{-6} ft²-sec, the same as for the propellant.

BETA A parameter that is required to maintain a stable finite difference solution to the differential equations of fluid motion. A value of 0.5 is known to work satisfactorily, but values up to 1.0 may work under certain conditions. BETA is directly proportional to the time interval between calculations and therefore inversely proportional to the machine time required for the calculation.

BGEN The "B" term in the propellant burning rate equation (see AGEN). A value of .0057 was used for M30 propellant in the checkout runs.

BGENBP Similar to BGEN but for black powder. This term was set equal to 0.

BØRED Average barrel diameter, taking lands and grooves of the rifling into account, = 6.962 in.

BPDENS Black powder granule density, = 116 lbm/ft³.

BPRADO Initial effective radius of black powder granules assuming a spherical configuration, = 0.067 in.

BTUB The coefficient in the equation, $\Delta t_h = B p^n$, which calculates center core tube wall recession due to combustion, assumed equal to 0.001 for recession measured in inches and pressure in psi.

CA Flow coefficient for igniter tube and "pseudo" holes, = 0.8.

CGEN The "C" term in the burning rate equation (see AGEN), set equal to 0.1246×10^{-3} for the propellant.

CHAM1 Logical variable set .TRUE. when the chamber grid matrix is two dimensional. This feature is not operational on the program described in this report, and it should always be set .FALSE.

CHAM2 Logical variable set .TRUE. when the chamber grid matrix consists of two parallel one-dimensional networks. Otherwise, it is set .FALSE.

CHAM3 Logical variable set .TRUE. when the chamber grid matrix consists of three parallel one-dimensional networks. Otherwise it is set .FALSE.

CHWT Propellant charge weight, set equal to 25.5 lb. for the 155mm gun XM123E2 charge.

CVHBP Slope of the specific heat versus enthalpy curve for black powder, assumed equal to $0.555 \times 10^{-4}/^{\circ}\text{R}$.

CVHM6 Slope of the specific heat versus enthalpy curve, equal to $0.358 \times 10^{-4}/^{\circ}\text{R}$.

CVOBP Intercept term in the equation for specific heat of black powder, assumed equal to 0.177.

CVOM6 Intercept term in the specific heat equation, $C_v = C_{v0} + C_{vH} H$, equal to $0.198 \text{ Btu/lbm } ^{\circ}\text{R}$.

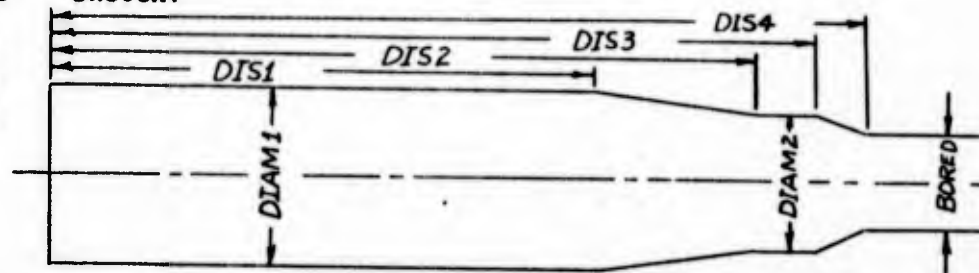
C1,C2 Constants in Sutherland's equation for viscosity in lbm/ft-sec. $C1 = 0.7535 \times 10^{-6}$; $C2 = 262.5$.

C3,C4 Constants in Sutherland's equation for thermal conductivity in Btu/ft-sec - $^{\circ}R$. $C3 = 0.291 \times 10^{-6}$; $C4 = 170.1$.

DIAMBT Inside diameter of the igniter tube, equal to 1.17 in.

DIAM1 Chamber dimensions as defined by the following sketch:

DIAM2
DIS1
DIS2
DIS3
DIS4



DIO Propellant grain minor or perforation diameter, equal to 0.03 in. for M6 propellant.

DØØ Propellant grain major or exterior diameter, equal to 0.420 in. for M6 propellant.

DR Radial grid dimension for the two-dimensional chamber grid matrix. This is not used in the version of the computer program described in this report.

DRAM Distance from the projectile base to the breech, equal to 32.5 in. for the 155mm gun base run configuration.

EXPBP Black powder burn rate pressure exponent, = 0.24.

FRCTF The flow friction factor of the propellant bed which was determined by experiment to be nearly equal to 1.0.

GAP Distance from the projectile base to the nearest end of the ball igniter tube.

HBP Heat of combustion of black powder, equal to 1223 Btu/lbm.

IMAX An approximate estimate of the maximum enthalpy to be encountered. This is used with BETA to determine a stable time interval for calculation. A value of 2200 Btu/lbm is good for the expected range of calculations.

IMB The internal energy added by burning propellant. This is not the same as heat of reaction. For M30, this is approximately 1540 Btu/lbm.

INBP The index that defines the grid number where the igniter becomes jammed in the event of failure of the bag ties.

NGR Number of radial grids in the chamber matrix. This number must coincide with the matrix selected by CHAM1, CHAM2, or CHAM3. Currently, the most this number can be is 5, as dictated by common storage allocated to the variables.

NGX Number of axial grids in the chamber matrix. This number cannot exceed 60, which is the number currently allocated the variables in common storage.

NHØLES Array giving the number of holes in a circumferential row on the igniter tube, a row being defined as all the holes with the same axial position on the tube. The 155mm gun igniter tube has holes at each axial position and NHØLES is filled with $4 * 1$, $81 * 0$, to fill up the entire array of 85 potential rows.

NPERF Number of perforations in the grains of the main charge. The M30 propellant used in the 155mm gun has 7 perforations.

NRØWH Number of rows of holes in the igniter tube, equal to 4 for the 155mm gun.

PCØMP Pressure load required for the center core tube to fail in compression, equal to 150 psi.

PEXP Pressure exponent to the propellant burn rate equation (see AGEN), equal to 0.65 for the base run.

PHØØP Pressure load required for the center core tube to fail in tension, equal to 150 psi.

PØ Initial pressure, = 14.7 psi.

PRMCOF Primer strength (energy) ratio compared to the M82 primer.

QSPRKS The heating by the primer jet due to sparks. A value of 10 has been used ($\text{Btu/ft}^2\text{-sec}$) for present results.

RADIØL Igniter tube hole radius array, equal to 0.125 in. for the 155mm gun igniter tube. The entire array is filled by $4 * 0.125$, $81 * 0.0$.

