AD-A013 577

PROPELLANT IGNITION AND COMBUSTION IN THE 155MM HOWITZER

E. B. Fisher, et al

Calspan Corporation

Prepared for:

Picatinny Arsenal

January 1975

Calspan

Technical Report

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

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

LIST OF FIGURES

LIST OF FIGURES (CONT'D.)

LIST OF TABLES

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.

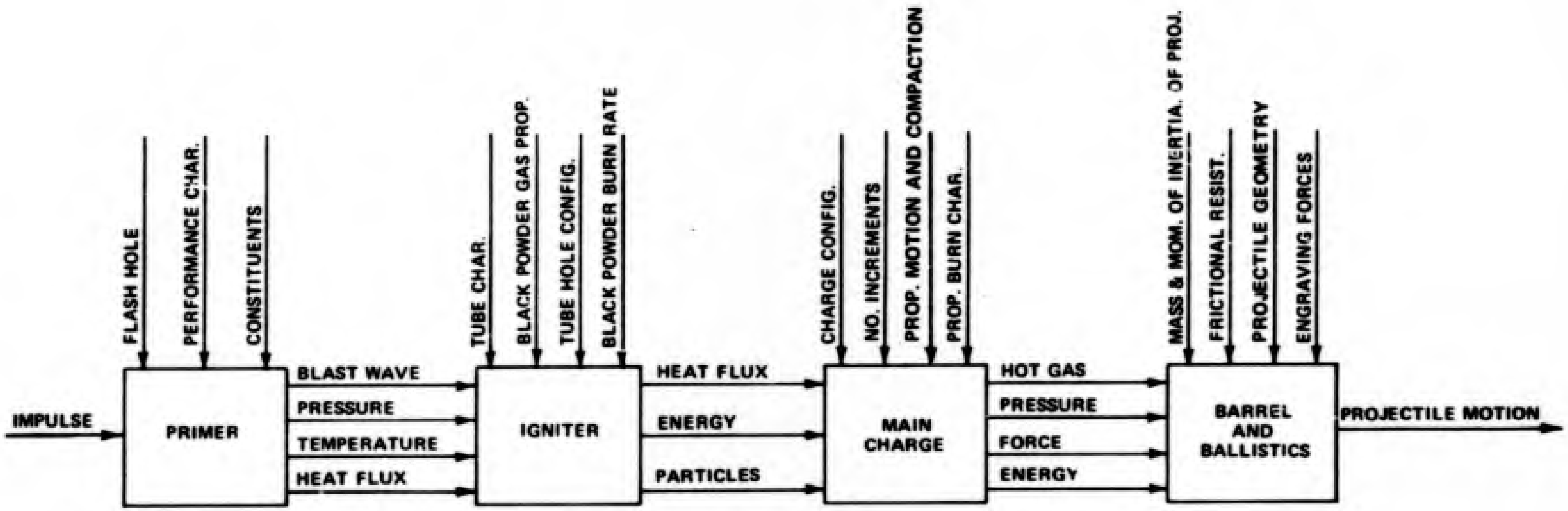
The problem of interest involves the ignition, flame propagation, propellant deflagration, and unsteady gas and propellant flow that occur during the firing of the 155mm gun. The interactions of the various components and phenomena are illustrated in Figure 1. It is important to include all these phenomena 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

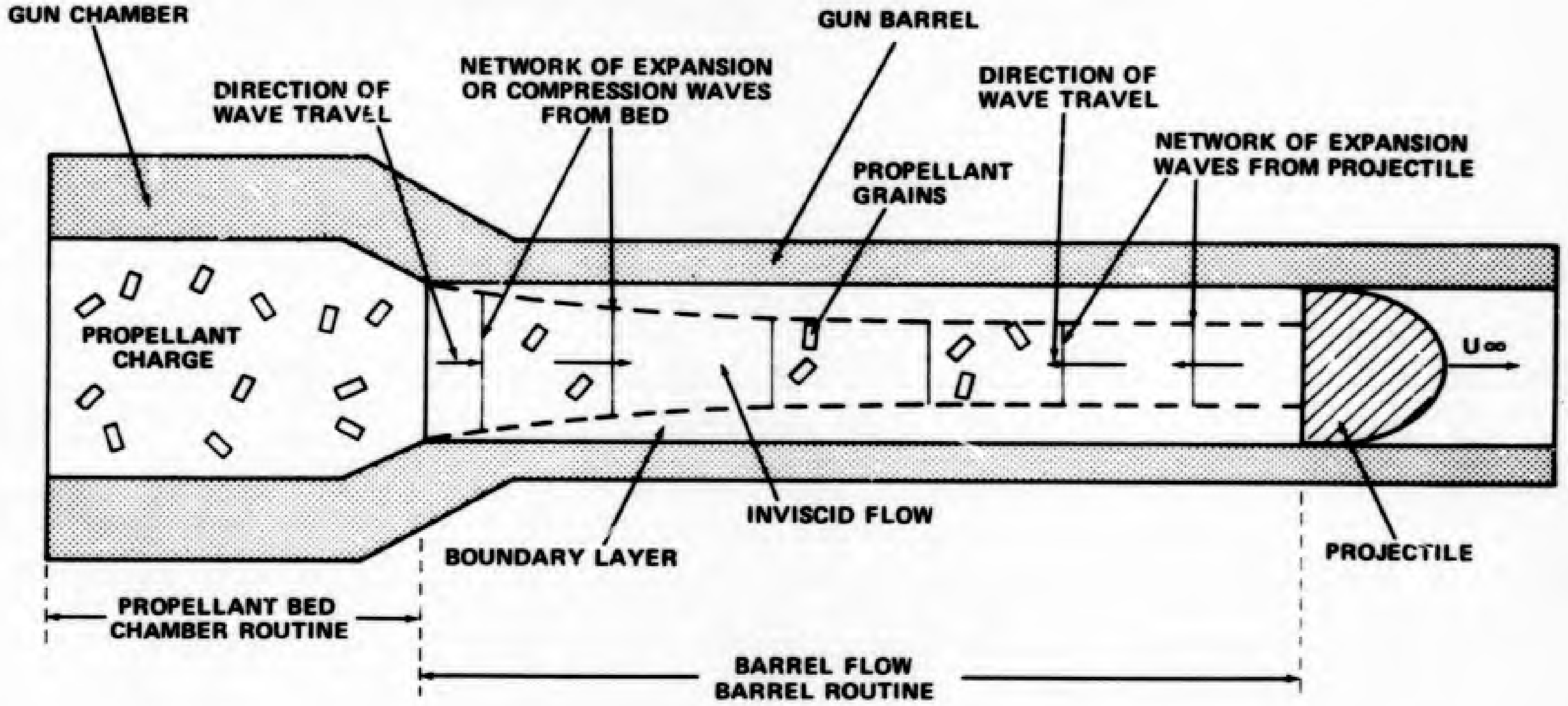

GUN CHAMBER
DIRECTION OF WAVE TRAVEL
NETWORK OF EXPANSION OR COMPRESSION WAVES FROM BED
GUN BARREL
DIRECTION OF WAVE TRAVEL
PROPELLANT GRAINS
NETWORK OF EXPANSION WAVES FROM PROJECTILE
PROPELLANT CHARGE
U∞
INVISCID FLOW
BOUNDARY LAYER
PROJECTILE
PROPELLANT BED CHAMBER ROUTINE
BARREL FLOW BARREL ROUTINE

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

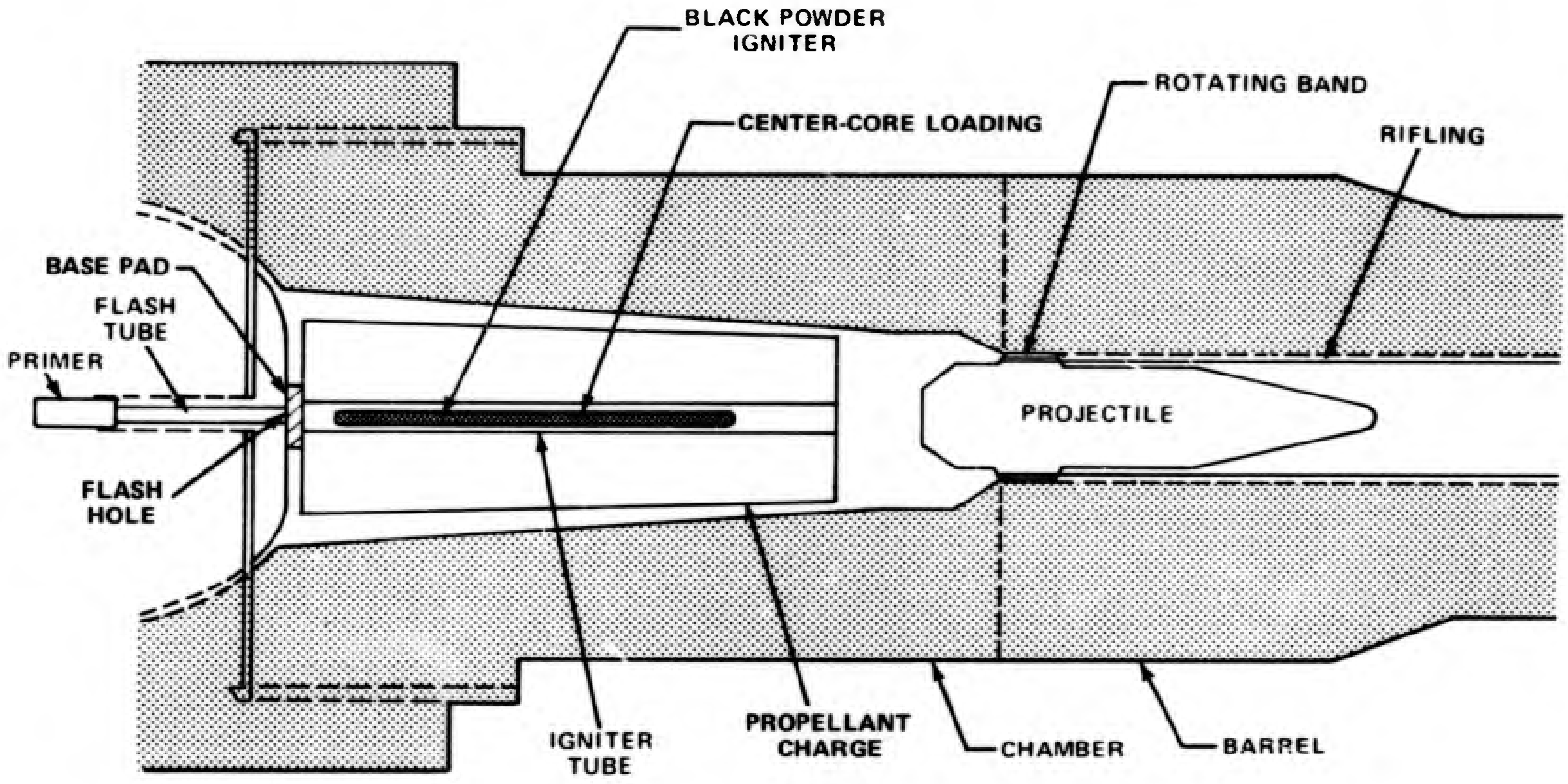


Figure 3 SCHEMATIC DIAGRAM OF THE 155 mm HOWITZER PRIOR TO FIRING

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.

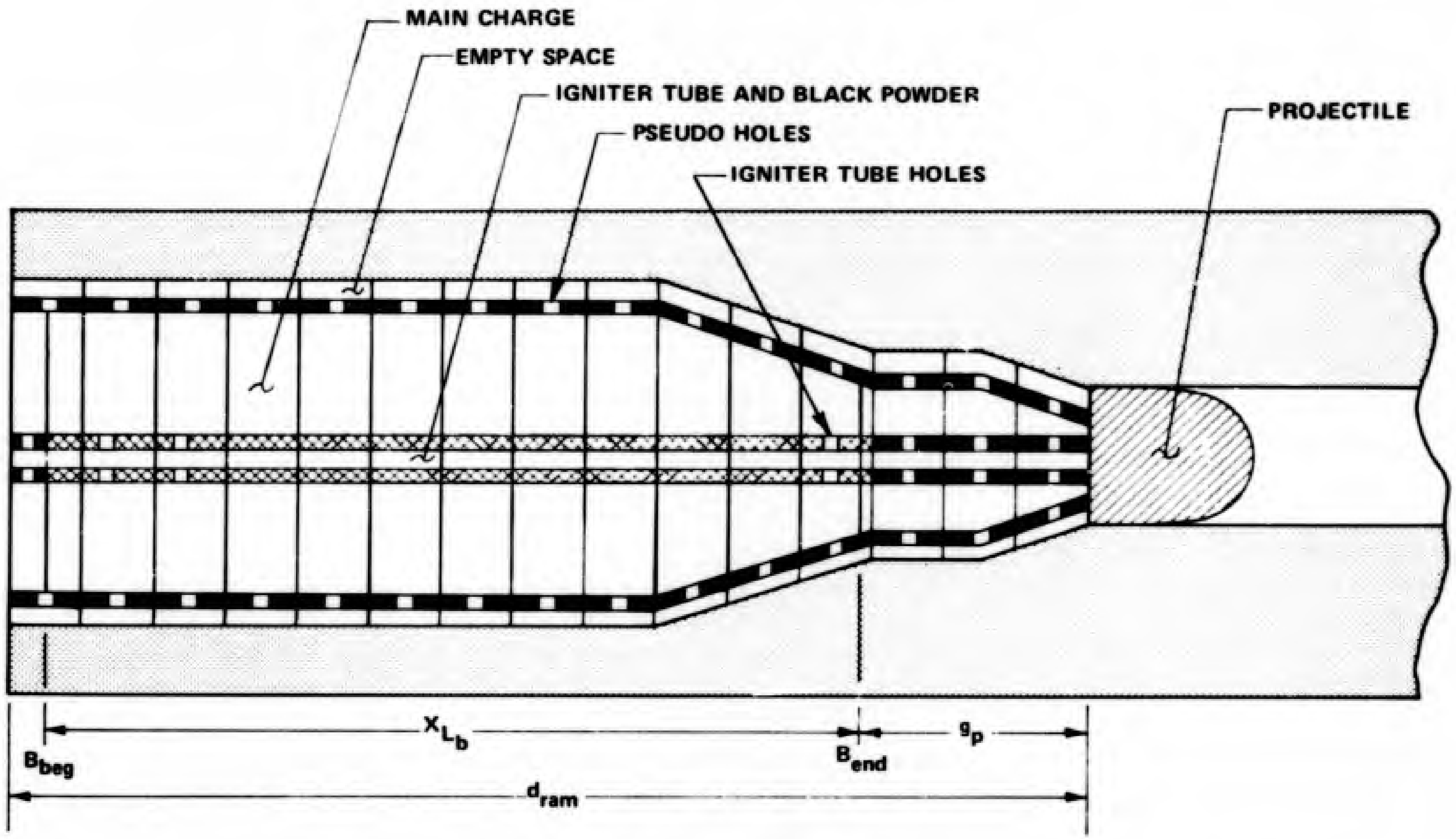


Figure 4 MULTIPLE ONE-DIMENSIONAL CHAMBER GRID NETWORK FOR THE 155 mm HOWITZER

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

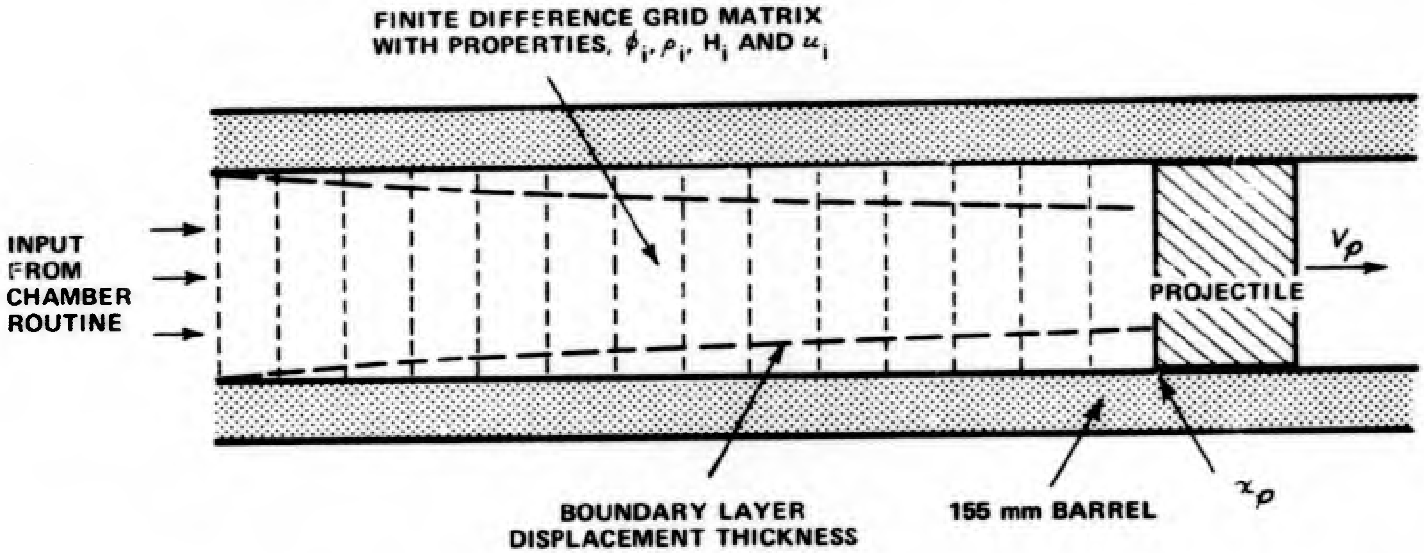


Figure 5 GRID MATRIX FOR THE BARREL OF THE 155 mm HOWITZER

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.