RRØBP Slope of the curve of gas state parameter versus density for black powder, assumed to be the same as a similar constant for the propellant, $1.9 \text{ (ft}^4\text{-lbf)/(lbm}^2\text{-}^\circ\text{R)}$, for the base run.

RRØM6 Slope of the curve of gas state parameter versus density for the propellant, equal to $1.9 \text{ (ft}^4\text{-lbf)/(lbm}^2\text{-}^\circ\text{R)}$ for the base run (see Figure 23).

ROBP The ordinate intercept of the curve of gas state parameter versus density for black powder, assumed to be the same as a similarly defined parameter for the propellant, $66 \text{ ft-lbf/lbm-}^\circ\text{R}$, for the base run.

ROM6 The ordinate intercept of the curve of gas state parameter versus density for the propellant, equal to $63 \text{ ft-lbf/lbm }^\circ\text{R}$ for the base run (see Figure 23).

RHØP Grain density of M30 propellant, equal to 103.7 lbm/ft^3 .

TF Time extent of the calculation. This provides an alternate method to terminate a computer run.

THICKT Initial thickness of the center core tube, equal to 0.117 in.

TIGN Propellant grain surface temperature at which ignition occurs, $\approx 900^\circ\text{R}$ for M6 propellant.

TIGNBP Ignition temperature for black powder, assumed equal to 1300°R .

TØPGAP Average distance between the main propellant charge and the gun chamber. This assumes that the charge is symmetrically located in the chamber. A value of 0.25 in. was used for base run calculations.

TW Initial temperature of gun surface, equal to 535°R .

TØ Initial temperature of the gas and propellant grains, 535°R .

UØ Initial gas velocity in the axial direction, equal to 0.0 ft/sec.

TYFCE Strength of igniter center core loading bag ties, equal to 24 lbs.

VO Initial gas velocity in the radial direction, equal to 0.0 ft/sec.

XCL Array that specifies the axial position of igniter tube holes with respect to the breech. It is specified as the location of the first hole and the distance between adjacent holes. The program inputs are XCL = 1.5, 1.75, 3.5, 15, 81 * 0.0 for the 155mm gun, with all distances given in inches.

XK Thermal conductivity of a grain of propellant, assumed equal to 0.2×10^{-4} Btu/ft-sec-°R.

XKBP Thermal conductivity of black powder, assumed equal to 0.2×10^{-4} Btu/ft-sec-°R.

XLBEL Location of the igniter tube end nearest the projectile, equal to 25 in. for the base run.

XLO Average initial propellant grain length, equal to 0.90 in. for M30 propellant.

XNTUB Pressure exponent for igniter tube burn rate equation, $\Delta t_b = B p^n$, and assumed equal to 0.86.

BLACK POWDER INPUTS - NAMELIST BPINP

BEGTC() Axial position of the beginning of the tube charge, the bag of black powder positioned inside the igniter tube, equal to (TEP + 0.5 + standoff, 4 x 0.). for the 155mm gun.

CEP() Black powder charge for the end pads, equal to 1.0, 0.0 oz. for the 155mm gun.

CHTC() Black powder charge for the bags located in the igniter tubes, equal to 4.0, 4 x 0.

DEP() Diameter of the end pads, equal to 6.10 in.

FTUBR Flash tube diameter, equal to 0.0104 ft.

ITUBE Number of tube charges in gun configuration of interest, equal to 1 for the 155mm gun.

PDALST End pad allowable dynamic shear stress, equal to 10,000 psi.

TEP() End pad thickness, assumed equal to 0.5, 0.0
for the 155mm gun.

XLTC() Length of the black powder tube charges, equal
to 22., 4 x 0. for the 155mm gun.

BARREL INPUTS - NAMELIST BARINP

CF Coefficient of friction between projectile
rotating band and the barrel, assumed equal to
0.0 in order to obtain reasonable agreement
of interior ballistics calculations.

DEPTH The distance a land extends above the barrel
surface, equal to 0.05 in. for the 155mm gun.

NX Number of barrel grids set initially, normally
equal to 1.

PDMAX Maximum pressure encountered during engraving of
rotating band, corresponding to the shot-start
force, assumed equal to 6,400 psi for the
base run.

PINER Projectile moment of inertia about its axis,
approximately equal to .107 lbf/sec²-ft.

PMASS Projectile mass, equal to 95.6 lb.

PL Portion of the rotating band circumference
to be engraved, 1.164 ft.

PLO Barrel pressure at point XLO equal to 800 psi.

PZO Barrel pressure at point 0, equal to 2700 psi.

PINT Barrel pressure at point XINT, equal to 2000 psi.

RADPB Effective radius of the projectile cross-
section upon which the pressure acts, equal
to 3.08 in.

RØTK Proportionality constant relating projectile
rotational and axial velocities. This constant
is related to the twist of the rifling and is
equal to 0.619 circumferential radius per foot
of barrel.

THETA Pitch of rifling, equal to 0.157 rad. ft. or
9 degrees. The input number is in radians.

WØB Width of the obturator, or rotating band,
equal to 1.0 in.

XLBAR Total length of the barrel, equal to 238 in.
 for the 175mm gun.

XINT Intermediate point of barrel pressure plot, equal
 to 10 inches.

XLO End point of barrel pressure plot, equal to 17.0
 inches.

XPMAX Point on barrel pressure plot corresponding to
 PDMAX, equal to 1.5 inches. (Same as WOB).

B. OUTPUT

The complete program output consists of several initial NAMELIST and array printouts and the primary output from the grid matrix during the run. The amount of output is regulated through the logical input array, IDEBUG. This variable provides optional display of the following groups of output:

- IDEBUG (1)** Printout of those quantities input to the program
 through NAMELIST MØDS.
- (2)** Printout of those quantities input to the program
 through NAMELIST CHINP.
- (3)** Printout of initial porosity arrays, PHIBG and
 PHIBP.
- (4)** Printout of NAMELIST CHKIN which lists many
 converted quantities and computed parameters in
 the units that are used in the program from sub-
 routine CHSET.
- (5)** Printout of the IPATH array, that governs the
 logical flow through the sequence of finite
 difference subroutines.
- (6)** Printout of chamber grid cross-sectional area
 arrays.

- (7) Printout of those quantities input to the program through NAMELIST BARINP.
- (8) Printout of NAMELIST BARCHK which lists many converted quantities and computed parameters in the units that are used in the program for subroutine BARSET.
- (9) Printout from subroutine BPFIR that states the time and indices of a grid when the black powder in that grid becomes ignited.
- (10) Printout from subroutine PRPFIR that states time and grid indices upon propellant ignition in that grid.
- (11) Printout from subroutine PRPFIR that states time and grid indices upon ignition of propellant under the influence of the holes in the igniter tube.
- (12) Printout of the time interval number, IPRINT, and the revised number of grids in the chamber, NGX, when subroutine NEWDX is called.
- (13) Printout of NAMELIST NEWCHK from subroutine NEWDX that displays computational parameters that were changed as a result of changing the grid size.
- (14) Printout of the revised chamber grid area arrays, AREAGP, AREAR, AREAC, and AREAG, from subroutine NEWDX after the grid matrix size is reduced.
- (15) Printout of all variables for the chamber grid matrix at time intervals specified by M_{TDCH} and space intervals specified by M_{TDGR}. A sample printout is shown at the end of Appendix C.
- (16) Printout of NAMELIST DOT from subroutine UPDATE that includes the arrays DOTMIG, DOTMBG, and TZR, shown at the end of Appendix C.
- (17) Printout from UPDATE of the time and time interval number at which the multiple one-dimensional grid network was reduced to a single one-dimensional network.

- (18) Printout from UPDATE of variables from the chamber grid matrix after it has been reduced to a one-dimensional network.
- (19) Printout of all variables from the barrel grid matrix at time intervals specified by MØDCH. A sample printout is shown at the end of Appendix C.
- (20) Printout of statements regarding the masses of propellant and gas that actually exist or were loaded into the system and those that are computed by summing over all the grids in the system and multiplier applied to the computed mass of gas to make the computed mass equal the actual mass, printed from subroutine UPDATE. A sample printout is shown at the end of Appendix C.
- (21) Printout from UPDATE of cumulative amount of propellant and black powder burned in chamber and barrel and the revised gas constants based on an average of black powder and propellant properties according to the amount of each in the system at time intervals regulated by MØDCH. A sample printout is shown at the end of Appendix C.
- (22) Printout from UPDATE of pressure $PCH(1,1)$ at intervals regulated by MØDCH.
- (23) Printout from subroutine MOTION of projectile motion variables at intervals regulated by MØDCH and the time the projectile passes from the barrel.
- (24) Printout from UPDATE of certain pressures in addition to projectile displacement and velocity and the time and time interval of the printout. This occurs every ten intervals and is not regulated by MØDCH. The items printed out do not have a heading but occur as follows:

 TIME, IPRINT, $PCH(1,1)$, $PCH(1,2)$, $PCH(NGX,1)$,
 $PCH(NGX,2)$, $PG(NX)$, XP, VP.
- (25) Not used at present.
 to
 (29)

- (30) Printout from subroutine BPINIT of certain parameters pertaining to initial black powder distribution calculations performed in that subroutine.
- (31) Not used at present.
- to
- (33) and 35
- (34) Printout of primer jet characteristics from subroutine UPDATE.

The title blocks and sample printcuts for several of the output options are given below. The terms are defined in Appendices C and D. The units of these terms follow the following rule:

Velocity	ft/sec
Density	lbm/ft ³
Enthalpy	Btu/lbm
Pressure	lbf/in ²
Propellant grain dimensions	ft.
Propellant burning	lbm/ft ³
Gas resistance	psf/ft
Areas	ft ²
Heat flux	Btu/ft ² -sec
Temperatures	°R

CHAMBER OUTPUT

208

14	1	.10000000E+01	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.00147000E+00	0.	0.	0.	0.	0.	0.	0.
		.53500000E+03	0.	0.	0.	0.	0.	0.	0.
14	2	.54019420E+00	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.10000000E+01	0.	0.	0.	0.	0.	0.	0.
		0.	.54733333E+01	.36503333E+01	.70000000E+00	0.	0.	0.	0.
14	3	.10000000E+01	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.10000000E+01	0.	0.	0.	0.	0.	0.	0.
		0.	0.	0.	0.	0.	0.	0.	0.
20	1	.10000000E+01	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.10000000E+01	0.	0.	0.	0.	0.	0.	0.
		0.	0.	0.	0.	0.	0.	0.	0.
20	2	.10000000E+01	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.10000000E+01	0.	0.	0.	0.	0.	0.	0.
		0.	0.	0.	0.	0.	0.	0.	0.
20	3	.10000000E+01	.50000000E+01	0.	0.	0.	.10000000E+01	.53500000E+03	.14700000E+02
		.10000000E+01	0.	0.	0.	0.	0.	0.	0.
		0.	0.	0.	0.	0.	0.	0.	0.

DEBUG(10)
END

ANTH10	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH20	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH30	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH40	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH50	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH60	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH70	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH80	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH90	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ANTH100	=	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

END

DEBUG(10)		BARREL OUTPUT									
		PHI	RMG	Q	U	NR	TC	PC	HP		
			OL	UOR	UOR	DIR	AMSS	AMM	AFMR		
1	.10000000E+01	.50000000E+01	0.	0.	0.	0.	.20000000E+00	.20000000E+00	.20000000E+00	0.	
DEBUG(10)											
MASS OF AIR IN THE SYSTEM IS 1.0000 THE MASS OF AIR IN THE SYSTEM IS 1.0000											
THE MASS OF PROPELLANT AND BLACK POWDER IN THE SYSTEM IS 56.7000 ACTUAL BAR IN THE SYSTEM IS 1.0000 ADJUSTING FUNCTION IS 1.0000											
DEBUG(10)											
10	.10000000E+01	.50000000E+01	.70000000E+01	.30000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	0.	
	.20000000E+01	.13000000E+01	.40000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	.20000000E+01	0.	

APPENDIX D

GLOSSARY

(1) Most of the FORTRAN words used in the computer program are defined here. A cross-reference list linking the terms used in this report to their respective FORTRAN names follows the glossary. Many definitions are given in Appendix C, and those are not repeated here. All items listed here contain units of lbm, ft, sec, and Btu, as used in the code.

ACC Projectile acceleration.

ACCL, ACCLF Propellant acceleration.

ADJUST Factor applied to porosity and gas density in order to maintain the proper gas and propellant mass in the system.

ADV Propellant grain surface area in one grid.