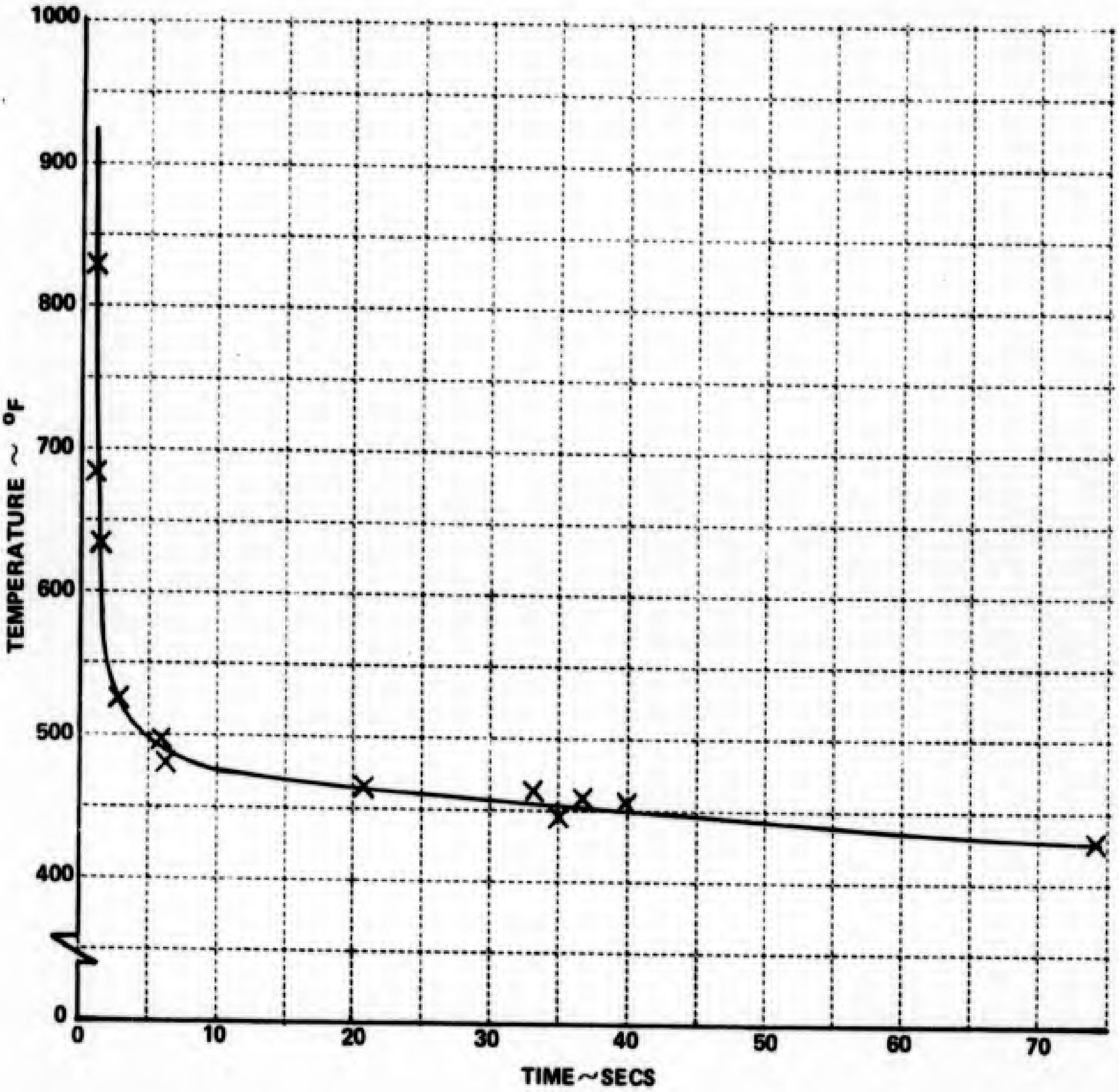


Figure 6 HOT PLATE IGNITION TESTS FOR M30 PROPELLANT

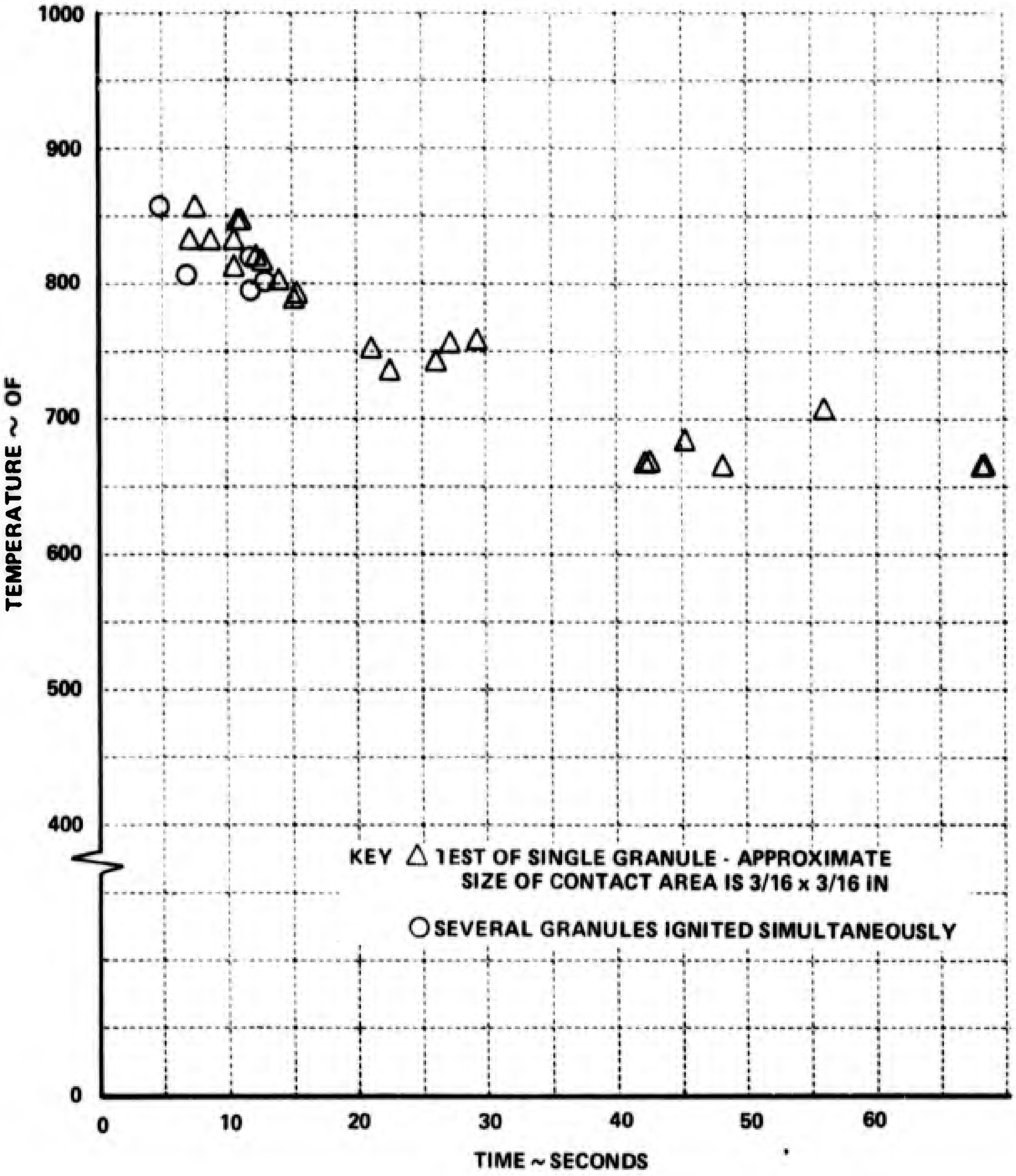


Figure 7 HOT PLATE IGNITION TESTS FOR BLACK POWDER

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 M82 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 standoff 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 standoff 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

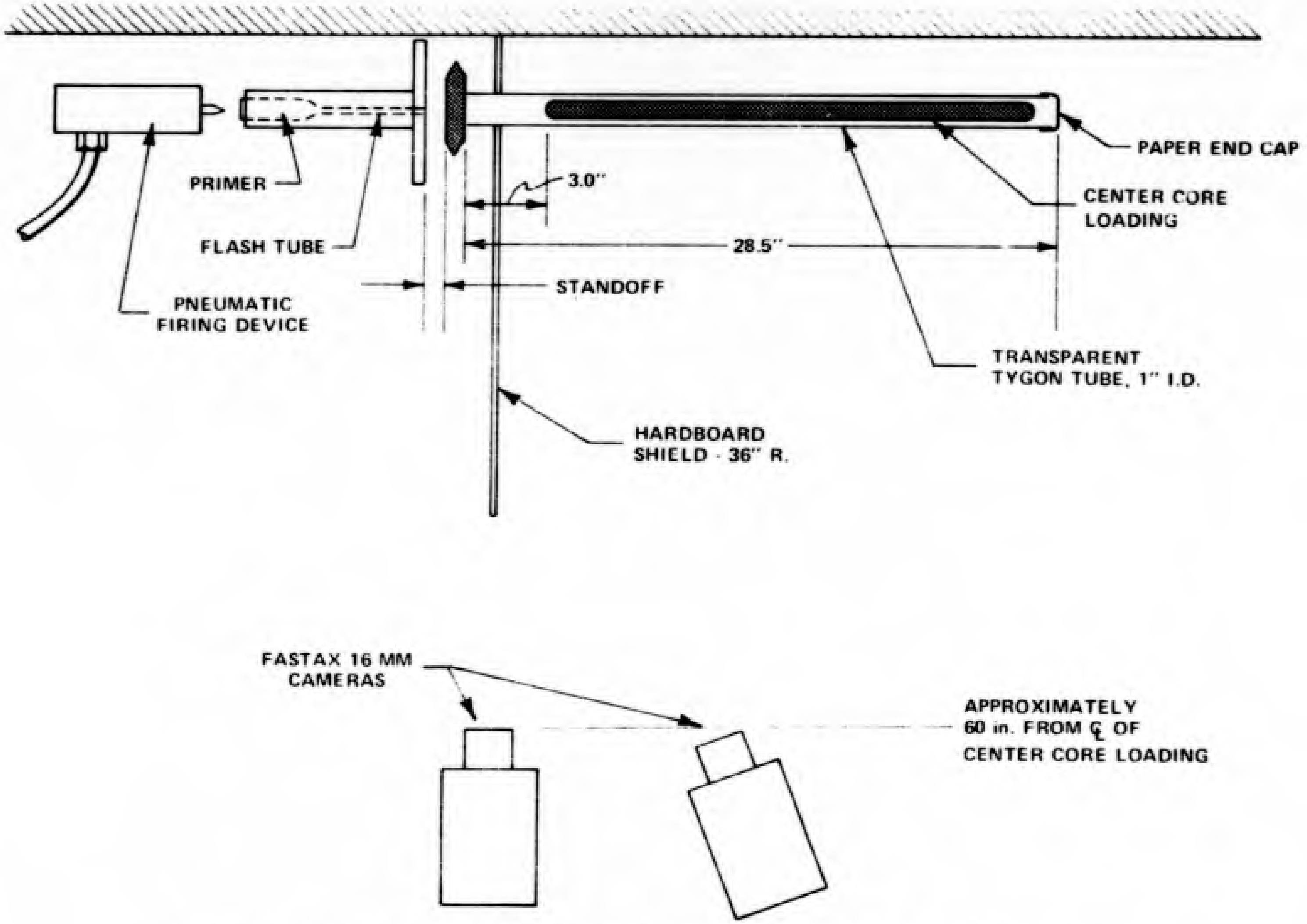


Figure 8 TEST ARRANGEMENT FOR CENTER CORE IGNITION STUDIES

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

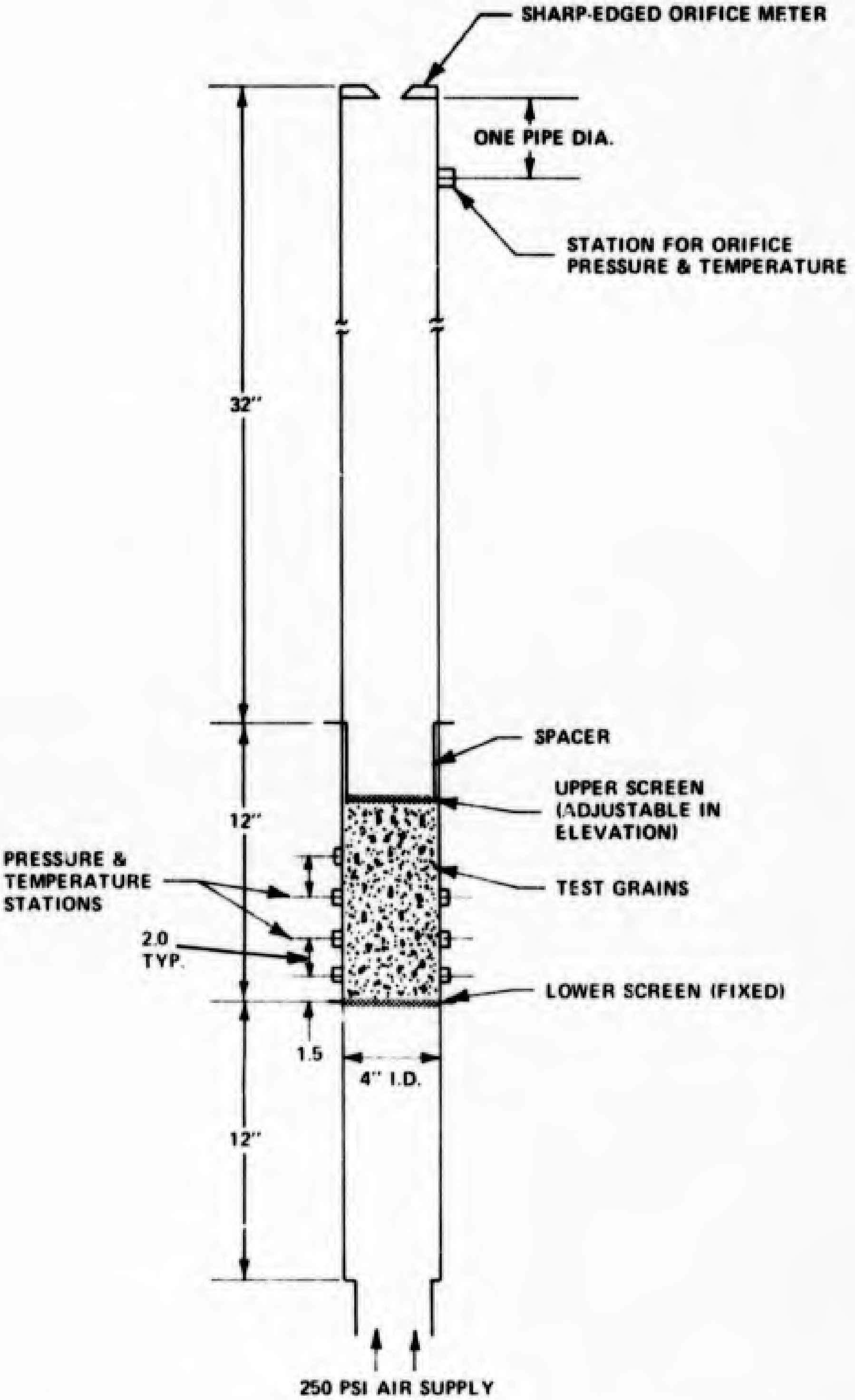

SHARP-EDGED ORIFICE METER
ONE PIPE DIA.
STATION FOR ORIFICE PRESSURE & TEMPERATURE
32"
SPACER
UPPER SCREEN (ADJUSTABLE IN ELEVATION)
12"
PRESSURE & TEMPERATURE STATIONS
TEST GRAINS
2.0 TYP.
LOWER SCREEN (FIXED)
1.5
4" I.D.
12"
250 PSI AIR SUPPLY

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:

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

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, = $\emptyset$ times actual velocity, fps

ρ is the fluid density, lbs/ft^2

$\emptyset$ is the porosity

$\emptyset_s$ 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	Δ P 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.

<u>Chamber Routine</u>

(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 (breech).

(10) BSURA2 Performs one-dimensional finite difference calculations in the igniter tube at the breech.

(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 breech end.

(15) BSURT3 Performs one-dimensional finite difference calculations for the void region between the propellant charge and the chamber wall at the breech 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:

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

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

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

and

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

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_o is the flash tube radius

Then the shear force per unit length of jet circumference is

$$F_\tau = p_{p_{x,t,M82}} y/2$$

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, PENFT, 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-psi^3. 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 $P_{P_{x,t}} = r\, P_{P_{x,t,M82}}$ to permit the representation of primers other than the M82. The quantity r, which is given by the input quantity PRMCOF, 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 $P_{P_{x,t}}$ 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 moveo 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

$$\ell_{min} = \Sigma \ell \left(1 - 0.03145 (F - 10)^{0.545} \right)$$

where $\Sigma \ell$ 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, $(1-\phi)$ 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, $\emptyset_o$, i.e., bed porosity was permitted to be as low as $0.2\emptyset_o$ 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:

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

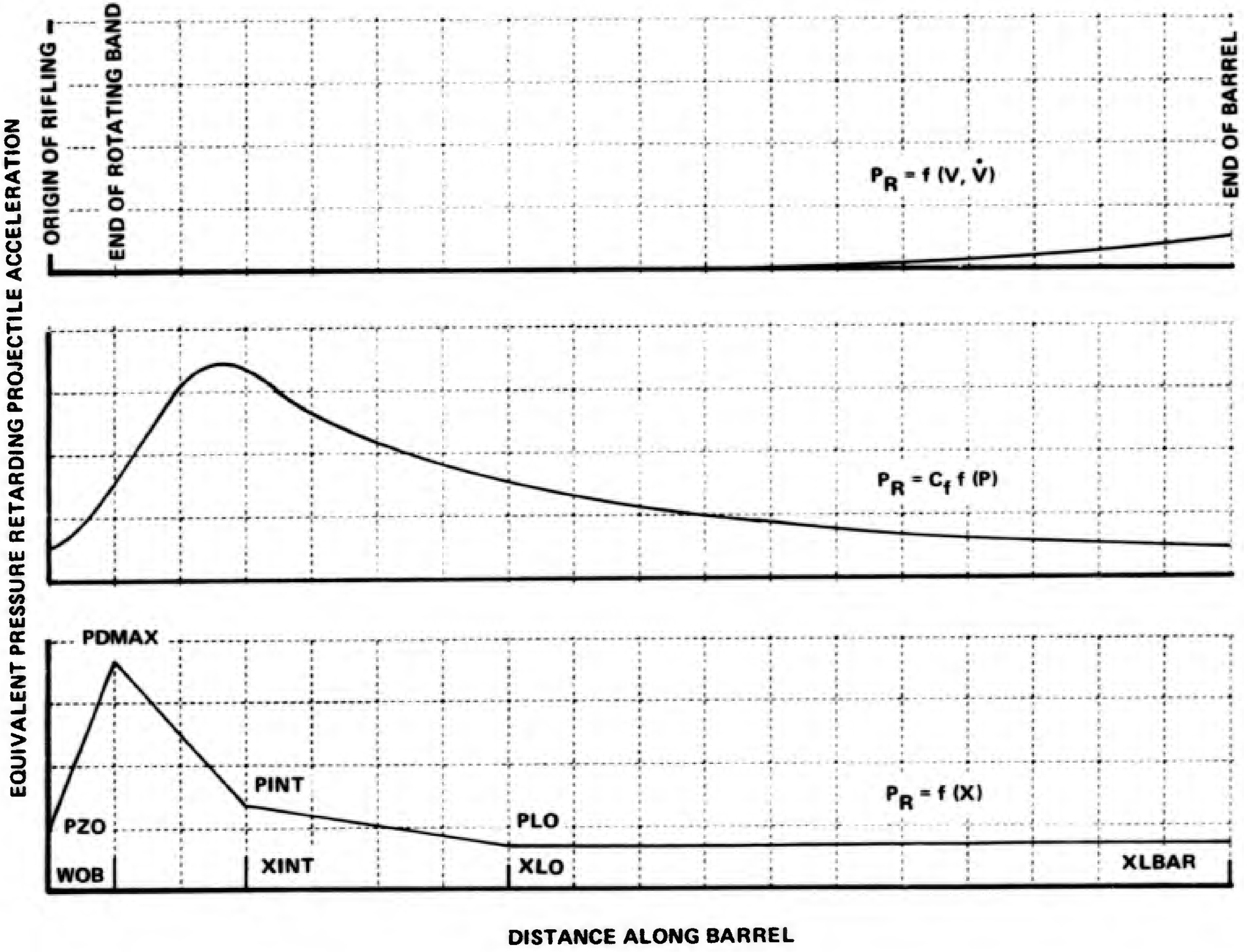


Figure 10 BARREL RESISTANCE FUNCTIONS BUILT INTO THE 155 mm HOWITZER CODE

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' Handboom, 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/ft^3
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, $= \phi$ 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 BREECH - 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	↓	↓	↓	↓	↓	↓	↓	61300	6800	
R-3	22.0	↓	↓	↓	↓	↓	↓	↓	57300	2900	
R-4	25.0	34.0	↓	↓	↓	↓	↓	↓	47000	1100	
R-5	↓	30.0	↓	↓	↓	↓	↓	↓	68000	6400	
R-6	↓	32.5	1.5	↓	↓	↓	↓	↓	80000+	6000	
R-7	↓	↓	6.0	↓	↓	↓	↓	↓	72500	11800	
R-8	↓	↓	0	20.0	↓	↓	↓	↓	64700	10800	
R-9	↓	↓	0	3.0	MEDIUM[2]	↓	↓	↓	58300	4700	
R-10	↓	↓	1.5	↓	MEDIUM[2]	↓	↓	↓	57700	5000	
R-11	↓	↓	1.5	↓	LOW[3]	↓	↓	↓	53500	4300	
R-12	↓	↓	0	↓	MEDIUM[2]	20,000	↓	↓	76700	4800	
R-13	↓	↓	1.5	↓	↓	2,700	↓	↓	53800	4500	
R-14	↓	↓	↓	↓	↓	10,000	-40	↓	-	-	
R-15	↓	↓	↓	↓	↓	↓	340	↓	58800	5000	
R-16	↓	↓	↓	0.2	↓	↓	70	↓	53000	1200	
R-17	↓	↓	↓	3.0		↓	↓	↓	58000	5500	CB
B-1	↓	↓	0	↓	HIGH[1]	↓	↓	↓	72100	17500	
C-1	↓	↓	0	↓	↓	↓	↓	2.15	?	4600	
C-2	↓	↓	1.5	↓	↓	↓	↓	1.0	57000	0	
C-3	↓	30.0	1.5	↓	↓	↓	↓	↓	62500	1000	
C-4	↓	30.0	0	↓	↓	↓	↓	↓	62400	2700	
C-5	↓	34.0	1.5	↓	MEDIUM[2]	↓	↓	↓	48000	1500	
C-6	↓	30.0	0	↓	↓	↓	↓	↓	62500	2400	FIT
C-7	↓	30.0	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