ADVBP Black powder granule surface area in one grid.

AEENER(I) Barrel cross-sectional area less that included in the energy thickness of the boundary layer.

AFN1, Parameters representing the product of
AFN2 area and porosity in grids $j = 1$ and 2 in subroutine TUBFAL.

AGEN See Appendix C.

AGENBP See Appendix C.

AH Igniter tube or pseudo hole area.

ALPHA See Appendix D.

AMASS(I) Barrel cross-sectional area less that included in the boundary layer displacement thickness.

AMOM(I) Barrel cross-sectional area less that included in the boundary layer momentum thickness.

ARDOT Rate of increase of surface area due to advancing flame spread along radius of base pad.

AREA Surface area of a grain of propellant or black powder at the beginning of a time interval.

AREAAX Cross-sectional area of the igniter tube.

AREAC(I) Area of the chamber grid matrix $J = 1$. This includes the igniter tube when $NGR > 1$ and the entire chamber when $NGR = 1$.

AREACH Chamber cross-sectional area obtained by summing AREAG(J) from $J = 1$ to NGR .

AREAG(J) Generalized chamber grid area at grid J , assumed to be constant over the chamber length.

AREAGP(I) Area of the gap between the propellant charge and the main bed.

AREAH(I) Area of hole contained in an axial grid, connecting radial grids $J = 1$ and $J = 2$.

AREAI2(I) Area of pseudo holes connecting radial rows 2 and 3 when CHAM3 is TRUE.

AREAPB Area of the projectile base.

AREAR(I) Area of the chamber less the igniter tube and the gap between the propellant charge and the chamber wall.

AREAS Area of a sector of a circle involved in hole area calculations of Subroutine HOLES.

AREAT Area of the triangle portion of the sector involved in hole area calculations of Subroutine HOLES.

AREAXP That portion of the area of a row of holes exposed to grid 1.

ATPB Propellant burning parameter, $= AGEN * T0 + BGEN$.

BEGTC(K) See Appendix C.

BELBEG Location of the breech end of the igniter tube with respect to the primer flash hole.

BELEND Location of the end of the igniter tube nearest the projectile with respect to the breech.

BETA See Appendix C.

BGBRN Logical variable set to TRUE when base pad burns through as a result of primer jet heating.

BGEN See Appendix C.

BGENBP See Appendix C.

BGFCE Pressure exerted on center core loading by primer jet..

BGJAM Logical variable that indicates bag tie failure when set to TRUE.

BGSTP Logical variable that indicates the end of travel of center core loading after it is pushed by the primer jet to grid INBP.

BØREA Barrel cross-sectional area.

BØRED See Appendix C.

BØREDB Diameter of the barrel divided by 8.

BØRER Radius of the barrel.

BPCHG The total weight of all black powder in the charge.

BPDENS See Appendix C.

BPIGN Logical variable set .TRUE. when all black powder is ignited and is used to eliminate the call to subroutine BPFIR when it is no longer required.

BPRAD(I,J) Radius of black powder grains.

BPRADO See Appendix C.

BRNOIT Logical variable indicating base pad burnout by flame spread when TRUE.

BTUB	See Appendix C.
BURNL	The overall recession of a grain of propellant or black powder in one time interval, based on diameter and length change.
CA	See Appendix C.
CEP(K)	See Appendix C.
CF	See Appendix C.
CFR	Ratio of compressible to incompressible skin friction coefficient used in boundary layer calculations.
CGEN	See Appendix C.
CHAM1	See Appendix C.
CHAM2	See Appendix C.
CHAM3	See Appendix C.
CHTG(K)	See Appendix C.
CHWT	See Appendix C.
CØN1 CØN2 CØN3 CØN4 CØN5	Constants calculation in BARSET for use in subroutine MOTION.
CP	Specific heat at constant pressure
CT	Burn rate parameter $CGEN * T_0$.
CV	Specific heat at constant volume.
CVH	See Appendix C.
CV0	See Appendix C.
C1,C2	See Appendix C.
C3,C4	See Appendix C.
DAVG	Average chamber diameter.

DELDST	Boundary layer energy displacement thickness.
DELSTR	Boundary layer displacement thickness.
DELT	Time interval for a calculation.
DELTA	Boundary layer thickness.
DELTR	Change in hole radius in base pad due to flame spread from center.
DELTR1, DELTR2	Thickness of igniter tube wall burned in one time interval at the boundary of grids $J = 1$ and $J = 2$, respectively.
DEP(K)	See Appendix C.
DEPTH	See Appendix C.
DI(I,J)	Propellant grain perforation diameter in the chamber matrix.
DIN1	New propellant grain perforation diameter at $J = 1$ after the igniter tube has failed in compression and propellant has shifted to the axis.
DIAMBT	See Appendix C.
DIAM1	See Appendix C.
DIAM2	See Appendix C.
DIB(I)	Propellant grain perforation diameter in the barrel matrix.
DISCRP DISCRM	Discriminant of the quadratic equation solved in Subroutine DRAG, P is positive and M is negative.
DIS1	See Appendix C.
DIS2	See Appendix C.
DIS3	See Appendix C.
DIS4	See Appendix C.
DISPR	Net distance from the flash hole to the central point of the pad in the Jth grid.

DITDT(I,J) Chamber matrix propellant grain perforation diameter at the end of a time interval.

DIO See Appendix C.

DM Equivalent propellant grain diameter,
 $= (1.5 D\phi_0^2 \times L_0)^{\frac{1}{3}}$

DMPI Mass of propellant shifted from grid I during a time interval.

DMPI1 Mass of propellant shifted from grid I1 during a time interval, where I1 is the grid adjacent to I into which the propellant from I is moving.

DMPI2 The mass of propellant moving from grid I2 into grid I1 where grid I2 and I are the grids adjacent to I1.

DMPIM1 The mass of propellant moving from grid I-1 into grid I.

DMPIP1 The mass of propellant moving from grid I+1 into grid I.

D ϕ (I,J) Outside diameter of propellant located in the chamber at the beginning of a time interval.

D ϕ B(I) Outside diameter of propellant located in the barrel at the beginning of a time interval.

D ϕ N1 New propellant grain exterior diameter at J = 1 after igniter tube has failed in compression and propellant has shifted to the axis.

D ϕ TDT(I,J) Outside diameter of propellant located in the chamber at the end of a time interval.

D ϕ TM(I,J) Dummy mass flow array used in Subroutine MFLOW.

D ϕ TMBG(I) Mass of gas flowing between grids (I,2) and (I,3).

DØTMBS Cumulative amount of propellant burned up to time t, Subroutine UPDATE.

DØTMIG(I,J) Mass of gas flowing between grids (I,1) and I,2).

DØTMP(I,J) Mass of black powder burned in one grid volume.

DØTMPM Mass flow produced by primer.

DØØ See Appendix C.

DR See Appendix C.

DRAM See Appendix C.

DTDR Finite difference parameter, DELT/DR

DTDSQ Parameter used in boundary layer calculations, DELT * BORED * BORED.

DTDx Finite difference parameter, DELT/DX.

DT2BD Parameter used in boundary layer calculations, -0.5 * DELT/BORED.

DVAXIS Volume of the grids (I,1).

DVAXIT Volume of the grids (I,2).

DX Axial grid length for both chamber and barrel matrices.

ELT Minimum compressed length of center core loading as a result of gas pressure forces (FORCE).

ELTCHG Length of center core loading during the time compression is occurring.

ENDTC(K) Axial location of the end of the kth black powder igniter tube charge.

EXF Equivalent flat plate length used in boundary layer calculations.

EXPBP See Appendix C.

FAIL A logical variable set equal to .TRUE. when the center core igniter tube has failed air compression along its entire length.

FDMAX	See Appendix C.
FDPRIM	Rotating band engraving force and sliding resistance,
FIBGN1, FIBGN2	New porosity in grids $i, J = 1, i, J = 2$ after compressive igniter tube failure.
FIRE	Logical variable set to TRUE after propellant ignition occurs in at least one grid.
FM	Mach number of flow through igniter tube and pseudo holes.
FN	Number of propellant grain perforations.
FORCE	Force exerted on center core loading by primer jet and pressure gradients
FPDBEG	Axial leading edge position of the base pad nearest the breech.
FRACT(I)	Fraction of propellant not associated with the accelerated heating that occurs adjacent to an igniter tube hole (subroutine PRPFIR).
FRACT(J,I)	Fraction of the I,J grid occupied by any given black powder charge (subroutine BPINIT).
FRACT1, FRACT2	Fractional portion of a grid at each end of the main propellant charge initially occupied by propellant, subroutine CHSET.
FRCTF	See Appendix C.
FRICT	Skin friction in boundary layer calculation.
FTUBR	See Appendix C.
GAM	Ratio of specific heats, cp/cv , for propellant gas.
GAMBP	Ratio of specific heats for black powder gas.

GAP	See Appendix C.
GASMAS	Total amount of propellant gas in the system.
GJ	Product of the gravitational constant and the mechanical equivalent of heat, = 25006.
GRAV	Gravitational constant, = 32.1416.
GTRHQP	Parameter used in computing propellant movement in the barrel, $GRAV \cdot \Delta T / RHP$.
HBG	Total enthalpy at grid I,J in the chamber matrix.
HBCN	Enthalpy of the gas flowing through the "pseudo" holes connecting the second and third parallel grid matrices. Also, new gas enthalpy calculated in subroutine TUBFAL after compressive tube failure.
HBGTD(I,J)	Updated total enthalpy at chamber matrix grid I,J.
HBP	See Appendix C.
HG(I)	Total enthalpy at barrel grid I.
HIGN	Enthalpy of gas flowing between the first and second parallel one dimensional grid matrices.
HMAX	See Appendix C.
HMB	See Appendix C.
HSTAR	Reference enthalpy used in boundary layer heat transfer calculations.
HSTAT	Static enthalpy used in mass flow and boundary layer calculations.
HW	Enthalpy of the gas at the wall or gun surface temperature.
HO	Gas enthalpy at the initial temperature.
I	Axial index.

IBEGB and IBEGC Grid at which the initial portion of the charge, including the base pad, is located.

IBEGBP Axial grid at which end pad of black powder nearest the projectile begins.

IBEGFP Axial grid at which end pad of black powder nearest the breech begins.

IBEGTC(K) Axial grid in which the kth igniter tube black powder charge begins.

IDEBUG(N) See Appendix C.

IENDB Grid at which the bell igniter tube ends. (nearest the projectile).

IENDBP Axial grid at which end pad of black powder nearest the projectile ends.

IENDFP Axial grid at which end pad of black powder nearest the breech ends.

IENDTC(K) Axial grid in which the kth igniter tube black powder charge ends.

IGNIT Logical variable set time when all propellant in the chamber is ignited.

INBP See Appendix C.

INCRE Indicator to determine which direction propellant in chamber grid I will move.

IPAD(K) Logical variable set .TRUE. if the kth end pad of black powder exists.

IPATH Indicator to denote which finite difference subroutines are to be called and the sequence in which they are to be called.

IPRINT Counter used in conjunction with M/DCH to print data periodically.

IPS1, IPS2 Equal to IBEGB and IENDB, respectively, used in subroutine HOLSET.

IR Radial index.

IX	Axial index
ITUBE	See Appendix C.
J	Radial index.
JTSR	Computed shear force in end pad caused by primer jet impingement.
LAMBDA	Friction coefficient parameter used in boundary layer calculations.
LF1	First grid containing packed propellant in first slug of propellant.
LF2	Last grid containing packed propellant in first slug of propellant.
LIM1	End of the first igniter tube segment.
LIM2	End of the second igniter tube segment.
MACHSQ	Mach number squared, used in boundary layer calculations.
MFAIL(I)	Index denoting igniter tube status: 0 means not in state of failure, 1 means failure in tension, 2 means failure in compression.
MM1	First grid containing packed propellant in second slug of propellant.
MM2	Last grid containing packed propellant in second slug of propellant.
MØDCH	See Appendix C.
MØDGR	See Appendix C.
NGR	See Appendix C.
NGT	Number of grids in the igniter tube.
NGT3	Number of grids in one-third of the igniter tube.
NGX	See Appendix C.
NHØLES	See Appendix C.