CB – 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

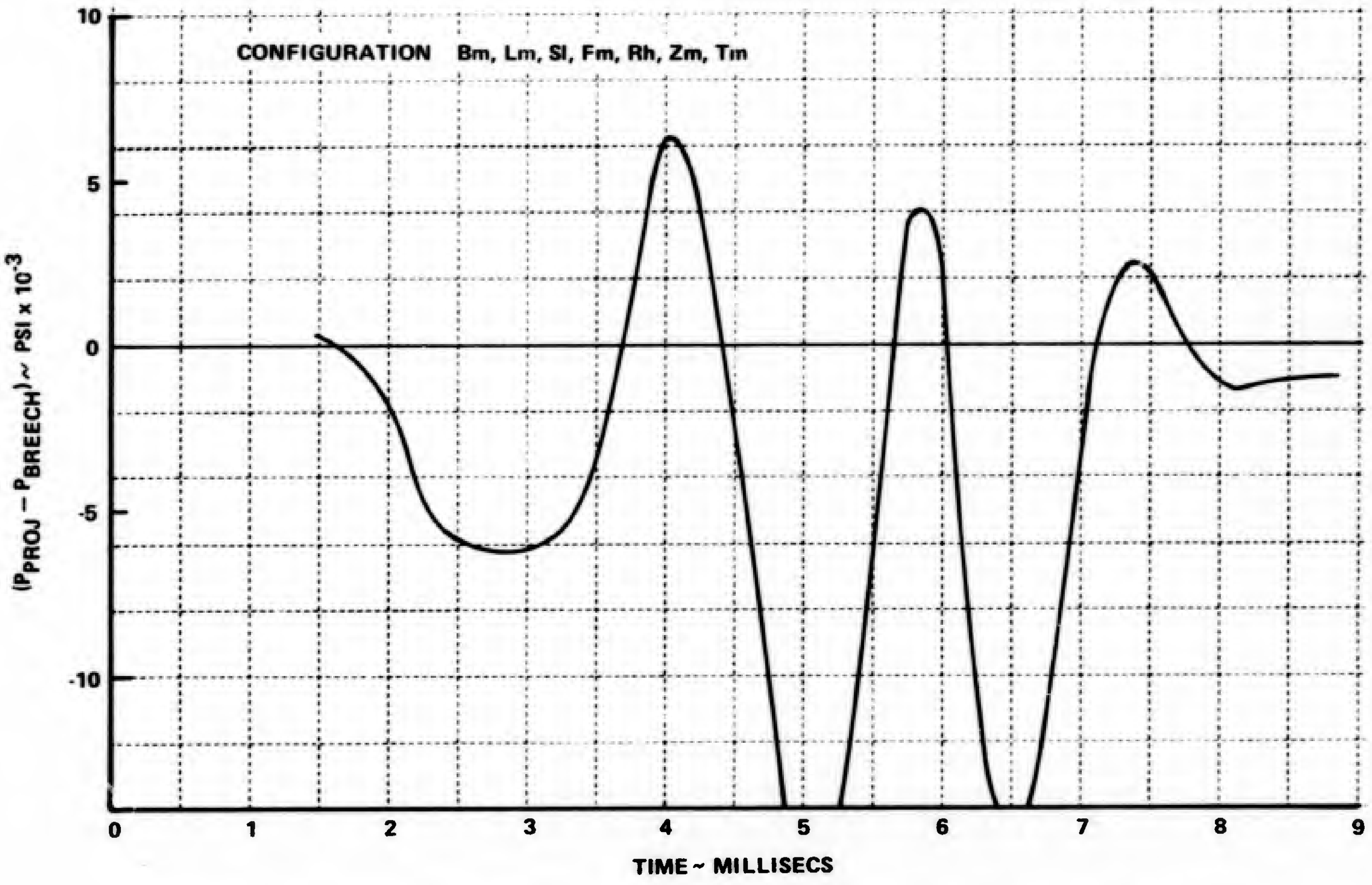


Figure 11 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-2

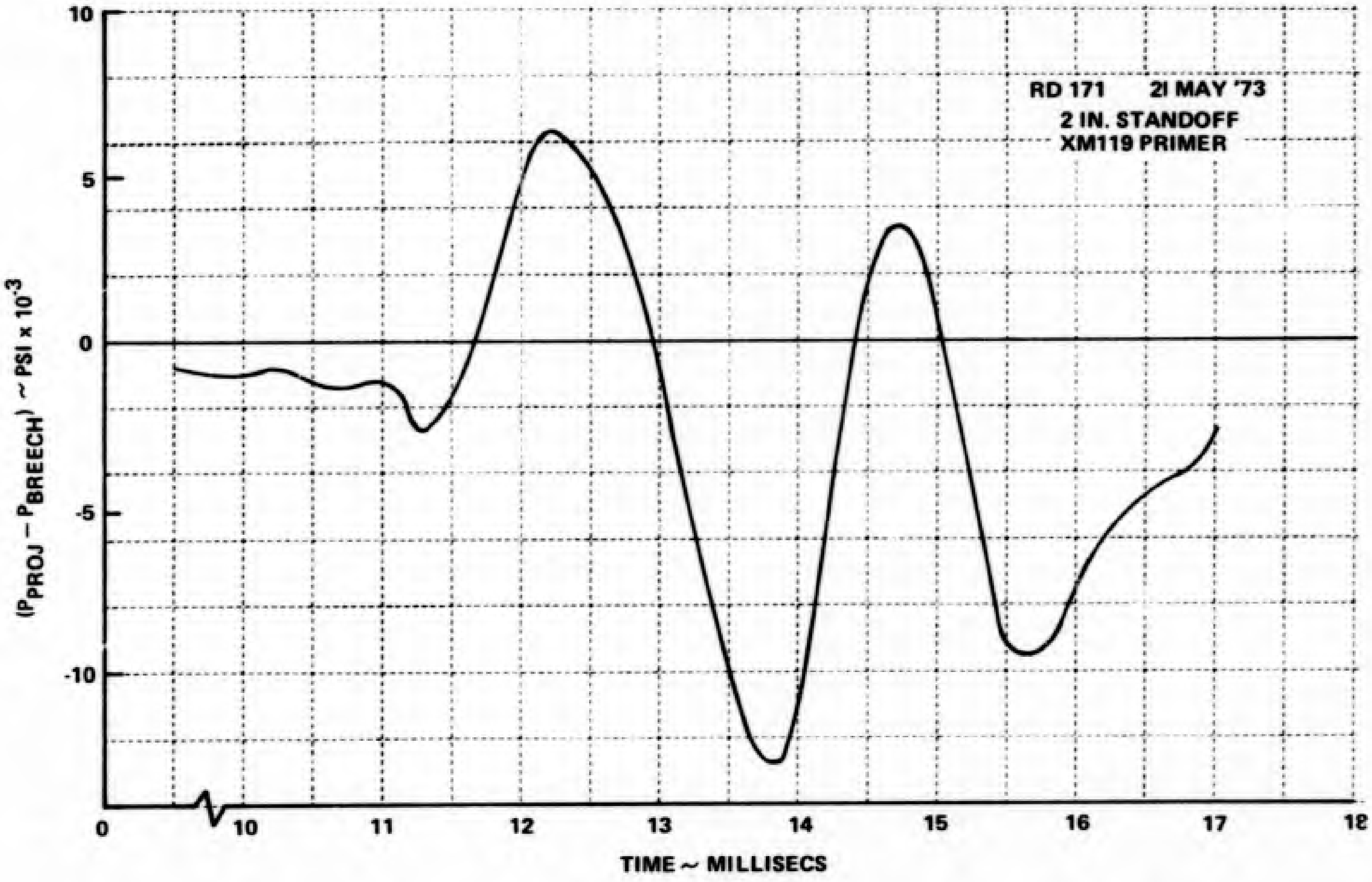


Figure 12 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ FIRING DATA

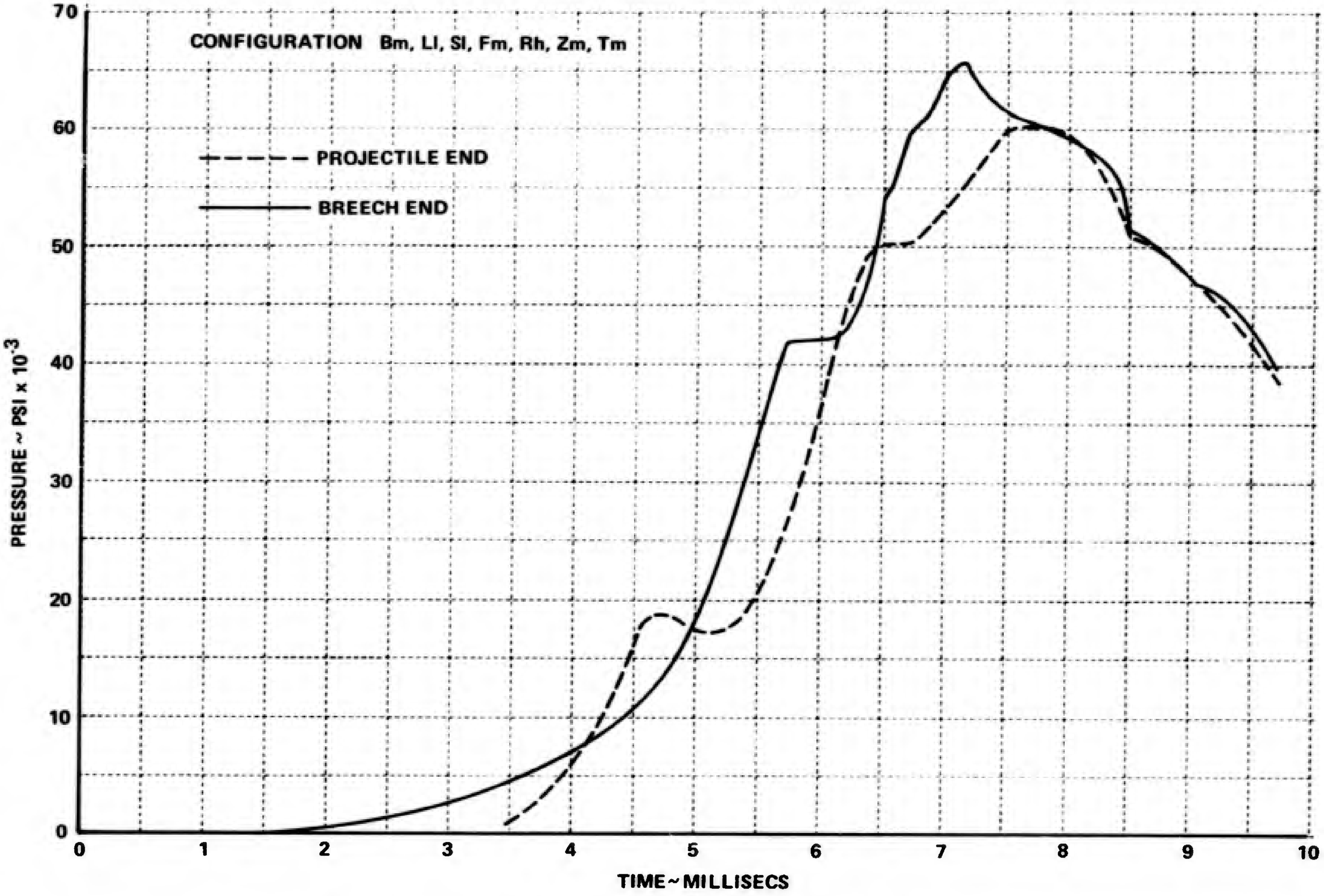


Figure 13 CHAMBER PRESSURE HISTORIES ~ RUN R-5

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.

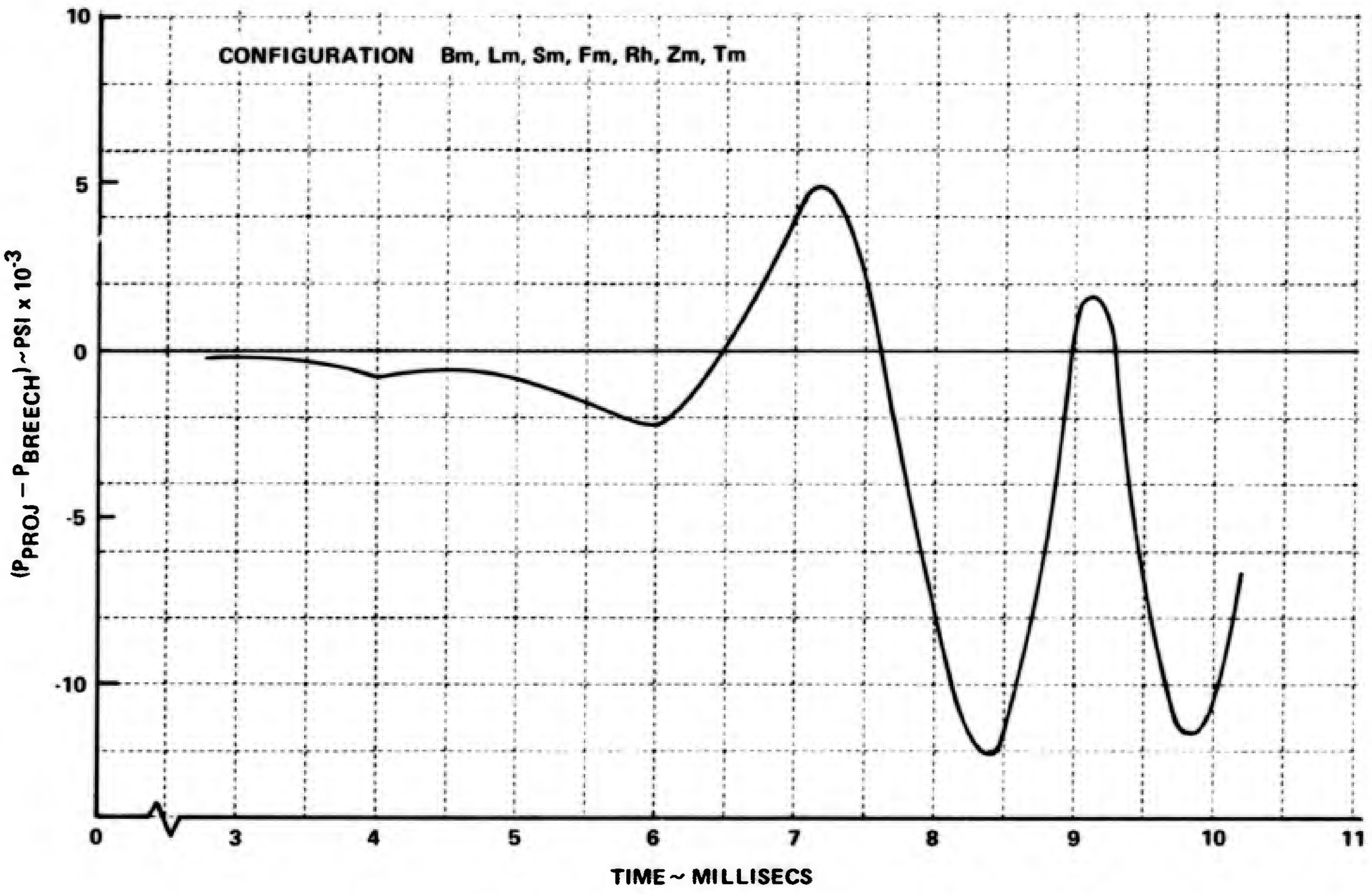


Figure 14 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-6

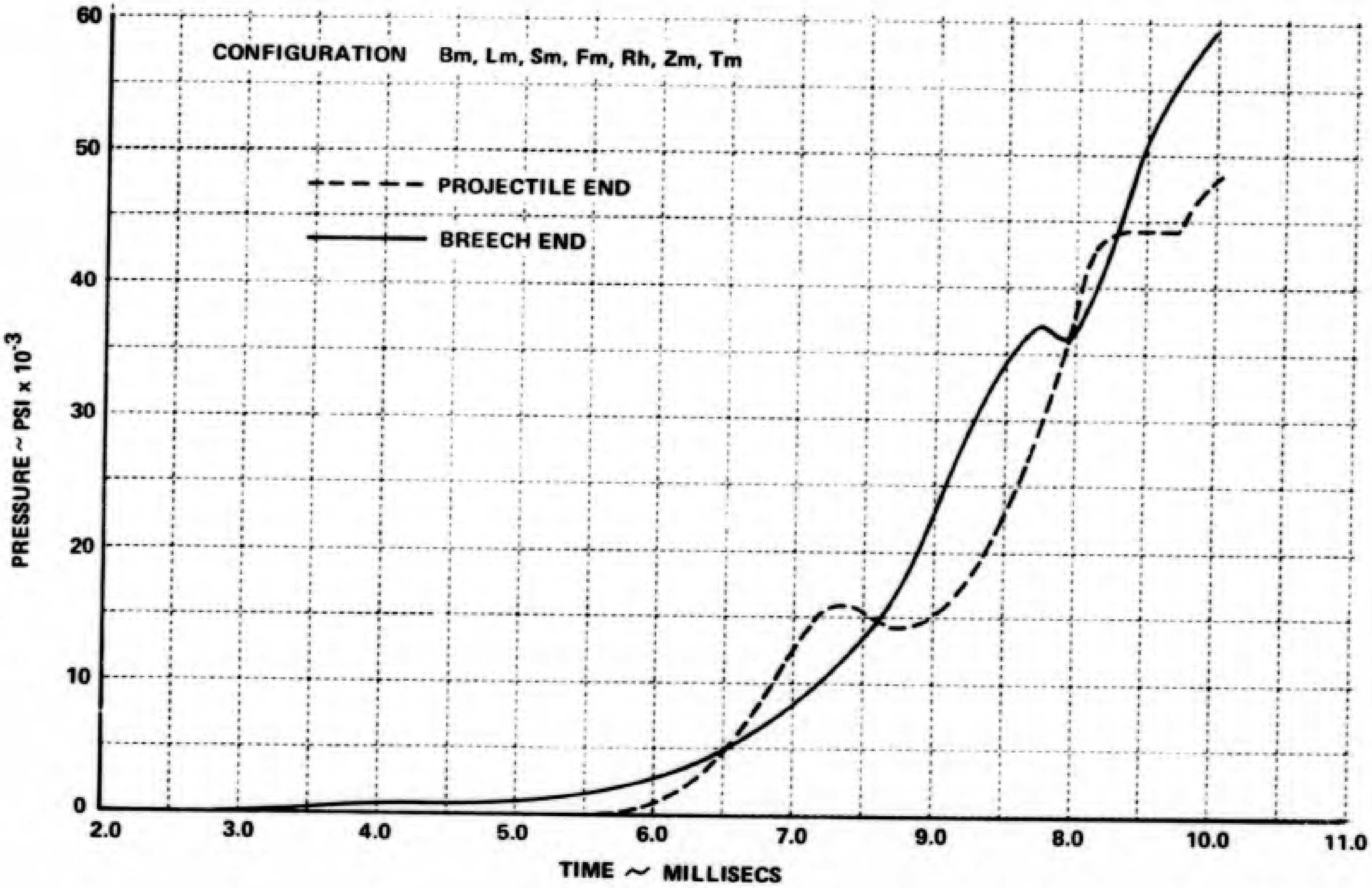


Figure 15 CHAMBER PRESSURE HISTORIES ~ RUN R-6

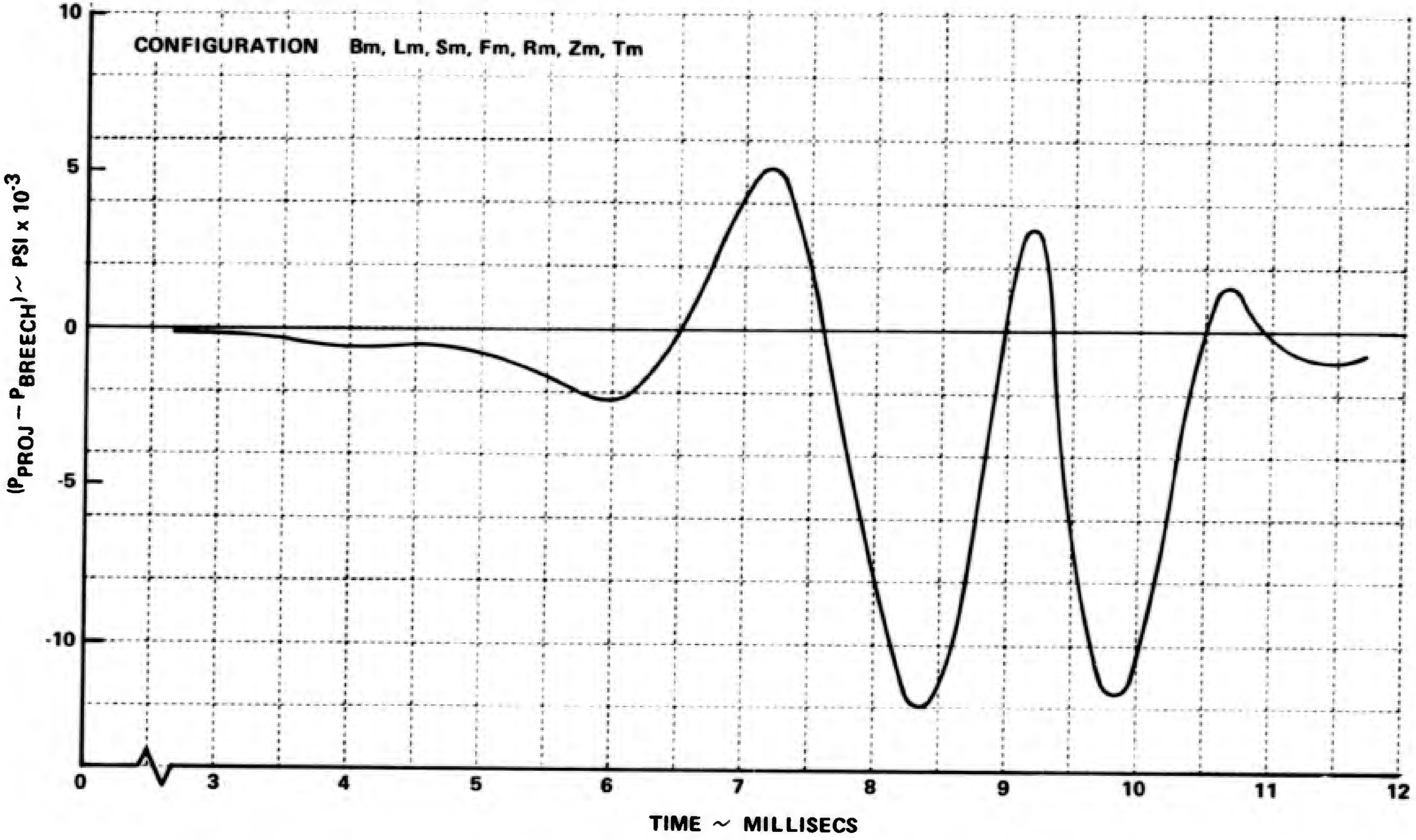


Figure 16 HISTORY OF PRESSURE DIFFERENCE AT CHAMBER ENDS ~ RUN R-10

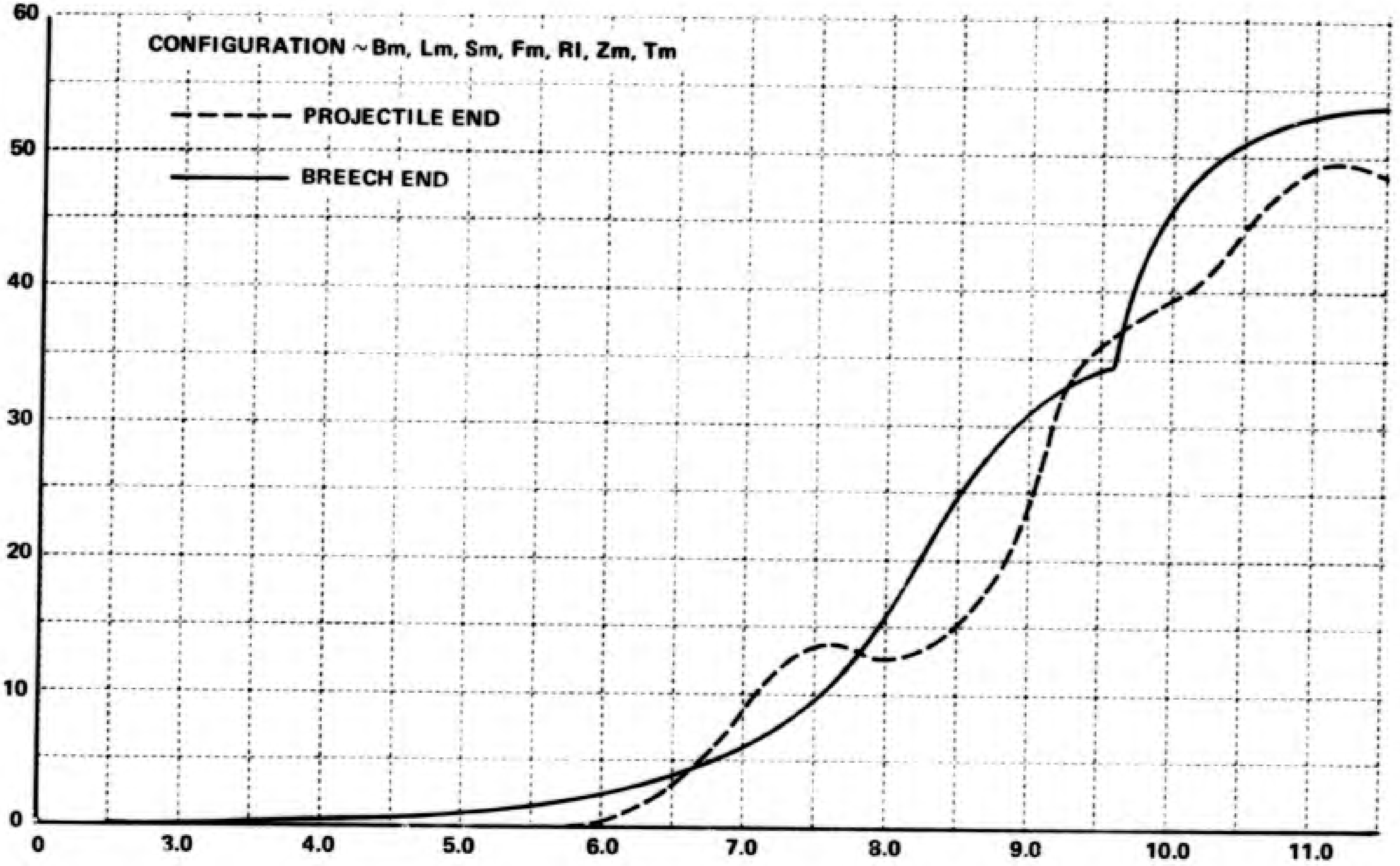


Figure 17 CHAMBER PRESSURE HISTORIES ~ RUN R-11

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

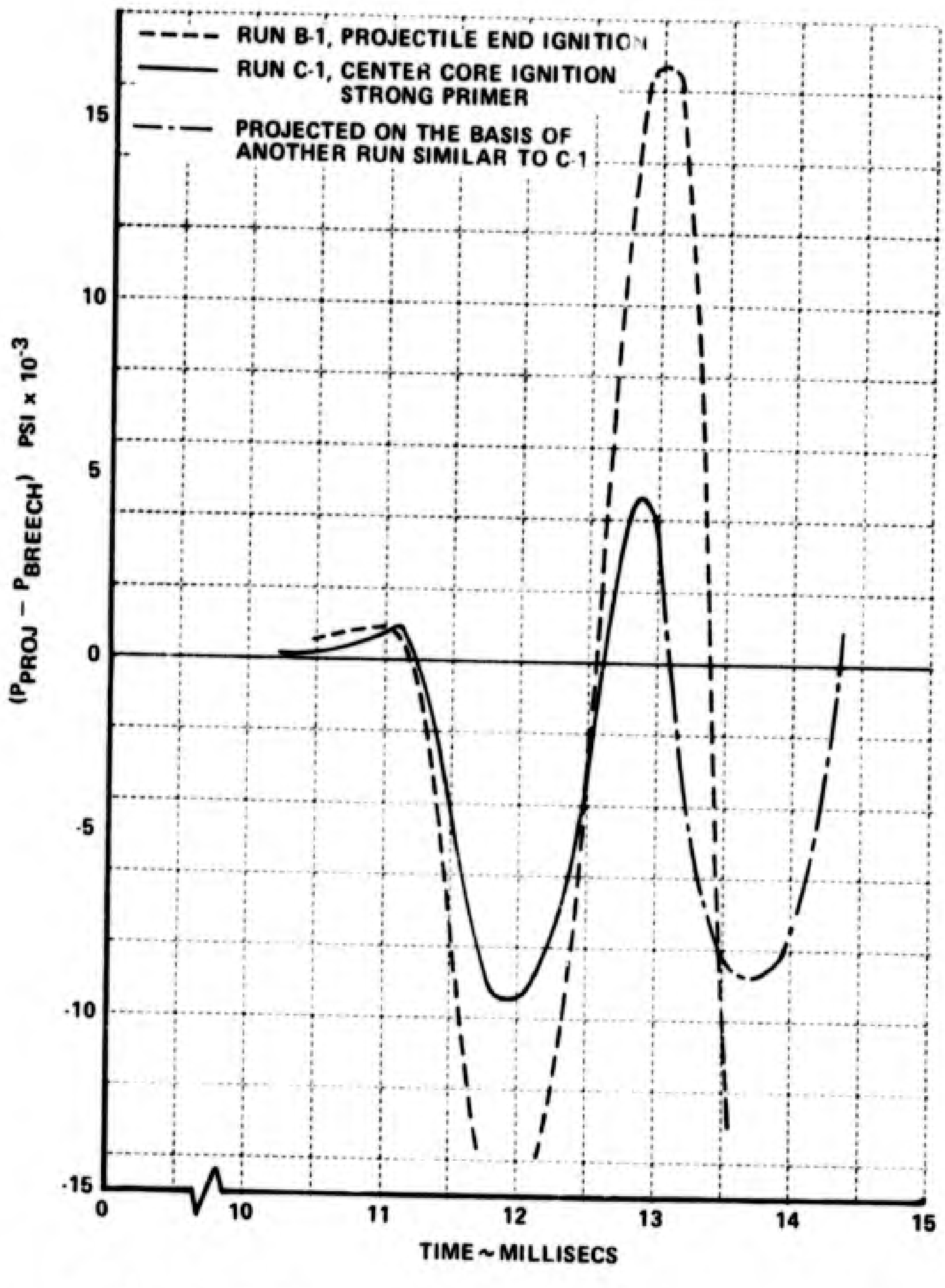


Figure 18 COMPARISON OF CHAMBER PRESSURE WAVES FOR END IGNITION AND ABNORMAL CENTER CORE IGNITION

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 millisecondsafter 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 interupts 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 (ΔP = 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 Δ p 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 nas 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.

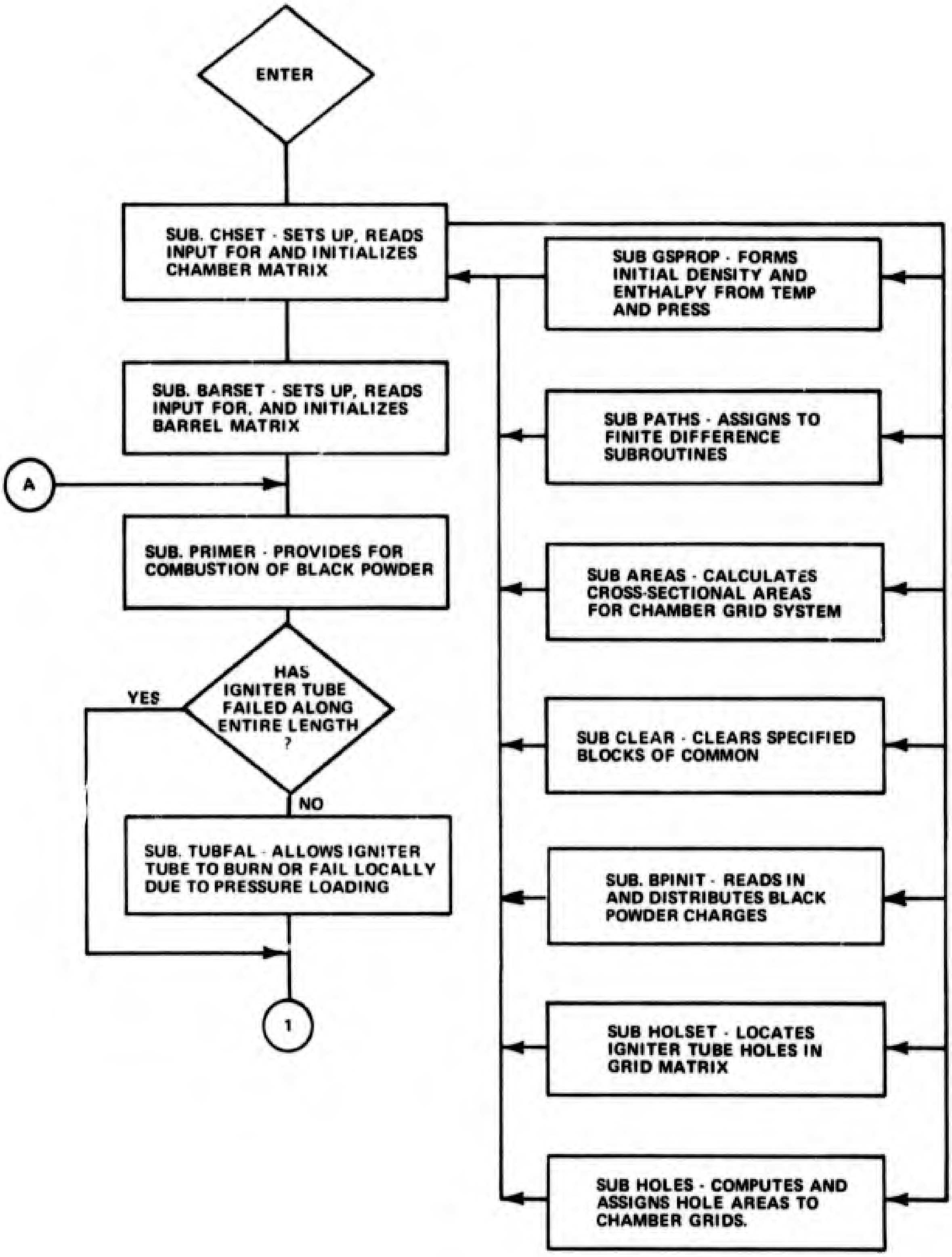
3. Cunningham, R.G., "Orifice Meters with Supercritical Compressible Flow," ASME Trans., July 1951.

4. Chemical Engineers' Handbook, Robert H. Perry, ed. Fifth Edition, McGraw-Hill Book Co., 1973.

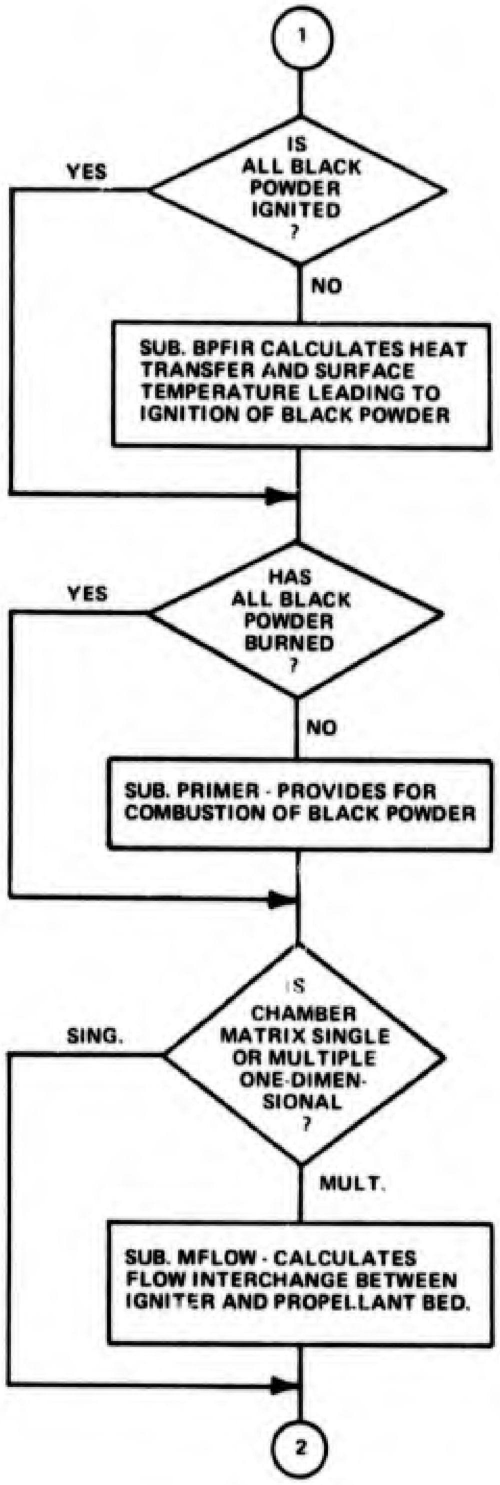
5. Love, E.S., et. al., "Experimental and Theoretical Studies of Axi-Symmetric Free Jets," NASA TR R-6, 1959.

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.

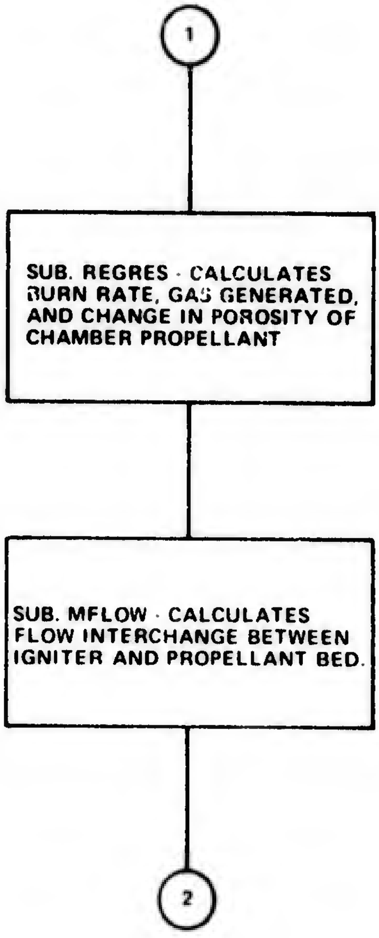
APPENDIX A
PROGRAM FLOW CHART

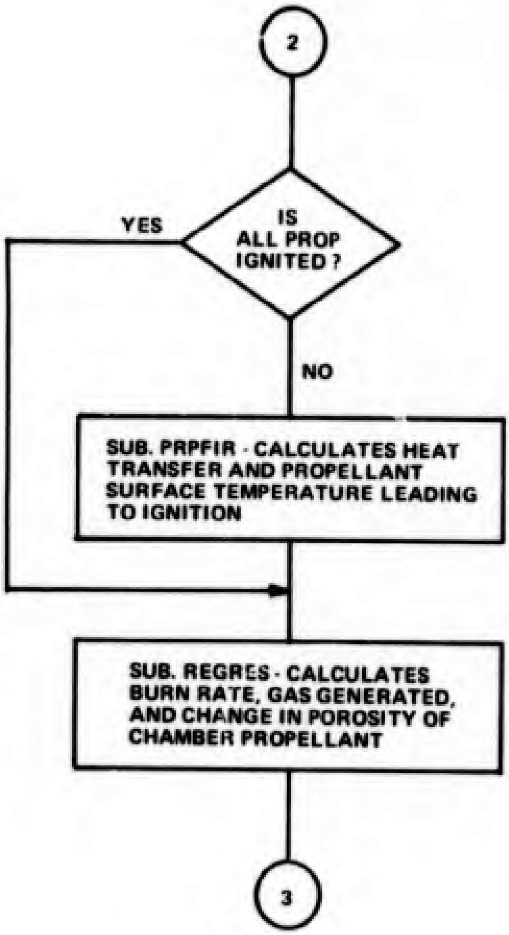


ENTER
SUB. CHSET - SETS UP, READS INPUT FOR AND INITIALIZES CHAMBER MATRIX
SUB. BARSET - SETS UP, READS INPUT FOR, AND INITIALIZES BARREL MATRIX
A
SUB. PRIMER - PROVIDES FOR COMBUSTION OF BLACK POWDER
HAS IGNITER TUBE FAILED ALONG ENTIRE LENGTH ?
YES
NO
SUB. TUBFAL - ALLOWS IGNITER TUBE TO BURN OR FAIL LOCALLY DUE TO PRESSURE LOADING
1
SUB GSPROP - FORMS INITIAL DENSITY AND ENTHALPY FROM TEMP AND PRESS
SUB PATHS - ASSIGNS TO FINITE DIFFERENCE SUBROUTINES
SUB AREAS - CALCULATES CROSS-SECTIONAL AREAS FOR CHAMBER GRID SYSTEM
SUB CLEAR - CLEARS SPECIFIED BLOCKS OF COMMON
SUB. BPINIT - READS IN AND DISTRIBUTES BLACK POWDER CHARGES
SUB HOLSET - LOCATES IGNITER TUBE HOLES IN GRID MATRIX
SUB HOLES - COMPUTES AND ASSIGNS HOLE AREAS TO CHAMBER GRIDS.