NPERF See Appendix C.
 NRQWH See Appendix C.
 NX Number of barrel grids currently in the system. Also see Appendix C.
 ØMEGA Projectile rotational velocity in RPM.
 ØNED Logical variable to convert the chamber matrix into a single one dimensional matrix when all propellant is ignited and the black powder is consumed.
 PU See Appendix C.
 PCH(I,J) Pressure at chamber grid (I,J).
 PCØMP See Appendix C.
 PCØMPL Igniter tube compressive failure pressure, reduced from PCØMP by tube combustion.
 PDALST See Appendix C.
 PDRAD Radius of hole in base pad during flame spread from central grid.
 PDSHR Shear strength of base pad.
 PENET Logical variable indicating base pad penetration when TRUE.
 PEXP Pressure exponent in propellant burn rate expression, see AGEN.
 PFORCE(I,J) Reaction force transmitted through a compacted bed of propellant as a result of elasticity.
 PG(I) Pressure at barrel grid I.
 PHI(I) Porosity at barrel grid I.
 PHI1 Estimated updated porosity in the grid, adjacent to the ith grid, to determine whether there is room for propellant to flow from the ith grid.

PHIAVE Average porosity calculated by the same averaging technique used in the finite difference subroutines (see Chapt. 6 of Ref.1)

PHIBG(I,J) Propellant porosity at chamber grid I,J.
Also, combined propellant and black powder porosity in chamber finite difference subroutines.

PHIBTD(I,J) Updated chamber grid propellant porosity.

PHIBP(I,J) Black powder porosity at chamber grid I,J.

PHII Updated porosity in the i^{th} grid after propellant movement.

PHIPP Porosity in grids containing the compressed center core loading.

PHIPTD(I,J) Updated black powder porosity.

PHIRHØ Product of updated density and porosity used in chamber finite difference routines.

PHIO Initial porosity of the main charge.

PHØØP See Appendix C.

PHØØPL Igniter tube tensile failure pressure, reduced from PHØØP by tube combustion.

PINER See Appendix C.

PINT See Appendix C.

PLO See Appendix C.

PL See Appendix C.

PMASS See Appendix C.

PMØØT(I) Local mass of gas generated due to burning propellant in the barrel.

PPRØP(I,J) Local pressure loading force on a bed of packed propellant.

PRESX Mach number weighting function used in boundary layer calculations.

PRI1 Logical variable set TRUE during those time intervals when printouts are desired.

PRMCOF	See Appendix C.
PRMXP	Maximum stagnation pressure of primer jet.
PRMAS	Total mass of propellant in the system, subroutine UPDATE.
PSTAT	Static pressure used in calculation of flow through igniter tube and pseudo holes.
PZO	See Appendix C.
QBAG(I,J)	Heat lost from the gas due to heat transfer to the grains of propellant prior to propellant ignition.
QBCONS	Parameter used in the calculation of black powder temperature rise.
QCNSP	Heat lost from gas as result of heating black powder.
QCNS	Parameter required for propellant surface temperature rise calculations, $= 2 * \text{SQRT}(\text{ALPHA}/3.1416)/\text{XK}$.
QCINV	Heat transfer per ft^2 - used in PRPFIR and BNDLYR.
QDOT	Heat flux per unit area to the barrel, subroutine BNDLYR.
QPMR	Heat transfer rate to a solid surface impinged upon by the primer jet.
QL(I)	Total convective heat transfer to the barrel, computed in subroutine BNDLYR.
R	Propellant burn rate.
RANOL	See Appendix C.
RADPB	See Appendix C.
RADPS	Calculated pseudo hole radius - Subroutines CHSET and HOLESET
RAPJT	Radius of primer jet at base pad.

RAF1, Parameters used in subroutine TUBFAL that
 RAF2 represent the products of density, area,
 and porosity for $J = 1$ and 2 , respectively.

RB Radial position of Chamber grid I, J - when
 CHAM1 is true.

RDRAG Drag due to radial gas flow through the
 propellant. This parameter does not
 actually exist in the current code.

RE Reynolds number based on equivalent
 flat plate length, used in boundary
 layer calculations.

RESTRX Reynolds number evaluated with the
 reference enthalpy in boundary layer
 calculations.

RETX Reynolds number based on effective
 propellant grain diameter, used in
 propellant heat transfer calculations.

RETXBP Reynolds number based on diameter, used
 in black powder heat transfer calculations.

RETXH Reynolds number based on effective pro-
 pellant grain diameter, used in calcula-
 tionsof heat transfer to propellant opposite
 an igniter tube hole.

RGAS Gas state parameter.

RHØAVE Average of gas density used in certain
 calculations, formed with the same
 averaging technique used in the finite
 difference subroutines discussed in
 Chapt. 6 of Ref. 1.

RHØB Propellant bed density.

RHØBG(I,J) Gas density in chamber grid I, J .

RHØBGN New value of gas density in the grid
 $i, J = 1$ after compressive tube failure.

RHØBTD(I,J) Updated gas density in chamber grid,
 I, J .

RHØG(I)	Gas density in barrel grid I.
RHØP	See Appendix C.
RHØSTR	Gas density based on static pressure and reference enthalpy for boundary layer calculations.
RHØO	Initial gas density.
RØTK	See Appendix C.
RRØ	Dummy variable used for RRØBP and RRØM6.
RRØBP	See Appendix C.
RRØM6	See Appendix C.
RO	Dummy variable for ROM6 and ROBP.
ROBP	See Appendix C.
ROM6	See Appendix C.
SMLTC	Sum of lengths of center core loadings.
SPDJ	Distance from the axis of the base pad to the center of the pad in the Jth grid.
STDOFF	Distance from flash hole exit to the black powder igniter base pad.
TO	See Appendix C.
TA(I)	Total area of a row of igniter tube holes exposed up to and including the i-1 grid.
TBG(I,J)	Gas temperature at chamber grid I,J.
TBP(I,J)	Black powder surface temperature in grid I,J.
TEFF	Effective time used propellant surface temperature calculations.
TEP(K)	See Appendix C.

TERMBP	A parameter used in subroutine DRAG that reflects the number of black powder granules in a grid.
TERMP	A parameter used in DRAG that reflects the number of propellant grains in a grid.
TF	See Appendix C.
TG(I)	Gas temperature at barrel grid I.
THETA	See Appendix C.
THICK(I)	Local thickness of the igniter tube at grid I as it is reduced by combustion.
THICKT	Initial igniter tube wall thickness.
TIGN	See Appendix C.
TIGNBP	See Appendix C.
TIME	Elapsed time from the instant the black powder begins to burn.
TØPGAP	See Appendix C.
TØTM	Total mass of black powder and propellant initially in the system.
TSTAR	Gas temperature based on reference enthalpy for boundary layer calculations.
TW	See Appendix C.
TWØMT	Calculation parameter, $2 * DFLT$.
TWØGJ	Calculation parameter, $2 * GJ$.
TWØTDR	Calculation parameter, $2 * DTDR$.
TXK	Gas thermal conductivity.
TXMU	Gas viscosity.
TXNUS	Nusselt number expressing the heat transfer coefficient to a grain of propellant.