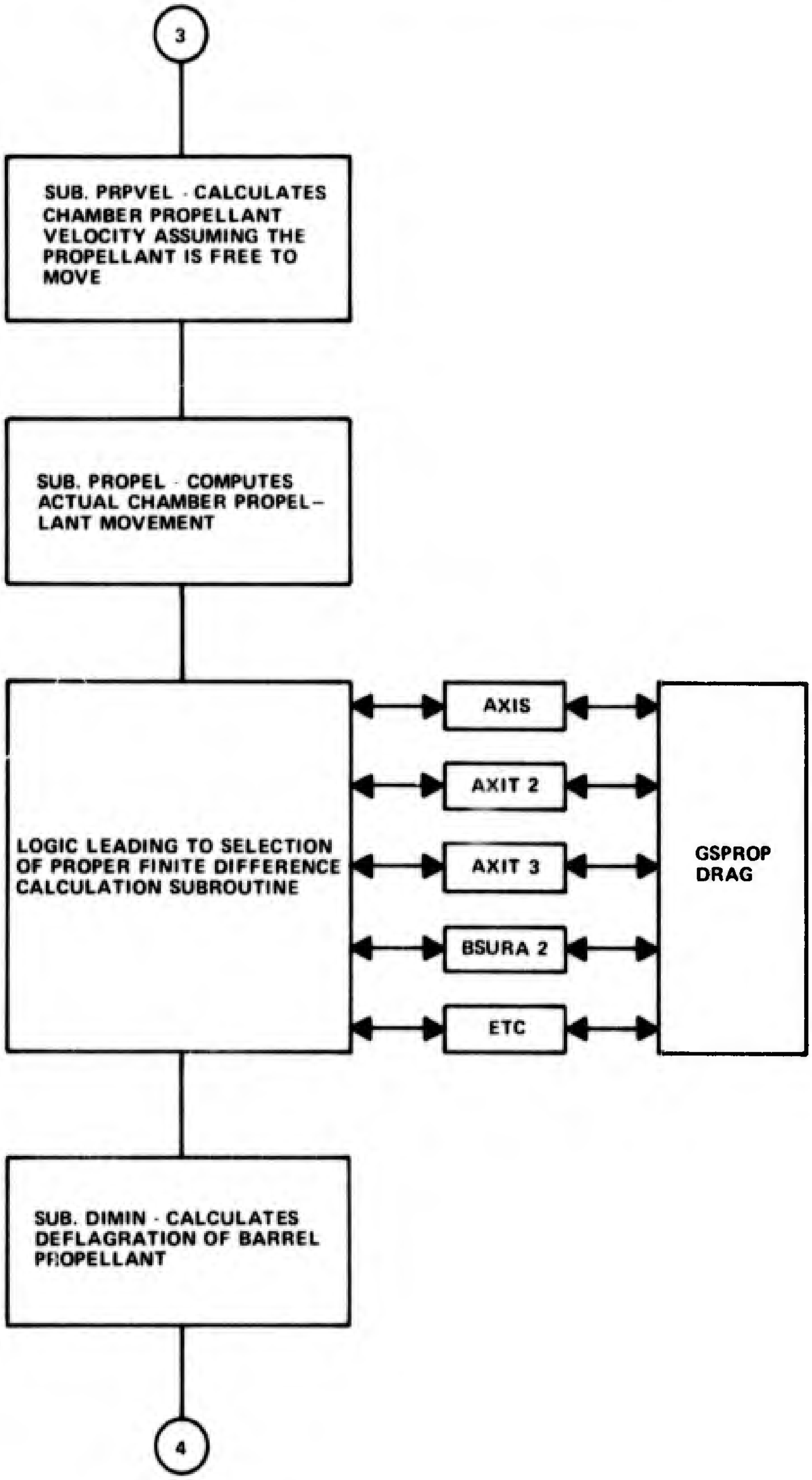

1
IS ALL BLACK POWDER IGNITED ?
YES
NO
SUB. BPFIR CALCULATES HEAT TRANSFER AND SURFACE TEMPERATURE LEADING TO IGNITION OF BLACK POWDER
HAS ALL BLACK POWDER BURNED ?
YES
NO
SUB. PRIMER - PROVIDES FOR COMBUSTION OF BLACK POWDER
IS CHAMBER MATRIX SINGLE OR MULTIPLE ONE-DIMEN- SIONAL ?
SING.
MULT.
SUB. MFLOW - CALCULATES FLOW INTERCHANGE BETWEEN IGNITER AND PROPELLANT BED.
2

PROGRAM FLOW CHART (CONT.)

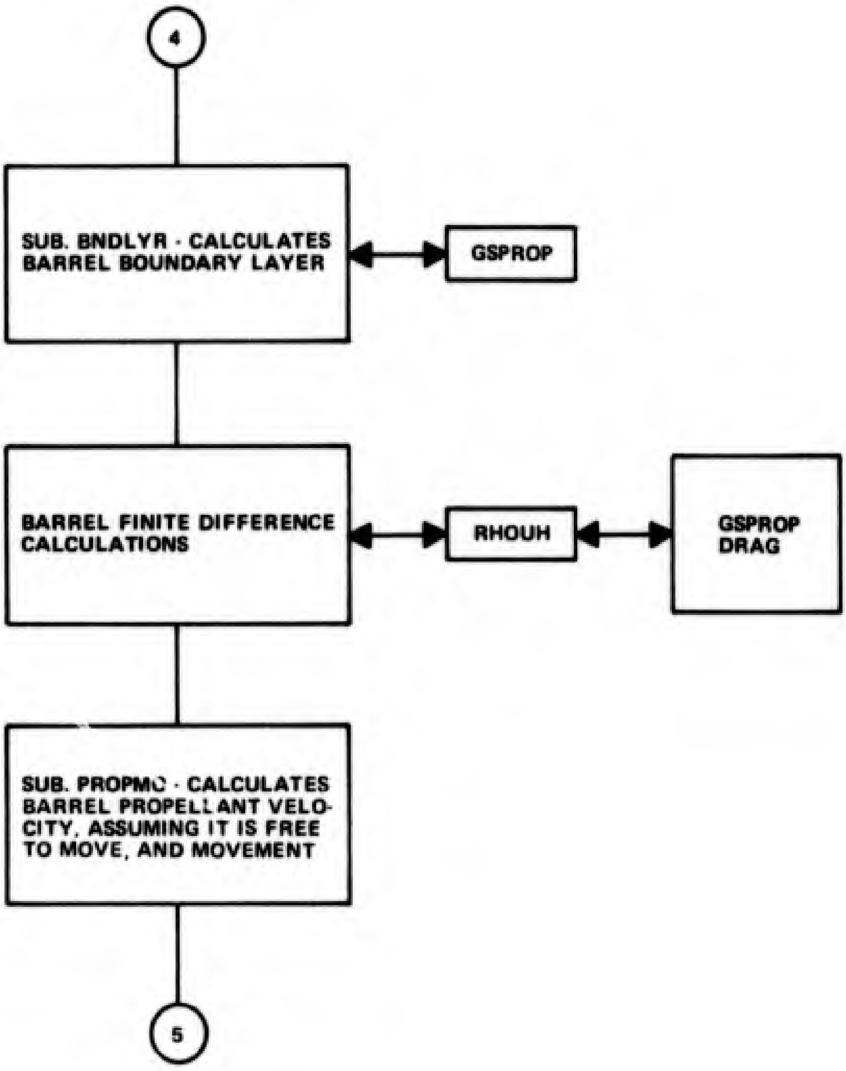

1
SUB. REGRES - CALCULATES BURN RATE, GAS GENERATED, AND CHANGE IN POROSITY OF CHAMBER PROPELLANT
SUB. MFLOW - CALCULATES FLOW INTERCHANGE BETWEEN IGNITER AND PROPELLANT BED.
2


2
IS
ALL PROP
IGNITED ?
YES
NO
SUB. PRPFIR - CALCULATES HEAT
TRANSFER AND PROPELLANT
SURFACE TEMPERATURE LEADING
TO IGNITION
SUB. REGRES - CALCULATES
BURN RATE, GAS GENERATED,
AND CHANGE IN POROSITY OF
CHAMBER PROPELLANT
3

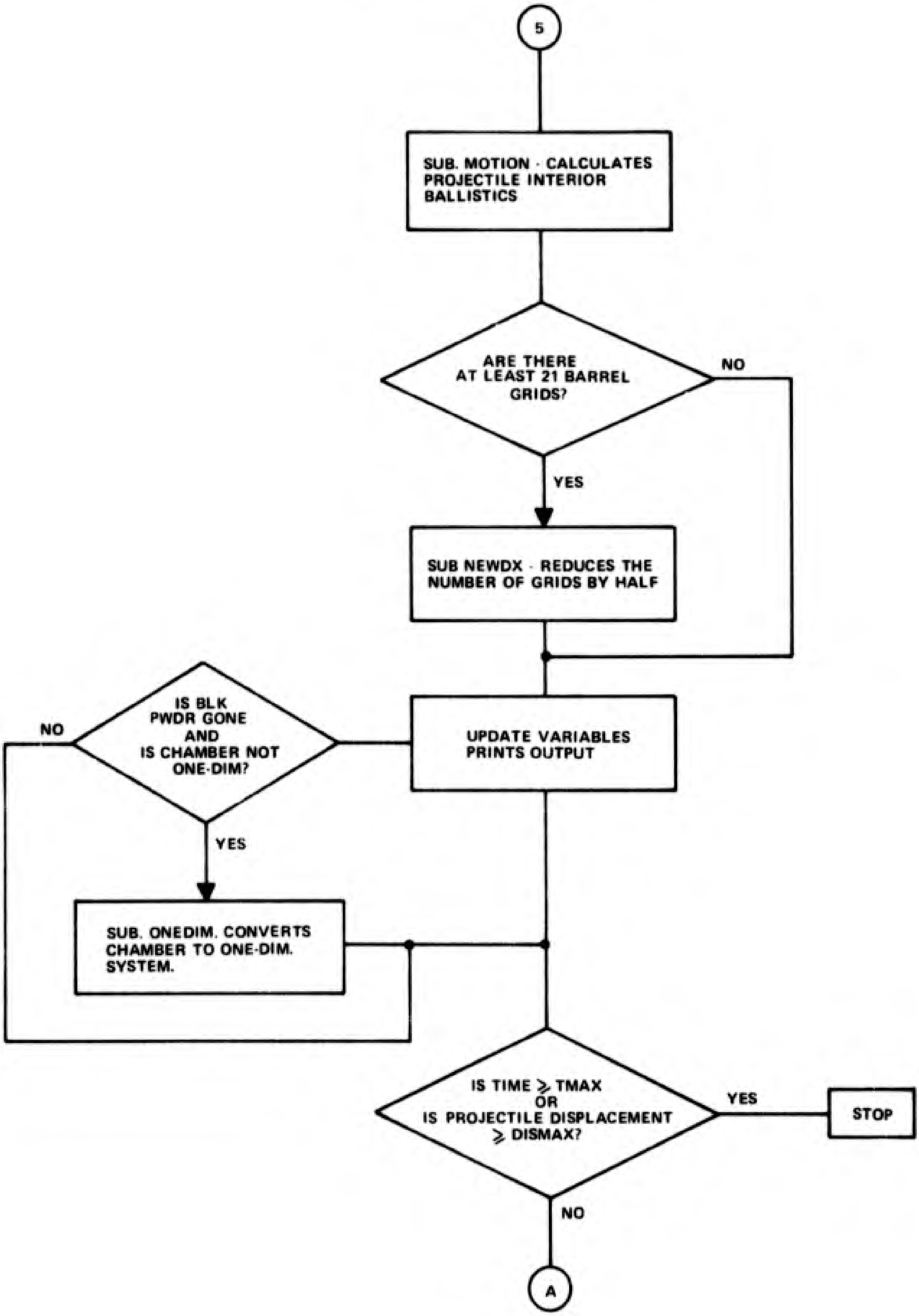
PROGRAM FLOW CHART (CONT.)


3
SUB. PRPVEL - CALCULATES CHAMBER PROPELLANT VELOCITY ASSUMING THE PROPELLANT IS FREE TO MOVE
SUB. PROPEL - COMPUTES ACTUAL CHAMBER PROPELLANT MOVEMENT
LOGIC LEADING TO SELECTION OF PROPER FINITE DIFFERENCE CALCULATION SUBROUTINE
AXIS
AXIT 2
AXIT 3
BSURA 2
ETC
GSPROP
DRAG
SUB. DIMIN - CALCULATES DEFLAGRATION OF BARREL PROPELLANT
4

PROGRAM FLOW CHART (CONT.)


4
SUB. BNDLYR - CALCULATES BARREL BOUNDARY LAYER
GSPROP
BARREL FINITE DIFFERENCE CALCULATIONS
RHOUH
GSPROP
DRAG
SUB. PROPMC - CALCULATES BARREL PROPELLANT VELOCITY, ASSUMING IT IS FREE TO MOVE, AND MOVEMENT
5

PROGRAM FLOW CHART (CONT.)


5
SUB. MOTION - CALCULATES PROJECTILE INTERIOR BALLISTICS
ARE THERE AT LEAST 21 BARREL GRIDS?
NO
YES
SUB NEWDX - REDUCES THE NUMBER OF GRIDS BY HALF
UPDATE VARIABLES PRINTS OUTPUT
IS BLK PWDR GONE AND IS CHAMBER NOT ONE-DIM?
NO
YES
SUB. ONEDIM. CONVERTS CHAMBER TO ONE-DIM. SYSTEM.
IS TIME ≥ TMAX OR IS PROJECTILE DISPLACEMENT ≥ DISMAX?
YES
STOP
NO
A

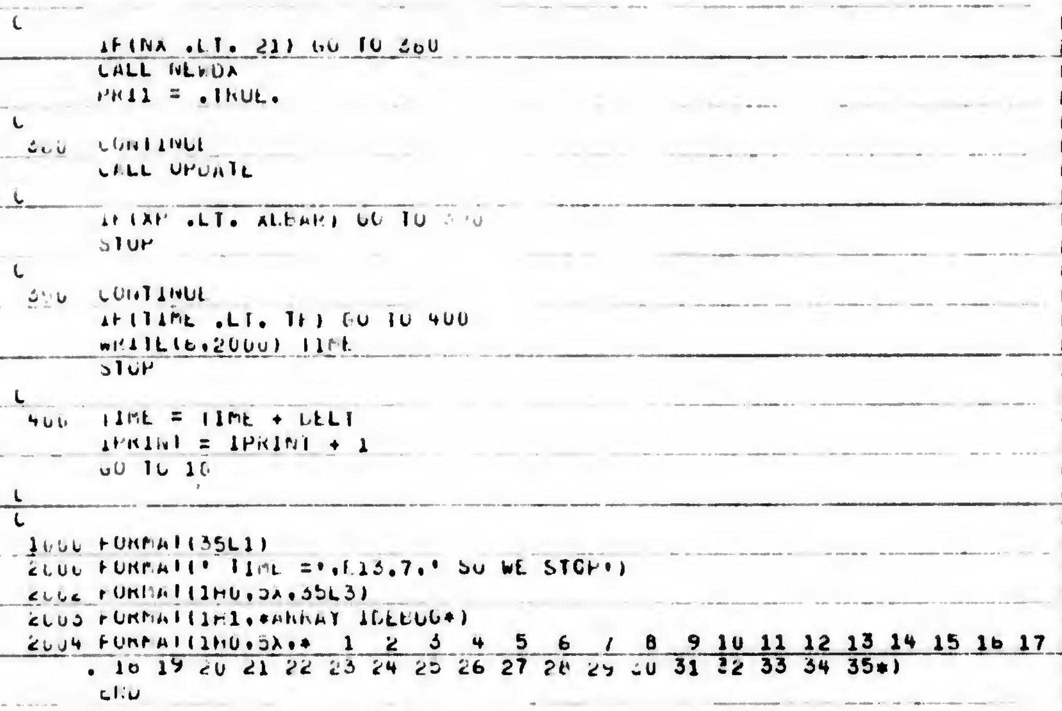
APPENDIX B
FORTRAN IV MACHINE LISTING

```
      PROGRAM PI07(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)                MOD0530   1
C     MAIN                                                               MAIN      3
      COMMON/OPT/IEP(2),DEP(2),CEP(2),FTOUM,PDALST                       MOD0925   1
      COMMON/PRMFLC/DCTPPM,DPRO,GPRM,PRPAP,FDSHR,GSFRKS,PRMCOF           MOD0925   2
      COMMON/FAILCC/THICKT,FHOOP,FCOMP,BTUE,XNTUB,FAIL,MFAIL(60),        MAIN      4
     1 THICK(60),PENET,TYPLE                                             MOD0925   3
      COMMON/BARRL2/BOREA,AP,VP,BORED,ECHEM,BOREDE,DT2BU,DTUSQ,XLBAR     MAIN      6
      COMMON/CALLP/BFLEFT                                                MAIN      7
      COMMON/CHAM/IX,IR,AC,RC,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),  CHAMIX    2
     2   AREACH,AREAC(60),IGNIT,OPEND,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX    3
     3  AREAR(60),AREAMX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,       CHAMIX    4
     4  AREARR,DIAPP1,RELEND,RELENG,IPS1,IPS2,RADFS,BPIGN                CHAMIX    5
      COMMON/CLOCK/TIME,DELT                                             MAIN      5
      COMMON/EQNS/C10X,TEER,T2DX,TWOTCP,QTCR,AMB,TWOGJ,DVAXIS,DVAXIT,    EQNS      2
     1  DX,DR,RX,GO,TWOGT,RRP                                            EQNS      3
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCONS,RHOF,PHI0,TF,CA,RHO0,      INPUTS    2
     1  H0,P0,U0,STRHOP,RK,BM,TIGNBP,GBCONS,TCTM                         INPUTS    3
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                        P         2
      COMMON/PROMO/FIRE                                                  MAIN     13
      COMMON/PRO/LF1,LF2,MF1,MF2,BGJAM,BGSTF,NVSP,IBGNP                  MOD0925   4
      COMMON/IGNITE/PORAD,EPCHG,BRBOUT,BGBRN,TMPBG,                      MOD0925   5
     1 BEGIC(5),XLIC(5)                                                  MOD0925   6
      COMMON/SPLINT/WHOLEC,WHOLEB                                        MAIN     14
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),      BARRL     2
     1  PG(100), TG(100),  EMCOT(100), GL(100), DRAG(100), FRICT(100),   BARRL     3
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100), BARRL     4
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),      BARRL     5
     4  UAENER(100)                                                      BARRL     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), TZC(60,5),                     BAG       3
     2   DOTMIG(60), DRAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBUT(60,5),  BAG       4
     3   PHIBID(60,5), RHOBID(60,5), HBGID(60,5), UBGID(60,5),           BAG       5
     4   VBGID(60,5),TBG(60,5),DOTMBG(60),DCTMP(60,5),PHIBP(60,5),       BAG       6
     5  PHIPID(60,5),TZR(60),TBP(60,5)                                   BAG       7
      LOGICAL PRI1,IDEBUG                                                PLOG      2
      LOGICAL IGNIT,OPEND,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG   2
      LOGICAL FIRE                                                       MAIN     19
      LOGICAL PENET                                                      MOD0604   1
      LOGICAL FAIL                                                       MAIN     20
      LOGICAL BFLEFT                                                     MAIN     21
      LOGICAL WHOLEC,WHOLEB                                              MAIN     22
      LOGICAL BRBOUT,BGBRN                                               MOD0925   7
      LOGICAL BGJAM,BGSTF                                                MOD0925   8
C                                                                        MAIN     23
C                                                                        MAIN     24
      READ(5,1000) IDEBUG                                                MAIN     25
      WRITE(6,2003)                                                      MAIN     26
      WRITE(6,2004)                                                      MAIN     27
      WRITE(6,2002) IDEBUG                                               MAIN     28
      IPRINT = 0                                                         MAIN     29
      NAMELIST/MODS/MODCH,MODGR                                          MAIN     30
      READ(5,MODS)                                                       MAIN     31
      IF(IDEBUG(1)) WRITE(6,MODS)                                        MAIN     32
C                                                                        MAIN     33
C    WHEN BLACK POWDER AT ALL GRIDS IS IGNITED, BPIGN WILL BE SET TRUE   MAIN     34
C    IN BPFIRE.                                                          MAIN     35
      BPIGN = .FALSE.                                                    MAIN     36
      BRNOUT = .FALSE.                                                  MOD0925    9
      BGJAM=.FALSE.                                                     MOD0925   10
      BGSTP=.FALSE.                                                     MOD0925   11
      BGBRN=.FALSE.                                                     MOD0925   12
      PENET=.FALSE.                                                     MOD0604    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(MFAIL(I) = 2) ALONG ITS ENTIRE LENGTH.        MAIN      44
C                                                                       MAIN      45
C                                                                       MAIN      46
C     ONED 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
      ONED = .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.  WHOLEB 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
      FIRE=.FALSE.                                                      MAIN      61
C                                                                       MAIN      62
C     FIRE 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
      IBGNP=IBEGB                                                       MOD0925   13
      IMPBG=10                                                          MOD0925   14
      NSVP=1                                                            MOD1018    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
      PRI1 = .FALSE.                                                    MAIN      77
      IF(MOD(IPRINT,MODCH) .EQ. 0) PRI1 = .TRUE.                        MAIN      78
C                                                                       MAIN      79
C********************************************************************** MAIN      80
C     SETTING DUMMY GRIDS                                               MAIN      81
C********************************************************************** MAIN      82
C     SET DUMMY CHAMBER GRIDS I = 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     GRIDS I = NGX.                                                    MAIN      85
```

```
C     UBG IS 0.0 AT CHAMBER GRID NGX+1                                  MAIN      86
C     AREAS AT BARREL GRID 1 WERE SET TO DGREA IN CHSET                 MAIN      87
C                                                                       MAIN      88
      NP1 = NGX+1                                                       MAIN      89
      IF(NX .GE. 2) GO TO 140                                           MAIN      90
      NM1 = NGX - 1                                                     MAIN      91
      DO 130 J=1,NGR                                                    MAIN      92
          RHOBG(NP1,J) = RHOBG(NM1,J)                                   MAIN      93
      UBG(NP1,J) = 2*VP- UBG(NM1,J)                                     MOD1205    1
          HBG(NP1,J) = HBG(NM1,J)                                       MAIN      95
          PCH(NP1,J) = PCH(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
          RHOBG(NP1,J) = RHOG(2)                                        MAIN     105
          UBG(NP1,J) = UG(2)                                            MAIN     106
          HBG(NP1,J) = HG(2)                                            MAIN     107
          PCH(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(ONED) GO TO 170                                                MAIN     116
      RHOG(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
          RHOG(1) = RHOG(1) + RHOBG(NGX,J)*AREAG(J)                     MAIN     125
          UG(1) = UG(1) + UBG(NGX,J)*AREAG(J)                           MAIN     126
          HG(1) = HG(1) + HBG(NGX,J)*AREAG(J)                           MAIN     127
          PG(1) = PG(1) + PCH(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
      RHOG(1) = RHOG(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 166                                                         MAIN     140
C                                                                       MAIN     141
  165 UP(1) = 0.0                                                       MAIN     142
```

```
      GO TO 180                                                         MAIN      143
C                                                                       MAIN      144
  170 RHOG(1) = RHOBG(NGX,1)                                            MAIN      145
      UG(1) = UBG(NGX,1)                                                MAIN      146
      HG(1) = HBG(NGX,1)                                                MAIN      147
      PG(1) = PCH(NGX,1)                                                MAIN      148
      PHI(1) = PHIBG(NGX,1)                                             MAIN      149
      UP(1) = UPB(NGX,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 BPFIR                                        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
      GAMBP=1.35                                                        MOD0925    16
      UPRM=156.1*SQRT((GAMBP-1.0)*HBP)                                  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
      FLGG=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(FLGG .EQ. 1.) GO TO 177                                        MOD0925    26
      DOTMP(I,J)=DOTMP(I,J) + DELT*DOTMPM                               MOD0925    27
      FLGG=1.                                                           MOD0925    28
      DOTMPM=DOTMP(I,J)                                                 MOD0925    29
  177 CONTINUE                                                          MOD0925    30
      IF(.NOT. ONEL) CALL MFLOW                                         MAIN      163
C                                                                       MAIN      164
C     FRPFIR 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 FRPFIR                                       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) + PHIBP(I,J) - 1.0                     MAIN      176
  190 CONTINUE                                                          MAIN      177
C     PATH SUBROUTINES                                                  MAIN      178
      XB = -DX                                                          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
  201     CALL AXIS                                                      MAIN
          GO TO 300                                                      MAIN
  202 IF(CHAM1) CALL AXIT1                                               MAIN
      IF(CHAM2) CALL AXIT2                                               MAIN
      IF(CHAM3 .AND. IR .EQ. 3) CALL AXIT3                               MAIN
      IF(CHAM3 .AND. IR .NE. 3) CALL AXIT2                               MAIN
          GO TO 300                                                      MAIN
  203     CALL INTER                                                     MAIN
          GO TO 300                                                      MAIN
  204     CALL BNDY                                                      MAIN
          GO TO 300                                                      MAIN
  205     CALL FSURFA                                                    MAIN
          GO TO 300                                                      MAIN
  206     CALL FSURFT                                                    MAIN
          GO TO 300                                                      MAIN
  207     CALL FSURFI                                                    MAIN
          GO TO 300                                                      MAIN
  208     CALL FSURFE                                                    MAIN
          GO TO 300                                                      MAIN
  209     IF(CHAM1) CALL BSURFA                                          MAIN
          IF(.NOT. CHAM1) CALL BSURA2                                    MAIN
          GO TO 300                                                      MAIN
  210 IF(CHAM1) CALL BSURFT                                              MAIN
      IF(CHAM2) CALL BSURT2                                              MAIN
      IF(CHAM3 .AND. IR .EQ. 3) CALL BSURT3                              MAIN
      IF(CHAM3 .AND. IR .NE. 3) CALL BSURT2                              MAIN
          GO TO 300                                                      MAIN
  211     CALL BSURFI                                                    MAIN
          GO TO 300                                                      MAIN
  212     CALL BSURFR                                                    MAIN
  300 CONTINUE                                                           MAIN
C                                                                        MAIN
C                                                                        MAIN
C     FIX ARRAY PHIGG SO THAT IT ONLY REPRESENTS POROSITY OF THE         MAIN
C     PROPELLANT AND NOT THE TOTAL POROSITY.                             MAIN
      DO 310 J=1,NGR                                                     MAIN
      DO 310 I=1,NGX                                                     MAIN
          PHIGG(I,J) = PHIGG(I,J) + 1.0 - PHIBP(I,J)                     MAIN
  310 CONTINUE                                                           MAIN
C                                                                        MAIN
C*********************************************************************** MAIN
C     BARREL SUBROUTINES                                                 MAIN
C*********************************************************************** MAIN
C                                                                        MAIN
      IF(NX .LT. 2) GO TO 350                                            MAIN
      CALL DIMIN                                                         MAIN
      IF(NX .GT. 2) CALL BNDLYR                                          MAIN
      CALL RHOOR                                                         MAIN
      CALL PROPMO                                                        MAIN
  350 CONTINUE                                                           MAIN
      CALL MOTION                                                        MAIN
C                                                                        MAIN
C                                                                        MAIN
C*********************************************************************** MAIN
C     UPDATE AND PRINT                                                   MAIN
C*********************************************************************** MAIN
```



```
C
      IF(NX .LT. 21) GO TO 300
      CALL NEWDX
      PRI1 = .TRUE.
C
  300 CONTINUE
      CALL UPDATE
C
      IF(XP .LT. XLBAR) GO TO 350
      STOP
C
  350 CONTINUE
      IF(TIME .LT. TF) GO TO 400
      WRITE(6,2000) TIME
      STOP
C
  400 TIME = TIME + DELT
      IPRINT = IPRINT + 1
      GO TO 10
C
C
 1000 FORMAT(35L1)
 2000 FORMAT(' TIME =',E13.7,' SO WE STOP')
 2002 FORMAT(1H0,5X,35L3)
 2003 FORMAT(1H1,*ARRAY IDEBUG*)
 2004 FORMAT(1H0,5X,*  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
     . 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35*)
      END
      SUBROUTINE CHSET                                                  CHSET        2
C                                                                       CHSET        3
      COMMON/IGNITE/PBFAP,BPCHG,BRKOUT,BGRAN,INPBG,                     MOD0925     35
     1 IBEGTC(5),XLTC(5)                                                MOD0925     36
      COMMON/EPT/TEE(2),BEP(2),CEP(2),FTOOR,PDALST                      MOD0925     37
      COMMON/PRTFLC/CCTTPP,CPRM,DPRM,PRMAXP,DSHR,CSPRKS,PRMCOF          MOD0925     38
      COMMON/FAILCL/THICKT,FHOOP,FCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       CHSET        4
     1 THICK(60),PENET,ITFCL                                            MOD0925     39
      COMMON/BARRL2/DCNFA,XF,VP,BORED,PCHCR,BORED8,DT2BD,DTDS0          CHSET        6
      COMMON/BURN/TFB,CT,PEXP                                           CHSET        7
      COMMON/CHAM/IX,IR,XE,RE,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX       2
     1   AREACH,AREAC(60),IGNIT,OPEN,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX       3
     2  AREAR(60),AREFAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),DAVG,       CHAMIX       4
     3  AREAM2,DIAPH1,BELLBL,FELBFG,IPS1,IPS2,RADFS,BPIGN               CHAMIX       5
      COMMON/CLOCK/TIME,DELT                                            CHSET        9
      COMMON/CAREAS/ARROW1,ARROW2,ARROW3,ARTOT                          MOD0301      1
      COMMON/DRGCON/VISC,FRCTF                                          MOD1215      1
      COMMON/TUBE/IENDTC(5),ITUBE,IBEGTC(5),INBP                        MOD1115      1
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     1  DX,DR,RX,GC,TWOGT,HBP                                           EQNS         3
      COMMON/FORCE/ PFORCE(60,5), PFORDT(60,5)                          CHSET       12
      COMMON/GASCON/RG,RRC,CVG,CVH                                      CHSET       13
      COMMON/GASES/RGMB,RRGMB,CVGMB,CVHMB,RGBP,RRCBP,CVOBP,CVHBP        CHSET       14
      COMMON/CFRACT/ FRACT1,FRACT2                                      CHSET       15
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN        2
     1 XLTOT(60,5), DOTOT(60,5), DITOT(60,5), XLO,  DOO, DIO,           GRAIN        3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN        4
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA        2
     1  AH(60),FRACT(60)                                                HOLEA        3
      COMMON/INPUTS/C1,C2,C3,C4,TC,TIGN,GCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS       2
     1  HO,PO,UO,GTRHOF,HW,EM,TIGCBP,GRCONS,TOTM                        INPUTS       3
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                       P            2
      COMMON/PRIMV/BPLEES,BFRAD(60,5),AGENBP,BGENBP,EXPBP               PRIMV        2
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),        BAG          2
     1   VBG(60,5), CPB(60,5), PCH(60,5), TZC(60,5),                    BAG          3
     2   DOTMIG(60), DRAG(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),CCIMP(60,5),PHIBP(60,5),      BAG          6
     5  PHIFTD(60,5),TZP(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),  FMDOT(100), GL(100), DDRAG(100), FRICT(100), BARRL        3
     2  QCONV(100), DUP(100), DPHI(100), DRHOG(100), DHG(100), DUG(100), BARRL       4
     3  AMASS(100), AMOM(100), AENER(100), DAMASS(100), DAMOM(100),     BARRL        5
     4  DAENER(100)                                                     BARRL        6
      LOGICAL IGNIT,OPEN,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG      2
      LOGICAL PRI1,IDEBUG                                               PLOG         2
      LOGICAL FAIL                                                      CHSET       25
      LOGICAL PENET                                                     MOD0605      3
C                                                                       CHSET       26
      DATA GRAV,XJUL,PI,PIDF/32.16,778.,3.141593,.785398/               CHSET       27
      NAMELIST/CHIFP/                                                   CHSET       28
     1  XLO,DOO,DIO,NFERF,FRCTF,INBP,                                   MOD1215      2
     2  RGMB,RRGMB,CVGMB,CVHMB,RHOP,CHWT,                               CHSET       30
     3 HMB, TIGN, TO, PO, UO, VO, TW, AGEN, BGEN, CGEN, PEXP,           CHSET       31
     4  ALPHA,BETA, C1,C2,C3,C4, CA, HMAX, XK, TF,                      CHSET       32
     5  NGR,NGX,DR,DRAM,XLBEL,GAP,NHOLES,XCL,NROWF,                     CHSET       33
     6  RADHOL,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,                         CHSET       34
     / DIAMBT,BOREL,CHAM1,CHAM2,CHAM3,PFIG,TYFCE,                       MOD0925     40
     & RUBP,RRUBP,BPRADD,BPDENS,AGENBP,BGENBP,EXPBP,PRMCOF,             MOD0925     41
     STOPGAP,HBP,CVUBP,CVHBP,TIGNBP,XPBP,ALPHBP,                        CHSET       37
     * THICKT,PHOOP,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
      MFAIL(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(IDEBUG(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
      FN = FLOAT(NPERF)                                                 CHSET       61
      THICKT = THICKT/12.0                                              CHSET       62
      DO 1 I=1,60                                                       CHSET       63
    1 THICK(I)=THICKT                                                   CHSET       64
      PHOOP = PHOOP*144.0                                               CHSET       65
      PCOMP = PCOMP*144.0                                               CHSET       66
      BTUB=BTUB/(12.0*144.0**XNTUB)                                     CHSET       67
      XL0 = XL0/12.                                                     CHSET       68
      D00 = D00/12.                                                     CHSET       69
      D10 = D10/12.                                                     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
      DIAMBT = DIAMBT/12.                                               CHSET       77
      DRAM = DRAM/12.                                                   CHSET       78
      GAP = GAP/12.                                                     CHSET       79
      XLBEL = XLBEL/12.                                                 CHSET       80
      TOPGAP = TOPGAP/12.                                               CHSET       81
      P0 = P0*144.                                                      CHSET       82
      AGEN = AGEN/(12.*144.**PEXP)                                      CHSET       83
      BGEN = BGEN/(12.*144.**PEXP)                                      CHSET       84
      CGEN = CGEN/12.                                                   CHSET       85
      BPRADD = BPRADD/12.                                               CHSET       86
      BPDENS = BPDENS/454.*16.38*1728.                                  CHSET       87
      AGENBP = AGENBP/(12.0*144.0**EXPBP)                               CHSET       89
      BGENBP = BGENBP/12.0                                              CHSET       90
C                                                                       CHSET       91
C     INITIALLY THE GAS CONSTANTS ARE THOSE 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 = ROBP                                                        CHSET      95
       RRO = RROBP                                                      CHSET      96
       CVO = CVOBP                                                      CHSET      97
       CVH = CVHBP                                                      CHSET      98
C                                                                       CHSET      99
C     BORED WAS TAKEN OUT OF BARINP BECAUSE IT IS NEEDED HERE FOR AREA  CHSET     100
C     CALCULATIONS.                                                     CHSET     101
       BORED = BORED/12.                                                CHSET     102
       BORER = 0.5*BORED                                                CHSET     103
       BOREA = PI*BORER*BORER                                           CHSET     104
C                                                                       CHSET     105
C     DETERMINE DX, IBEGB, IBEGC, IENDB                                 CHSET     106
       DX = DRAM/FLOAT(NGX-1)                                           CHSET     107
       IF(XLBEL + GAP .GT. DRAM) GAP = DRAM - XLBEL                     CHSET     108
C                                                                       CHSET     109
C     IF THE BELL TUBE BEGINS IN GRID 1, PUSH IT INTO GRID 2.           CHSET     110
       BELBEG = DRAM - (XLBEL + GAP)                                    CHSET     111
       IBEGC=BELBEG/DX+1.5                                              CHSET     112
       IBEGB=IBEGC                                                      CHSET     113
       BELEND = DRAM - GAP                                              CHSET     114
       IENDB = NGX - IFIX(GAP/DX + 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 BELL TUBE OCCUPIES.                         CHSET     118
       IF(IBEGB.EQ.1)FRACT1=1.0-2.0*BELBEG/DX                           CHSET     119
       IF(IBEGB.GT.1)FRACT1=((FLOAT(IBEGB)-0.5)*DX-BELBEG)/DX           CHSET     120
       FRACT2 = (BELEND - (FLOAT(IENDB) - 1.5)*DX)/DX                   CHSET     121
C                                                                       CHSET     122
C     DM, GCONS, AND GBCONS ARE NEEDED IN SUBROUTINE PRPFIR.            CHSET     123
       DM = (1.5*DOD*DCD*XLD)**0.333                                    CHSET     124
       GCONS = 2.0*SQRT(ALPHA/3.141593)/XK                              CHSET     125
       GBCONS = 2.0*SQRT(ALPHBP/3.141593)/XKBP                          CHSET     126
       PBRAD=DIAMBT/2.                                                  MOD0925    43
C                                                                       CHSET     127
C     ATPB AND CT ARE NEEDED FOR CALCULATING THE BURN RATE IN REGRES    CHSET     128
       ATPB = AGEN*T0 + BGEN                                            CHSET     129
       CT = CGEN*T0                                                     CHSET     130
C                                                                       CHSET     131
C      VISG IS A CONSTANT USED IN SUBROUTINE DRAG                       CHSET     132
       VISG = DM*GRAV                                                   CHSET     133
C                                                                       CHSET     134
C     CALCULATE HW FROM TW AND P0                                       CHSET     135
       CALL GSPROP(RO,RRO,R,CVO,CVH,CV,P0,HW,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
       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
      GJ = GRAV*XJOL                                                       CHSET     152
      TWOGJ = 2.0*GJ                                                       CHSET     153
      DELT = BETA*DX/SQRT(TWOGJ*HMAX)                                      CHSET     154
      TWODT = 2.0*DELT                                                     CHSET     155
      DTDX = DELT/DX                                                       CHSET     156
      DTDR = DELT/DR                                                       CHSET     157
      T2DR = 0.5*DTDR                                                      CHSET     158
      T2DX = 0.5*DTDX                                                      CHSET     159
      TWOTDR = 2.0*DTDR                                                    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
      CALL AREAS                                                           CHSET     169
C                                                                          CHSET     170
C                                                                          CHSET     171
C                                                                          MOD0301     2
C     USING THE AREAS JUST CALCULATED, CALCCLATE THE INITIAL VOLUME        MOD0301     3
C     OF CHAMBER RADIAL ROWS 1, 2 AND 3.  ALSO CALCULATE THE TOTAL         MOD0301     4
C     CHAMBER VOLUME.                                                      MOD0301     5
      ARROW1 = (FLOAT(NGX) - 0.5)*DX*AREAAX                                MOD0301     6
C                                                                          MOD0301     7
      ARROW2 = AREAR(1)*DX*0.5                                             MOD0301     8
      DO 100 I = 2,NGX                                                     MOD0301     9
      ARROW2 = ARROW2 + AREAR(I)*DX                                        MOD0301    10
  100 CONTINUE                                                             MOD0301    11
C                                                                          MOD0301    12
      ARTOT = ARROW1 + ARROW2                                              MOD0301    13
      IF(CHAM2) GO TO 120                                                  MOD0301    14
C                                                                          MOD0301    15
      ARROW3 = AREAGP(1)*DX*0.5                                            MOD0301    16
      DO 110 I = 2,NGX                                                     MOD0301    17
      ARROW3 = ARROW3 + AREAGP(I)*DX                                       MOD0301    18
  110 CONTINUE                                                             MOD0301    19
      ARTOT = ARTOT + ARROW3                                               MOD0301    20
C                                                                          MOD0301    21
  120 CONTINUE                                                             MOD0301    22
C************************************************************************* CHSET     172
C     INITIALIZE CHAMBER MATRIX                                            CHSET     173
C************************************************************************* CHSET     174
C                                                                          CHSET     175
      CALL CLEAR(XL(1,1),GI(60,5))                                         CHSET     176
      CALL CLEAR(XLTOT(1,1),GITOT(60,5))                                   CHSET     177
      CALL CLEAR(PHIGG(1,1),TBP(60,5))                                     CHSET     178
      CALL CLEAR(BPRAD(1,1),BPRAD(60,5))                                   CHSET     179