TXNUBP Nusselt number expressing the heat transfer coefficient to a granule of black powder.

TXNUSH Nusselt number expressing the heat transfer coefficient to a grain of propellant opposite an igniter tube hole.

TZC(I,J) Propellant grain surface temperature.

TZR(I,J) Surface temperature of a propellant grain adjacent to an igniter tube hole.

T2DR Calculation parameter, $0.5 * \Delta T / DR$.

T2DX Calculation parameter, $0.5 * \Delta T / DX$.

UO See Appendix C.

UAENER(I) Updated barrel area less the boundary layer energy thickness.

UAMASS(I) Updated barrel area less the boundary layer energy thickness.

UAMOM(I) Updated barrel area less the boundary layer momentum thickness.

UBG(I,J) Gas axial velocity at chamber grid I,J.

UBGAVE Average gas velocity formed with the same averaging techniques used in the finite difference routines described in Chapter 6 of Ref. 1.

UBGN New velocity for the chamber after tube failure has occurred.

UBGTD(I,J) Updated gas velocity in chamber grid I,J.

UDIB(I) Updated propellant perforation diameter in barrel grid I.

UDØB(I) Updated propellant grain exterior diameter in barrel grid I.

UDRAG(I) Drag at barrel grid I due to gas flow through a porous bed.

UG(I)	Updated gas axial velocity at barrel grid I.
UGAS	Resultant gas velocity used in propellant heating calculation adjacent to an igniter tube hol.
UHG(I)	Updated gas total enthalpy at barrel grid I.
UHØLE	Velocity of gas flowing through an igniter tube hole.
UP(I)	Propellant grain velocity in barrel grid I.
UPB(I,J)	Propellant grain velocity in chamber grid I,J.
UPBAVE	Average propellant grain velocity, calculated with the same averaging techniques used in the finite difference subroutines discussed in Chapter 6 of Ref. 1.
UPBDT(I,J)	Updated propellant grain velocity in chamber grid I,J, in Chapt. 6 of Ref. 1.
UPHI(I)	Updated porosity in barrel grid I.
UPN1	New propellant velocity on the axis (J = 1) after tube failure.
UPRM	Velocity of primer gas as it flows from the flash hole.
URHØG(I)	Updated gas density in barrel grid I.
UTDT	Scalar version of updated axial gas velocity in the chamber, used to load the UBGTD array.
UUG(I)	Updated gas velocity in barrel grid I.
UIP(I)	Updated propellant grain velocity in barrel grid I.
UXLR(I)	Updated propellant grain length in barrel grid I.

VO See Appendix C.

VBG(I,J) Gas radial velocity in chamber grid I,J.

VBGAVE Average gas velocity, calculated with the same averaging techniques used in the finite difference subroutines discussed in Chapt. 6 of Ref. 1.

VBGTD(I,J) Updated radial gas velocity in chamber grid I,J.

VBP Volume of a black powder granule.

VISTR Gas viscosity based on the reference enthalpy, as used in boundary layer calculations.

VNEW Updated propellant or black powder grain size as a result of combustion.

VØLCHG Total volume of the main propellant charge, subroutine CHSET.

VØLD Propellant or black powder grain size at the beginning of a time interval.

VP Projectile velocity.

VPRØP Volume of a propellant grain.

VTDT Scalar version of updated radial gas velocity in the chamber, used to load the VBGTD array.

VTSG Parameter used in subroutine Drag, equal to $(1.5 * DØØ ** 2 * XLO) ** 0.333 * GRAV$.

VTSGG Parameter calculated in subroutine DRAG, similar to VTSG, but based on current rather than initial propellant dimensions.

WEGHT Weight of black powder in motion after bag tie failure.

WHOLEB Logical variable set FALSE in subroutine REGRES when the propellant in at least one barrel grid forms splinters.

WHOLEC Logical variable set FALSE in subroutine REGRES when the propellant in at least one chamber grid forms splinters.

WQB See Appendix C.

XB Axial location of grid I,J.

XCL See Appendix C.

XDRAG(I,J) Drag due to axial gas flow through a porous bed at chamber grid I,J.

XIUL Mechanical equivalent of heat, 778 ft-lbf/Btu.

XK See Appendix C.

XL(I,J) Propellant grain length in chamber grid, I,J.

XLB(I) Propellant grain length in barrel grid I.

XLBAR See Appendix C.

XLBEL See Appendix C.

XLN1 New propellant grain length for J = 1 after compressive igniter tube failure.

XLTC(K) See Appendix C.

XLTOT(I,J) Updated propellant grain length in chamber grid I,J.

XLO See Appendix C.

XNTUB See Appendix C.

XQB Initial position of the projectile with respect to the barrel matrix.

XP Axial location of the projectile as a function of time.

XPMAX See Appendix C.

(2) This list is a cross-reference between the terms used in this report and Reference 1 and their corresponding FORTRAN names. The definition of these terms is given in the glossary preceding this list.

A	AGEN	B_t	BTUB
A	AGENBP	C	CGEN
A	AREAH	C_{beg}	BEGTC
a	ACC	C_{cn}	CHTC
A_{bp}	ADVBP	C_{cp}	CEP
A_{ci}	ARRAC	c_f	CF
A_{dj}	ADJUST	c_f/c_{fi}	CFR
Aener	AEENER	C_{hwt}	CHWT
$A_{ener_{t+\Delta t}}$	UAENER	CL	XLTC
$A_{ex_{i,n}}$	AREAXP	cp	CP
A_{ex_n}	TA	cv	CV
A_{fi}	AREAH	c_{vh}	CVH
A_{f1}	AFN1	c_{vo}	CV0
A_{f2}	AFN2		
A_{gpi}	AREAGP	$D, D_\emptyset$	$D_\emptyset$
A_{hi}	AI	d, d_i	DI
Amass	AMASS	d_{bt}	DIAMBT
$A_{mass_{t+\Delta t}}$	UAMASS	dc_1	DIAM1
Amom	AMQM	dep	DEP
$A_{mom_{t+\Delta t}}$	UAMQM	D	DEFF
Ap	ADV	p	
A_{ri}	AREAR	$D_{t+\Delta t}$	$D\emptyset TDT$
A_{sector}	AREAS	$d_{t+\Delta t}$	DITDT
$A_{triangle}$	AREAT	DInl	DIN1
B	BGEN	dis	DISCRM, DISCRP
B	BGENBP	$D\emptyset n1$	$D\emptyset N1$
Beg	BELBEG	dpr	DM
Bend	BELEND	Dr	RDRAG
BP_{chg}	BPCHG	d_{ram}	DRAM
		D_x	XDRAG