      CALL CLEAR(PFORCE(1,1),PFORET(60,5)),                                CHSET     180
C                                                                          CHSET     181
C     UPB 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,NCR
      DO 510 I=1,NGX
          PHIBG(I,J) = 1.0
          PHIGTC(I,J) = 1.0
          PHIBP(I,J) = 1.0
          PHIPTB(I,J) = 1.0
  510 CONTINUE
C
C   CALCULATE INITIAL POROSITIES OF GRIDS CONTAINING BLACK POWDER
C
      CALL BPINIT
C
C
      DO 1001 I=1,NGX
      DO 1000 J=1,2
      BPRAD(I,J)=BPRAD0
 1000 TBP(I,J)=T0
 1001 CONTINUE
C    NUMBER OF GRIDS OF TUBE
      NGT=IENDB-IBEGC
      NGT3 = NGT/3
      LIM1 = IBEGC + NGT3
      IBEG2 = LIM1 + 1
      LIM2 = LIM1 + NGT3
      IBEG3 = LIM2 + 1
C
C    SINCE THE INTEGER DIVISION BY 3 MIGHT INVOLVE TRUNCATION THE LAST
C    THIRD OF THE TUBE MAY HAVE MORE THAN NGT3 GRIDS.
C
      DO 530 I=IBEGB,LIM1
          BPRAD(I,1) = BPRAD0
          TBP(I,1) = T0
  530 CONTINUE
      DO 540 I=IBEG2,LIM2
          BPRAD(I,1) = BPRAD0
          TBP(I,1) = T0
  540 CONTINUE
      DO 550 I=IBEG3,IENDB
          BPRAD(I,1) = BPRAD0
          TBP(I,1) = T0
  550 CONTINUE
C
C
C    PROPELLANT IS PACKED IN GRIDS ABOVE THE BELL TUBE ONLY.  GRAIN
C    DIMENSIONS AND GRAIN SURFACE TEMPERATURE WILL BE LEFT 0.0 AT OTHER
C    GRIDS.
      IF(CHAMB) NN = 2
      IF(.NOT. CHAMB) NN = NGR
      DO 570 I=IBEGB,IENDB
      TZR(I) = T0
      DO 565 J=2,NN
          XL(I,J) = XL0
          DO(I,J) = DO0
          DI(I,J) = DI0
          TZC(I,J) = T0
          XLTOT(I,J)=XL0
          DOTOT(I,J)=DO0
```

```
            UITOT(I,J)=UIG
  560   CONTINUE                                                          CHSET     248
  570   CONTINUE                                                          CHSET     249
C                                                                         CHSET     250
C                                                                         CHSET     251
C       CALCULATE  VOLUME  ABOVE  THE  BELL  TUBE  AND  PORCSITY  OF  GRIDS  CHSET  252
C       WITH  PROPELLANT.                                                 CHSET     253
        IBP1 = IBEGB + 1                                                  CHSET     254
        IBM1 = IENDB - 1                                                  CHSET     255
        VOLCHG=AREAR(IBEGB)*FRACT1*DX/2.                                  CHSET     256
        IF(IBEGB .GT. 1) VOLCHG = VOLCHG*2.0                              CHSET     257
        DO 580 I=IBP1,IBM1                                                CHSET     258
            VOLCHG = VOLCHG + AREAR(I)*DX                                 CHSET     259
  580   CONTINUE                                                          CHSET     260
C                                                                         CHSET     261
        VOLCHG = VOLCHG + AREAR(IENDB)*FRACT2*DX                          CHSET     262
C                                                                         CHSET     263
        RHOB = CHWT/VOLCHG                                                CHSET     264
        PHIO=1.0-RHOB/RHOP                                                CHSET     265
C                                                                         MOD1111     1
        DO 585 J=2,NR                                                     CHSET     267
        PHIBG(IBEGB,J)=1.0-FRACT1 + FRACT1*(1.0-RHOE/RHOP)                CHSET     268
  585   CONTINUE                                                          MOD0925    44
C                                                                         CHSET     270
        DO 590 J=2,NR                                                     CHSET     271
        DO 590 I=IBP1,IBM1                                                CHSET     272
        PHIBG(I,J)=1.0-RHOB/RHOP                                          CHSET     273
  590   CONTINUE                                                          MOD0925    45
C                                                                         CHSET     275
        DO 595 J=2,NR                                                     CHSET     276
        PHIBG(IENDB,J)=1.0-FRACT2 + FRACT2*(1.0-RHOE/RHOP)                CHSET     277
  595   CONTINUE                                                          MOD0925    46
C     DETERMINE HO AND RHOO FROM PO AND TO AND GSPROP                     CHSET     279
        CALL GSPROP(HO,RHO,R,CVO,CVH,CV,PO,HO,TO,RHOO,0.0 ,0.0 ,GAM,CP,1) CHSET     280
C                                                                         CHSET     281
        DO 600 J=1,NGR                                                    CHSET     282
        DO 600 I=1,NGX                                                    CHSET     283
            PCH(I,J) = PO                                                 CHSET     284
            HBG(I,J) = HO                                                 CHSET     285
            RHOBG(I,J) = RHOO                                             CHSET     286
            UBG(I,J) = UO                                                 CHSET     287
            VBG(I,J) = VO                                                 CHSET     288
            TBG(I,J) = TO                                                 CHSET     289
  600   CONTINUE                                                          CHSET     290
C                                                                         CHSET     291
        IF(IDEBUG(3)) WRITE(6,2006)                                       CHSET     292
        DO 630 J=1,NGR                                                    CHSET     293
            IF(IDEBUG(3)) WRITE(6,2007) J,(PHIBG(I,J),I=1,NGX)            CHSET     294
  630   CONTINUE                                                          CHSET     295
C                                                                         CHSET     296
        IF(IDEBUG(3)) WRITE(6,2008)                                       CHSET     297
        DO 640 J = 1,NGR                                                  CHSET     298
            IF(IDEBUG(3)) WRITE(6,2007) J,(PHIBP(I,J),I=1,NGX)            CHSET     299
  640   CONTINUE                                                          CHSET     300
C                                                                         CHSET     301
C                                                                         CHSET     302
C*************************************************************************CHSET     303
                                                                          CHSET     304
```

```
C     SET UP HOLES                                                      CHSET     305
C*********************************************************************** CHSET     306
C                                                                       CHSET     307
C     SET UP HOLES IN CELL TUBE AND ANY PSEUDO HOLES BETWEEN RADIAL ROWS CHSET     308
C     ONE AND TWO.                                                      CHSET     309
       CALL HOLSET                                                      CHSET     310
C                                                                       CHSET     311
C     GET HOLE AREA AT EACH GRID BETWEEN RADIAL ROWS ONE AND TWO.       CHSET     312
       CALL HOLES                                                       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 SUBROUTINE AREAS.                                   CHSET     317
       IF(.NOT. CHAM3) GO TO 700                                        CHSET     318
       AREAH2 = 0.05*(PI/BETA)*0.5*(DAVG - TOPGAP)*TOPGAP*              CHSET     319
      1     SQRT(2.0/(GAM - 1.0))                                       CHSET     320
C                                                                       CHSET     321
C                                                                       CHSET     322
  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
       IF(CHAM1) WRITE(6,2000)                                          CHSET     330
       IF(CHAM1) GO TO 800                                              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 THEM  CHSET     335
C     AFTER SUMMING.                                                    CHSET     336
C     FIRST SUM ALONG THE AXIS.  THERE ALL AREAS ARE AREAAX SO FACTOR IT. CHSET     337
       SUM = 0.5*(PHIBG(1,1) + PHIBP(1,1) - 1.0)                        CHSET     338
       DO 720 I=2,NGX                                                   CHSET     339
          SUM = SUM + PHIBG(I,1) + PHIBP(I,1) - 1.0                     CHSET     340
  720  CONTINUE                                                         CHSET     341
       TOTM = SUM*AREAAX                                                CHSET     342
C                                                                       CHSET     343
       SUM = 0.5*(PHIBG(1,2) + PHIBP(1,2) - 1.0)*AREAR(1)               CHSET     344
       DO 730 I=2,NGX                                                   CHSET     345
          SUM = SUM + (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)          CHSET     346
  730  CONTINUE                                                         CHSET     347
       TOTM = TOTM + SUM                                                CHSET     348
C                                                                       CHSET     349
       IF(CHAM2) GO TO 750                                              CHSET     350
C****PHIBG(I,3) AND PHIBP(I,3) ARE 1.0 FOR ALL I                        CHSET     351
       SUM = 0.5*AREAGP(1)                                              CHSET     352
       DO 740 I=2,NGX                                                   CHSET     353
          SUM = SUM + AREAGP(I)                                         CHSET     354
  740  CONTINUE                                                         CHSET     355
       TOTM = TOTM + SUM                                                CHSET     356
C                                                                       CHSET     357
  750  CONTINUE                                                         CHSET     358
       TOTM = TOTM*RHOG*DX                                              CHSET     359
C                                                                       CHSET     360
C     ADD IN MASS OF BLACK POWDER AND PROPELLANT.                       CHSET     361
```

```
      TOTM = TOTM + BPCHG + CHWT                                          CHSET      362
C                                                                         CHSET      363
 800  CONTINUE                                                            CHSET      364
C                                                                         CHSET      365
      NAMELIST/CHRIN/XLO,DOO,DIO,RADHOL,PO,EGEN,XCL,NROWH,                CHSET      366
     $  NHOLES,AREAH,TIPL,DELT,TWOGJ,DTCX,DTDR,T2CR,T2DX,TWOTDR,PHIO,     CHSET      367
     $ RO,RHOO,TO,DVAXIS,DVAXIT,GCONS,ATPB,CT,VTSG,HW,DIAM1,DIAM2,        CHSET      368
     $  DIS1,DIS2,DIS3,DIS4,DRED,IGNIT,AGEN,CGEN,                         CHSET      369
     $  DIAMBT,CHAM2,BORED,BORER,BOREA,AREAAX,AREACH,                     CHSET      370
     $  BPRADO,BPDENS,AGENBF,RO,RRO,CHWT ,BPCHG,                          MOD1215      3
     $  DRAM,GAP,XLBEL,DX,IBEGC,IBEGB,IENDB,                              CHSET      372
     $  BELBEG,ADD,BELEND,FRACT1,FRACT2,VOLCHG,RHCB,                      CHSET      373
     $  RHOBP1,RHOBP2,RHOBP3,NGT,NGT3,LIM1,IBEG2,LIM2,IBEG3,              CHSET      374
     $  TOPGAP,DAVG,GAM,AREAH2,TOTM,DM,GBLCKS,AH,FRACT,IPS1,IPS2          CHSET      375
      IF(IDEBUG(4)) WRITE(6,CHRIN)                                        CHSET      376
C                                                                         CHSET      377
C                                                                         CHSET      378
C                                                                         CHSET      379
      RETURN                                                              CHSET      380
C                                                                         CHSET      381
 2000 FORMAT(//,' LOGIC FOR FINDING TOTM WHEN CHAM1 IS TRUE IS NOT WRITT  CHSET      382
     $EN YET')                                                            CHSET      383
 2006 FORMAT(///,' ARRAY PHIBG',/)                                        CHSET      384
 2007 FORMAT(/,' RADIAL ROW',I2,/,(20X,10F10.6))                          CHSET      385
 2008 FORMAT(///,' ARRAY PHIBP',/)                                        CHSET      386
      END                                                                 CHSET      387
      SUBROUTINE PATHS
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     2  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,
     3  AREAP2,DIAMB1,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
C
C     SUBROUTINE PATHS INITIALIZES ARRAY IPATH.
C     VALUES OF IPATH CORRESPOND TO THE PATH SUBROUTINES IN THE
C     FOLLOWING WAY--
C          1  -  AXIS      2  -  AXIT      3  -  INTER    4  -  BNDY
C          5  - FSURFA     6  - FSURFT     7  - FSURFI    8  - FSURFB
C          9  - BSURFA     10 - BSURFT     11 - BSURFI    12 - BSURFB
C     IF CHAM2 IS TRUE THE CHAMBER CONSISTS OF TWO ONE-DIMENSIONAL ROWS,
C     ONE ROW USING AXIS ROUTINES, THE OTHER USING AXIT ROUTINES.
C     IF CHAM3 IS TRUE THE CHAMBER CONSISTS OF THREE ONE-DIMENSIONAL
C     ROWS, TWO ROWS LIKE THOSE WHEN CHAM2 IS TRUE AND  THE THIRD USING
C     SIMILAR ROUTINES.  VALUES OF IPATH CORRESPONE TO THE NEW ROUTINES
C     AS FOLLOWS--
C         2 - AXIT2,AXIT3
C         9 - BSURA2
C         10 - BSURT2,BSURT3
C
      IF(CHAM2 .OR. CHAM3) GO TO 80
C
C
C     WHEN I=1, BSURFA, FSURFI, AND BSURFB ARE CALLED.
      IPATH(1,1) = 9
      J = 2
   10 IPATH(1,J) = 11
      J = J+1
      IF(J.LT.NGR) GO TO 10
      IPATH(1,J) = 12
C
C     FROM I=2 UNTIL THE BELL IGNITER TUBE, AXIS, INTER, AND BNDY
C     ARE CALLED.
      I=2
   20 IF(I.GT.IBEGB) GO TO 40
      IPATH(I,1) = 1
      J=2
   30 IPATH(I,J) = 3
      J=J+1
      IF(J.LT.NGR) GO TO 30
      IPATH(I,J) = 4
      I=I+1
      GO TO 20
C
C     FOR VALUES OF I WHERE THE BELL IGNITER TUBE IS AXIS, AXIT, INTER,
C     AND BNDY ARE CALLED.
C     FOR THE GRID ON THE SURFACE OF THE BELL IGNITER TUBE AT EITHER
C     END, INTER IS CALLED.
   40 IF(I.GE.IENDB) GO TO 60
      IPATH(I,1) = 1
      IPATH(I,2) = 2
      J=3
```

```
 50   IPATH(I,J) = 3
      J=J+1
      IF(J.LT.NGR) GO TO 50
      IPATH(I,J) = 4
      I=I+1
      GO TO 40
C
C     AFTER THE BELL IGNITER TUBE IS PASSED, AXIS, INT R, AND BNDY ARE
C     CALLED.
 60   IF(I.GT.NGX) GO TO 90
      IPATH(I,1) = 1
      J=2
 70   IPATH(I,J) = 3
      J=J+1
      IF(J.LT.NGR) GO TO 70
      IPATH(I,J) = 4
      I=I+1
      GO TO 60
C
C
C     THE FOLLOWING LOGIC IS USED FOR BOTH CHAM2 AND CHAM3 TRUE
 80   CONTINUE
      IPATH(1,1) = 9
      IPATH(1,2) = 10
      IPATH(1,3) = 10
      DO 85 I=2,NGX
          IPATH(I,1) = 1
          IPATH(I,2) = 2
          IPATH(I,3) = 2
 85   CONTINUE
C
C
 90   CONTINUE
      IF(.NOT. IDEBUG(5)) RETURN
      WRITE(6,2000)
      DO 95 J=1,NGR
          WRITE(6,2001) J,(IPATH(I,J),I=1,NGX)
 95   CONTINUE
      RETURN
C
 2000 FORMAT(///,' ARRAY IPATH',/)
 2001 FORMAT(' RADIAL ROW',I2,/,10X,30(I4),/,10X,20(I4))
      END
      SUBROUTINE HOLSET                                                 HOLSET     2
      COMMON/CHAM/IX,IR,AD,RD,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GG,TWODT,HBP                                           EQNS       3
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA      2
     $  AH(60),FRACT(60)                                                HOLEA      3
      DIMENSION DTEMP(50),NTEMP(50),RTEMP(50)                           HOLSET     6
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
C                                                                       HOLSET     8
C     THE DISTANCE (IN INCHES) BETWEEN THE CENTER LINE OF HOLES IN ROW I HOLSET    9
C     AND THE CENTER LINE OF THE PREVIOUS RO. OF HOLES WAS INPUT INTO   HOLSET    10
C     XCL(I).  XCL(1) CONTAINS THE DISTANCE OF THE CENTER LINE OF HOLES HOLSET    11
C     IN ROW 1 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(NROWH .EQ. 0) GO TO 102                                        HOLSET    21
      DO 100 I=1,NROWH                                                  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     RADPS, 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
      RADPS=0.1*SQRT(DIAMBT*DX)                                         MOD0925   48
      IPS1 = IBEGB                                                      HOLSET    33
C                                                                       HOLSET    34
      DO 105 I=1,IPS1                                                   HOLSET    37
      XCL(I)=FLOAT(I-1)*DX+DX/4.                                        MOD0925   49
          NHOLES(I) = 1                                                 HOLSET    39
          RADHOL(I) = RADPS                                             HOLSET    40
  105 CONTINUE                                                          HOLSET    41
      IF(BELBEG.LT.DX)RADHOL(1)=RADHOL(1)*SQRT(BELBEG/DX)               HOLSET    42
C                                                                       HOLSET    43
C     SET UP HOLES ON THE BELL TUBE.                                    HOLSET    44
      IF(NROWH .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(NROWH .EQ. 1) GO TO 115                                        HOLSET    50
      DO 110 I=2,NROWH                                                  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(XCL(I1) + RADHOL(I1) .LT. BELEND) GO TO 110
            ILAST = I - 1
            WRITE(6,2010) NROWH,ILAST
            I1 = I1 - 1
            GO TO 115
  110   CONTINUE
  112   IF(NROWH .EQ. 0) I1 = IPS1
C
C     THE LAST VALUE OF I1 IS THE NUMBER OF HOLES SO FAR.
C     SET UP PSEUDO HOLES AFTER THE BELL TUBE.
  115   NROWH=I1
        IPS2 = IENDB
C
C     IF THE BELL TUBE EXTENDS BEYOND GRID POINT IENDB, THERE SHOULD NOT
C     BE A PSEUDOHOLE AT GRID POINT IENDB.
        IF(BELEND .GT. FLOAT(IENDB-1)*DX) IPS2 = IENDB + 1
        DO 120 I=IPS2,NGX
            NROWH=NROWH+1
            XCL(NROWH)=FLOAT(I-1)*DX
            NHOLES(NROWH)=1
            RADHOL(NROWH) = RADPS
            IF(NROWH.EQ. 85)GO TO 140
  120   CONTINUE
C
C     CLEAR HOLE ARRAY ENTRIES WHERE THERE ARE NO HOLES
        NR1 = NROWH + 1
        CALL CLEAR(XCL(NR1),XCL(85))
        CALL CLEAR(RADHOL(NR1),RADHOL(85))
        DO 130 I=NR1,85
            NHOLES(I) = 0
  130   CONTINUE
C
  140   CONTINUE
C
        RETURN
 2010 FORMAT(//,I5,' WAS INPUT AS THE NUMBER OF ROWS OF HOLES ON THE BEL
     $L TUBE, BUT ONLY ',I4,'  FIT ON THE TUBE')
        END
```

```
      SUBROUTINE HOLES                                                  HOLES      2
C*                                                                      HOLES      3
      COMMON/CHAM/IX,IY,XL,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     1   AREACT,AREAC(60),IGNIT,OPEN,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     1  AREAM(60),AREAMX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     1  AREAM2,DIAMBT,DELEND,DELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/EQNS/ETOX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     1  DX,DR,EA,QC,TWODT,HBP                                           EQNS       3
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA      2
     1   AH(60),FRACT(60)                                               HOLEA      3
      DIMENSION TA(85)                                                  HOLES      7
      LOGICAL IGNIT,OPEN,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
      DATA PI/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*       NROWH - NUMBER OF ROWS OF HOLES                                HOLES     18
C*       XCL(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(85))                                          HOLES     26
      CALL CLEAR(AREAH(1),AREAH(60))                                    HOLES     27
C                                                                       HOLES     28
C*    I1 INDEXES THE GRIDS                                              HOLES     29
      X = -DX                                                           HOLES     30
      DO 40 I1=1,NGX                                                    HOLES     31
C*                                                                      HOLES     32
C*    X IS THE COORDINATE OF THE CENTER OF GRID I1                      HOLES     33
      X = X + DX                                                        HOLES     34
      GRDEND = 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 INDEXES THE ROWS OF HOLES                                      HOLES     40
      DO 30 I2 = 1,NROWH                                                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(GRDEND .LE. XCL(I2) - RADHOL(I2)) GO TO 40                     HOLES     45
      FNH = 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(GRDEND .LT. XCL(I2) + RADHOL(I2)) GO TO 10                     HOLES     52
      AREAXP = FNH*PI*RADHOL(I2)*RADHOL(I2)                             HOLES     53
      GO TO 20                                                          HOLES     54
C                                                                       HOLES     55
C                                                                       HOLES     56
C*                                                                      HOLES     57
 10    TERM = GRDEND - XCL(I2)                                          HOLES     58
C*    FIND HALF OF THE CENTRAL ANGLE TO THE CHORD AT THE GRID END       HOLES     59
       THETA = ACOS(ABS(TERM)/RADHOL(I2))                               HOLES     60
C*                                                                      HOLES     61
C*    FIND AREA OF CIRCULAR SECTOR WHOSE CENTRAL ANGLE IS 2*THETA AND   HOLES     62
C*    AREA OF TRIANGLE FORMED BY THE CHORD AND RADII                    HOLES     63
       RSQ = RADHOL(I2)*RADHOL(I2)                                      HOLES     64
       AREAS = THETA*RSQ                                                HOLES     65
       AREAT = SQRT(RSQ - TERM*TERM)*ABS(TERM)                          HOLES     66
C*                                                                      HOLES     67
C*    EXPOSED AREA FROM ONE HOLE IS SECTOR AREA - TRIANGLE AREA OR THE  HOLES     68
C*    HOLE AREA MINUS THIS                                              HOLES     69
       AREAXP = (AREAS - AREAT)*FNH                                     HOLES     70
       IF(TERM .GT. 0.0) AREAXP = PI*RSQ*FNH - AREAXP                   HOLES     71
C*                                                                      HOLES     72
 20    AREAH(I1) = AREAH(I1) + AREAXP - TA(I2)                          HOLES     73
C*                                                                      HOLES     74
       TA(I2) = AREAXP                                                  HOLES     75
 30    CONTINUE                                                         HOLES     76
C                                                                       HOLES     77
       IF( AREAH(I1) .LT. 0.00001 ) AREAH(I1) = 0.0                     HOLES     78
 40    CONTINUE                                                         HOLES     79
C                                                                       HOLES     80
C                                                                       HOLES     81
C     CALCULATE AH(I), THE BELL TUBE AREA IN GRID I.  AH(I) DOES NOT    HOLES     82
C     INCLUDE AREA OF PSEUDO-HOLES.                                     HOLES     83
       CALL CLEAR(AH(1),AH(60))                                         HOLES     84
       DO 60 I=IBEGB,IENDB                                              HOLES     85
           AH(I) = AREAH(I)                                             HOLES     86
 60    CONTINUE                                                         HOLES     87
C                                                                       HOLES     88
C     IF THERE IS A PSEUDO HOLE AT GRID IBEGB OR AT GRID IENDB, ITS     HOLES     89
C     AREA SHOULD NOT HAVE BEEN INCLUDED.                               HOLES     90
       APSEUD = PI*RADPS*RADPS                                          HOLES     91
       IF(IPS2 .EQ. IENDB) AH(IENDB) = AH(IENDB) - APSEUD               HOLES     92
       IF(AH(IENDB) .LT. 0.) AH(IENDB) = 0.                             MOD0925   50
       NN=IBEGB+1                                                       MOD0925   51
       DO 70 I=NN,NCX                                                   MOD0925   52
C                                                                       HOLES     93
C     CALCULATE FRACT(I), THE FRACTIONAL VOLUME OF A GRID NOT UNDER THE HOLES     94
C     INFLUENCE OF THE BELL TUBE HOLES.  THE VOLUME CONSIDERED TO BE    HOLES     95
C     INFLUENCED IS BASED ON THREE TIMES THE HERMISPHERICAL VOLUME      HOLES     96
C     COMPUTED FROM THE RADIUS OF A TUBE HOLE.                          HOLES     97
           FRACT(I) = 1.0 - 2.0*SQRT(AH(I)*AH(I)*AH(I)/PI)/(AREAR(I)*DX) HOLES    99
           IF(FRACT(I) .LT. 0.0) FRACT(I) = 0.0                         HOLES    100
 70    CONTINUE                                                         HOLES    101
       RETURN                                                           HOLES    102
       END                                                              HOLES    103
      SUBROUTINE AREAS                                                  AREAS       2
      COMMON/BARRL2/DENBAR,XP,VP,BORED,BCHER,BORED2,DT2BU,DTDSQ,XLBAR   AREAS       3
      COMMON/CHAM/IX,IR,XI,NN,NGX,NGR,IEDGE,IENDR,IPATH(60,5),AREAG(5), CHAMIX      2
     1  AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,    CHAMIX      3
     2  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX      4
     3  AREAM2,DIAMBT,BLEEDD,BLEEDG,IPS1,IPS2,RADFS,BPIGN               CHAMIX      5
      COMMON/EQNS/CTTS,T2BR,T2BX,TWOTDR,DTBR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS        2
     1  X,DR,RX,QCPT,QCT,HBP                                            EQNS        3
      COMMON/P/IPRINT,PDUCH,HOUGR,PRI1,IDEBUG(35)                       P           2
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,LPIGN                        CHAMLOG     2
      LOGICAL PRI1,IDEBUG                                               PLOG        2
      DATA PIOF/.7853982/                                               AREAS       9
C                                                                       AREAS      10
C     SUBROUTINE AREAS CALCULATES CROSS-SECTIONAL AREAS AND VOLUME      AREAS      11
C     INCREMENTS ASSOCIATED WITH THE CHAMBER.                           AREAS      12
C                                                                       AREAS      13
C         AREAAX - AREA OF BELL IGNITER TUBE                            AREAS      14
C         AREAGP(I) - AREA OF GAP AT TOP OF CHAMBER AT AXIAL GRID I     AREAS      15
C         AREAC(I) - AREA OF ENTIRE CHAMBER AT AXIAL GRID I WHEN CHAMBER AREAS     16
C               BECOMES ONE DIMENSIONAL.  SO THAT CALCULATIONS IN AXIS GO AREAS    17
C               SMOOTHLY, AREAC(I) IS INITIALLY SET TO 1.0 IF CHAM1 IS  AREAS      18
C               TRUE AND AREAAX OTHERWISE.                              AREAS      19
C         AREAR(I) - AREA OF ENTIRE CHAMBER MINUS THE BELL IGNITER TUBE AREAS      20
C               AT AXIAL GRID I IF CHAM1 OR CHAM2 IS TRUE.  IF CHAM3 IS AREAS      21
C               TRUE, AREAGP(I) IS ALSO SUBTRACTED.                     AREAS      22
C         AREAG(J) - GENERALIZED AREA OF RADIAL GRID J, USED FOR        AREAS      23
C               SETTING DUMMY BARREL GRID 1 PROPERTIES.  IF CHAM2 IS    AREAS      24
C               TRUE AREAG(1) IS AREAAX AND AREAG(2) IS AREAR(NGX).     AREAS      25
C               IF CHAM3 IS TRUE AREAG(1) AND AREAG(2) ARE THE SAME AS  AREAS      26
C               WHEN CHAM2 IS TRUE AND AREAG(3) IS AREAGP(NGX).  IF CHAM1 AREAS    27
C               IS TRUE, AREAG(J) IS SET USING DR.                      AREAS      28
C         AREACH = AREAG(1) + . . . + AREAG(NGR)                        AREAS      29
C     IF CHAM1 IS TRUE, AREAC IS USED FOR MAKING THE CHAMBER ONE        AREAS      30
C     DIMENSIONAL AT EVERY I.  IF CHAM2 IS TRUE, AREAAX AND AREAR(I) ARE AREAS     31
C     USED AT GRID I.  IF CHAM3 IS TRUE, AREAGP(I) IS ALSO USED.        AREAS      32
C     FOR DOTMIG AND DOTMEG TERMS VOLUME INCREMENTS ARE NEEDED.         AREAS      33
C         DVAXIS - VOLUME INCREMENT OF GRIDS FOR J = 1.                 AREAS      34
C         DVAXIT - VOLUME INCREMENT OF GRIDS FOR J = 2.  DVAXIT WOULD   AREAS      35
C               NEED TO BE AN ARRAY FOR CHAM2 OR CHAM3 TRUE BUT IT IS NOT AREAS    36
C               NEEDED THEN.                                            AREAS      37
C                                                                       AREAS      38
C                                                                       AREAS      39
      AREAAA = PIOF*DIAMBT*DIAMBT                                       AREAS      40
C                                                                       AREAS      41
C     CALCULATE ARRAY AREAC AND FROM IT GET AREAR                       AREAS      42
C     CALCULATE ARRAY AREAGP IF CHAM3 IS TRUE.                          AREAS      43
C     ALSO CALCULATE AVERAGE DIAMETER TO BE USED IN CALCULATING PSEUDO- AREAS      44
C     HOLES IF CHAM3 IS TRUE.                                           AREAS      45
      DAVG = 0.0                                                        AREAS      46
      TWOGAP = 2.0*TOPGAP                                               AREAS      47
      XP = 0.0                                                          AREAS      48
C                                                                       AREAS      49
C     THE CHAMBER DIAMETER IS DIAM1 UP TO DIS1                          AREAS      50
      DIAMSQ = DIAM1*DIAM1                                              AREAS      51
      IF(CHAM3) D2 = DIAM1 - TWOGAP                                     AREAS      52
      IF(CHAM3) D2SQ = D2*D2                                            AREAS      53
      DO 250 I=1,NGX                                                    AREAS      54
```