E	PEXP	I _{epd}	ILNDFP
e	EXPBP	Incre	INCRE
f_1	FRACT	J	XJUL
F_d	FDPRIM		
F_{dmax}	FDMAX	k	RQTK
f_{bi}	FRACT	k_{ij}	TXK
Fract1	FRACT1	k_p	XK
Fract2	FRACT2		
Frict	FRICT	L	XL
f_m	FRCTF	$L_{t+\Delta t}$	XLTDT
g	GRAV	Lim1	LIM1
gp	GAP	Lim2	LIM2
gt	TOPGAP	l_{min}	ELT
H	HBG	M	FM
$H_{t+\Delta t}$	HBGTD	M^2	MACISQ
H^*	HSTAR	m	PMAS
H_o	HO	$\dot{m}$	DOTM
Hbgn	HBGN	m_B	DOTMBS
H_{bp}	HBP	$\dot{m}_M$	DOTMB
H	HG	m_g	PRGAS
H_g	HBGN	$\dot{m}_{gp}$	DOTMGG
H_{gp}	HIGN	$\dot{m}_{ign}$	DOTMIG
H_{ign}	HMAX	$\dot{m}_p$	GASMAS
H_{max}	HMB	$\dot{m}_p$	PMDOT
H_{mb}	HSTAT	$\dot{m}_{pr}$	DOTMPM
H_s	HSTAT		
H_{stat}		N_{op}	TERMBP
I	PINER	Ngr	NGR
I_{begb}	IBEGB	N_{gt}	NGT
I_{begc}	IBEGC	N_{gt3}	NGT3
I_{endb}	IENDB	N_{gx}	NGX
I_{epb}	IBEGFP		

n_h	NRØWH	r_{bp}	BPRAD
N_{pr}	TERMP	R_o	RO
n_t	XNTUB	R_e	RETX
N_u	TXNUS	R_e^*	RESTRX
Nu_{bp}	TXNUBP	$R_{e_{bp}}$	RETXBP
Nu_h	TXNUSH	R_{eh}	RETXH
		R_{ex}	RE
p	PCH	r_h	RADHØL
p_o	Po	r_o	FTUBR
p_c	PCØMPL	r_{pb}	R
p_{co}	PCØMP	r_{ps}	RADPS
p_{force}	PFORCE	R_p	RRØ
p_g	PG		
p_{ho}	PHØØP	S_{do}	STDØFF
p_h	PHØØPL	S_{dj}	SPWJ
p_L	PL	S_{dtot}	DISPR
p_{prop}	PRØP	S_o	AREA
p_s	PSTAT		
p_x	PRESX	T	TBG
$p_{px,t,M82}$	PRMXP	T	TG
		t	TIME
Q	QDØT	T_o	TO
q	QBAG	T^*	TSTAR
Q_b	QBAG	T_{bp}	TBP
q_{bp}	QCØNBP	t_{eff}	TEFF
q_{conv}	QCØNV	t_{ep}	TEP
		t_{ep1}	TEP(1)
R	RADPB	T_{hi}	THICK
R	RGAS	T_{no}	THICKT
r	R	T_z	TZC
R_{af1}	RAF1		
R_{af2}	RAF2	u	UBG
		u_{bgn}	UBGN

u_{hi}	UHØLE	z_o	ZO
$u_{t+\Delta t}$	UBGTD	z_d	ZD
u_{ave}	UBGAVE	z_u	ZU
u_g	UG	α	ALPHA
u_p	UPB	β	BETA
$u_{pt+\Delta t}$	UPBDT	γ	GAM
u_{pave}	UPBAVE	γ_{bp}	GAMBP
u_{pn1}	UPN1	δ	DELTA
u_{pr}	UPRM	δ	DEPTH , STDOFF
u_R	UGAS	δ^*	DELSTR
		δ^{**}	DELDST
v	VBG	Δm_{bp}	DØTMP
$v_{t+\Delta t}$	VBGTD	Δm_{pi1}	DMP11
$V(t)$	VNEW	Δm_{pi}	DMPI
V_0	VØLD	Δm_{pi2}	DMPI2
V_{ave}	UBGAVE	Δm_{pi+1}	DMPIP1
V_{br}	VBP	Δm_{pi-1}	DMPIM1
V_{chg}	VØLCHG	Δr	DR
V_{new}	VNEW	Δr_1	DELTR1
V_{old}	VØLD	Δr_2	DELTR2
V_p	VP	Δr_{bp}	BURNL
V_{pr}	VPRØP	Δt	DELT
		ΔV_{axis}	DVAXIS
w_{ob}	WØB	ΔV_{axit}	DVAXIT
		ΔX	DX
X	EXF		
x	XB		
x_{cL}	XCL	λ	LAMBDA
x_{epb}	FPDBEG		
x_{lb}	XLBEL	μ	TXMU
x_{Ln1}	XLN1	$\mu(t)$	BURNL
x_{ob}	XØB	μ^*	VISTR
x_p	XP		

ϕ or ϕ_{bg}	PHIBG	ρ^*	RHØSTR
ϕ_{I1}	FIBGN1	ρ_o	RHØO
ϕ_{I2}	FIBGN2	ρ_{ave}	RHØAVE
ϕ_i t+ Δt	PHI1	ρ_b	RHØB
ϕ_{il} t+ Δt	PHI1	ρ_{bgn}	RHØBGN
$\phi_{t+\Delta t}$	PHIBTD	ρ_{bp}	BPDENS
ϕ_o	PHIO	ρ_g	RHØG
ϕ_{bp}	PHIBP	ρ_p	RHØP
ϕ_g	PHI		
ζ_s	SHAPE	θ	THETA
ρ or ρ_{bg}	RHØBG	ω	ØMEGA
$\rho_{t+\Delta t}$	RHØBTD		