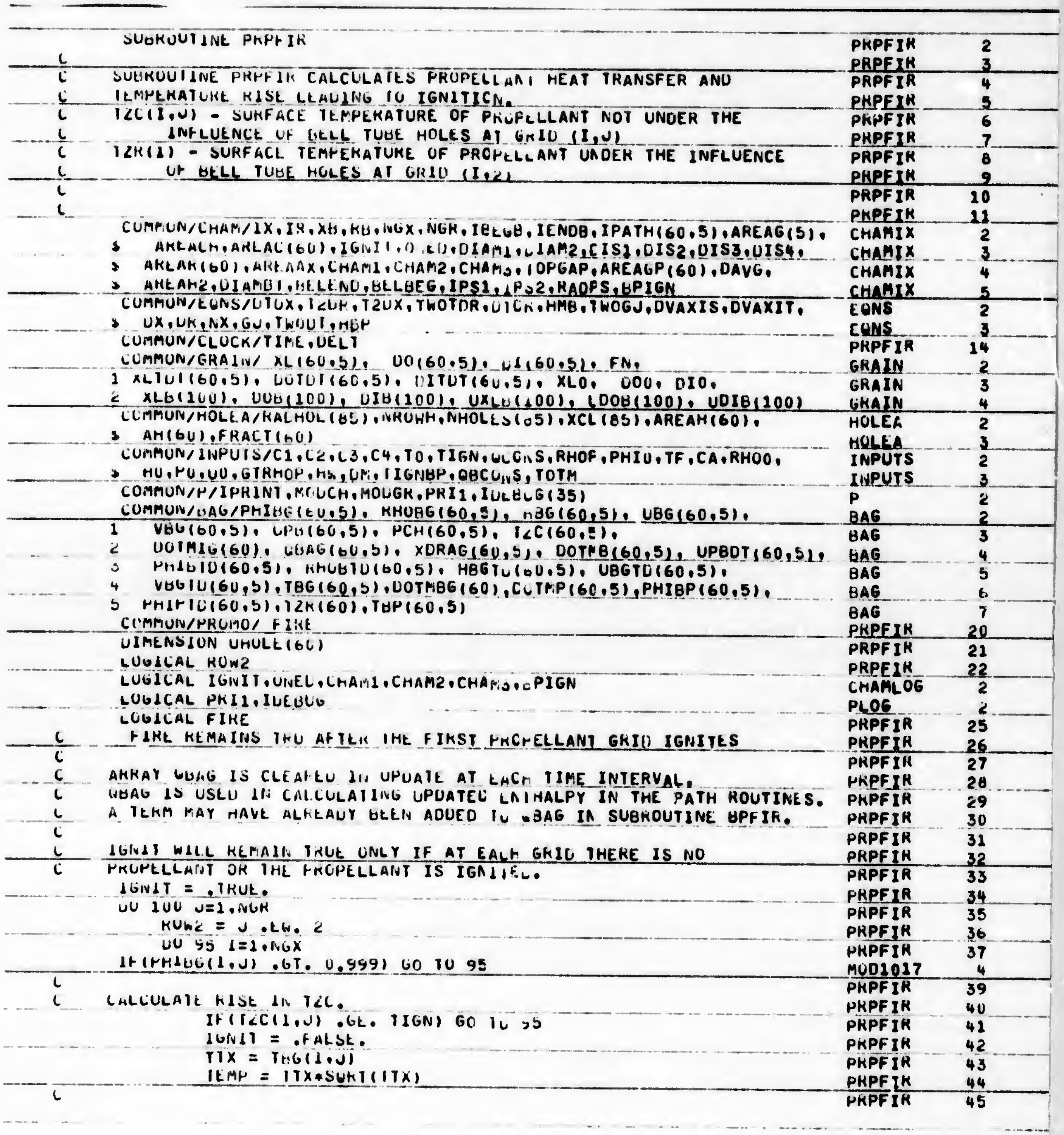
```
            ITEMP = I
            AREAC(I) = PIDF*DIAMSQ
            IF(CHAM3) AREAGP(I) = AREAC(I) - PIDF*D2SQ
            IF(CHAM3) DAVG = DAVG + DIAM1
            XB = XB + DX
            IF(XB .GT. DIS1) GO TO 260
  250   CONTINUE
        GO TO 350
C
C
C     THE CHAMBER DIAMETER DECREASES LINEARLY FROM DIS1 TO DIS2
  260   CONTINUE
        I = ITEMP
        I1 = I + 1
        IF(ABS(DIS1 - DIS2) .GT. 0.00001) GO TO 265
        SLOPE = 0.0
        GO TO 268
  265   SLOPE = (DIAM1 - DIAM2)/(DIS1 - DIS2)
  268   CONTINUE
        DO 270 I=I1,NGX
            ITEMP = I
            DIAM = DIAM1 + SLOPE*(XB - DIS1)
            IF(CHAM3) D2 = DIAM - TWOGAP
            AREAC(I) = PIDF*DIAM*DIAM
            IF(CHAM3) AREAGP(I) = AREAC(I) - PIDF*D2*D2
            IF(CHAM3) DAVG = DAVG + DIAM
            XB = XB + DX
            IF(XB .GT. DIS2) GO TO 280
  270   CONTINUE
        GO TO 350
C
C
C     THE CHAMBER DIAMETER IS DIAM2 BETWEEN DIS2 AND DIS3
  280   CONTINUE
        I = ITEMP
        I1 = I + 1
        DIAMSQ = DIAM2*DIAM2
        IF(CHAM3) D2 = DIAM2 - TWOGAP
        IF(CHAM3) D2SQ = D2*D2
        DO 290 I=I1,NGX
            ITEMP = I
            AREAC(I) = PIDF*DIAMSQ
            IF(CHAM3) AREAGP(I) = AREAC(I) - PIDF*D2*D2
            IF(CHAM3) DAVG = DAVG + DIAM2
            XB = XB + DX
            IF(XB .GT. DIS3) GO TO 300
  290   CONTINUE
        GO TO 350
C
C
C     THE CHAMBER DIAMETER DECREASES LINEARLY BETWEEN DIS3 AND DIS4
  300   CONTINUE
        I = ITEMP
        I1= I + 1
        IF(ABS(DIS3 - DIS4) .GT. 0.00001) GO TO 305
        SLOPE = 0.0
        GO TO 308
  305 SLOPE = (DIAM2 - BOREE)/(DIS3 - DIS4)
  309 CONTINUE
      DO 310 I=I1,NGX
         DIAM = DIAM2 + SLOPE*(XB - DIS3)
         IF(CHAM3) D2 = DIAM - TWOGAP
         AREAC(I) = PIQF*DIAM*DIAM
         IF(CHAM3) AREAGP(I) = AREAC(I) - PIQF*D2*D2
         IF(CHAM3) DAVG = DAVG + DIAM
         XB = XB + DX
  310 CONTINUE
C
  320 CONTINUE
      IF(CHAM3) DAVG = DAVG/FLOAT(NGX)
      NGP1 = NGX + 1
      AREAC(NGP1) = BOREA
      IF(CHAM3) AREAGP(NGP1) = BOREA - PIQF*(BOREE - TWOGAP)**2
C
C     FOR VERSION 3 AREAR(I) IS CALCULATED DIFFERENTLY THAN FOR VERSIONS
C     1 AND 2.
      IF(CHAM3) GO TO 365
C
      DO 360 I=1,NGP1
         AREAR(I) = AREAC(I) - AREAAX
  360 CONTINUE
      GO TO 375
C
  365 CONTINUE
      DO 370 I=1,NGP1
         AREAR(I) = AREAC(I) - AREAAX - AREAGP(I)
  370 CONTINUE
C
  375 CONTINUE
      IF(CHAM2 .OR. CHAM3) GO TO 400
C
C     CALCULATIONS WHEN CHAM1 IS TRUE
      DO 380 I=1,NGP1
         AREAC(I) = 1.0
  380 CONTINUE
C
      NGR1 = NGR - 1
      PIDRSQ = PI*DR*DR
      AREAG(1) = 0.25*PIDRSQ
      DO 390 J=2,NGR1
         AREAG(J) = 2.0*PIDRSQ*FLOAT(J-1)
  390 CONTINUE
      AREAG(NGR) = PIDRSQ*(FLOAT(NGR) - 1.25)
      AREACM = PIDRSQ*FLOAT(NGR1)**2
C****SHOULD THIS BE AREAG(1)*DX
      DVAXIS = PIDRSQ*DX
C     DVAXIT IS PI*(R + DR/4)*DR*DX
      DVAXIT = 1.25*DVAXIS
      GO TO 500
C
C
C     CALCULATIONS WHEN CHAM2 OR CHAM3 IS TRUE
  400 CONTINUE
      DO 410 I=1,NGP1
      AREAC(1) = AREAAX
  410 CONTINUE
      AREAG(1) = AREAAX
      AREAG(2) = AREAR(NGX)
      IF(CHAM3) AREAG(3) = AREAGP(NGX)
      AREACH = AREAG(1) + AREAG(2)
      IF(CHAM3) AREACH = AREACH + AREAG(3)
      DVAXIS = AREAAX*DX
C
C
  500 CONTINUE
C
      IF(.NOT. IDEBUG(8)) RETURN
      WRITE(6,2002)
      WRITE(6,2003) (AREAR(I),I=1,NGP1)
      WRITE(6,2004)
      WRITE(6,2003) (AREAC(I),I=1,NGP1)
      WRITE(6,2005) (AREAG(I),I=1,NGR)
      IF(CHAM3) WRITE(6,2006)
      IF(CHAM3) WRITE(6,2003) (AREAGP(I),I=1,NGP1)
      RETURN
C
 2002 FORMAT(///,' ARRAY AREAR',//)
 2003 FORMAT(9X,10F11.7,/)
 2004 FORMAT(///,' ARRAY AREAC',//)
 2005 FORMAT(///,' ARRAY AREAG',//,20X,5F11.7)
 2006 FORMAT(///,' ARRAY AREAGP',//)
      END
      SUBROUTINE BARSET                                                 BARSET     2
      COMMON/BARRL2/BOREA,XP,VP,BOREO,BORER,BOREDP,DT2BO,DTOSQ,XLBAR     BARSET     3
      COMMON/CLOCK/TIME,DELT                                            BARSET     4
      COMMON/GRIDNX/DXPRIM                                              MODD301   23
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $ DX,DR,NX,GO,TWODT,HPP                                            EQNS       3
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTOT(60,5), DOTOT(60,5), DITOT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN      4
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,QLCNS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,QBCONS,IOTM                        INPUTS     3
      COMMON/MOCON/CON1,CON2,CON3,CON4,CON5,AREAPB,ZO,WOB,XOB,ROTK,     BARSET     8
     $  FOMAX,CON6,XINT,XLO,PINT,PLO                                    MOD1111    2
      COMMON/P/IPRINT,MODCH,MODCR,PRI1,IDEBUG(35)                       P          2
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL      2
     1  PG(100), TG(100),  PMDOT(100), GL(100), DDRAG(100), FRICT(100), BARRL      3
     2  QCONV(100), DUP(100), DPHI(100), DRHOG(100), DHG(100), DUG(100),BARRL      4
     3  AMASS(100), AMOM(100), AENER(100), DAMASS(100), DAMOM(100),     BARRL      5
     4  DAENER(100)                                                     BARRL      6
      LOGICAL PRI1,IDEBUG                                               PLOG       2
      DATA GRAV/32.16/                                                  BARSET    13
C                                                                       BARSET    14
      NAMELIST/BARINP/THETA,DEPTH,RADPB,PL,WOB,PMASS,PINER,ROTK,CF,     MOD1111    3
     $ NX,XLBAR,XMAX,XINT,XLO,PZO,POMAX,PINT,PLO                        MOD1111    4
      READ(5,BARINP)                                                    BARSET    17
      IF(IDEBUG(7)) WRITE(6,BARINP)                                     BARSET    18
C                                                                       BARSET    19
      RADPB = RADPB/12.                                                 BARSET    20
      XLBAR = XLBAR/12.                                                 BARSET    21
      WOB = WOB/12.                                                     BARSET    22
      DEPTH = DEPTH/12.                                                 BARSET    23
      XOB=0.                                                            MOD1111    5
      XMAX=XMAX/12.                                                     MOD1111    6
      XINT=XINT/12.                                                     MOD1111    7
      XLO=XLO/12.                                                       MOD1111    8
      PZO=PZO*144.                                                      MOD1111    9
      POMAX=POMAX*144.                                                  MOD1111   10
      PINT=PINT*144.                                                    MOD1111   11
      PLO=PLO*144.                                                      MOD1111   12
C                                                                       BARSET    25
C                                                                       BARSET    26
C*********************************************************************** BARSET    27
C    CONSTANTS FOR BARREL SUBROUTINES                                   BARSET    28
C*********************************************************************** BARSET    29
C                                                                       BARSET    30
C    CONSTANTS FOR SUBROUTINE MOTION                                    BARSET    31
      CS = COS(THETA)                                                   BARSET    32
      SN = SIN(THETA)                                                   BARSET    33
      CON1 = CS - CF*SN                                                 BARSET    34
      CON2 = RADPB/(PMASS/GRAV*RADPB*CON1 + PINER*ROTK*(SN + CF*CS))    BARSET    35
      TEMP = DEPTH*PL                                                   BARSET    36
      AREAPB = 3.141593*RADPB*RADPB                                     BARSET    39
      CON5 = 0.5*DELT*DELT                                              BARSET    40
      CON3=(POMAX-PZO)*AREAPB/XMAX                                      MOD1111   13
      CON4=(FOMAX-PINT)*AREAPB/(XINT-XMAX)                              MOD1111   14
      FOMAX = POMAX*AREAPB                                              MOD1111   15
      ZO = PZO*AREAPB                                                   MOD1111   16
      CONE=((PINT-PLO)/(XLO-XINT))*AREAPB                               MOD1111     17
C                                                                       BARSET      41
C     INITIALLY THE PROJECTILE IS NOT MOVING                            BARSET      42
       XP = XOB                                                         BARSET      43
       VP = 0.0                                                         BARSET      44
       DXPRIM = DX                                                      MOD0301     24
C                                                                       BARSET      45
C                                                                       BARSET      46
C     CONSTANTS FOR SUBROUTINE BNDLYR                                   BARSET      47
       BORED8 = BORED/8.0                                               BARSET      48
       DTUSQ = DELT*BORED*BORED                                         BARSET      49
       DT2BD = -0.5*DELT/BORED                                          BARSET      50
C                                                                       BARSET      51
C                                                                       BARSET      52
C     CONSTANT FOR SUBROUTINE PROPMO                                    BARSET      53
       GTRHOP = GRAV*DELT/RHOP                                          BARSET      54
C                                                                       BARSET      55
C     FINISH TOTALLING 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)*RHO0*BOREA*DX                BARSET      59
       GO TO 30                                                         BARSET      60
  25   TOTM = TOTM + 0.5*RHO0*BOREA*DX                                  BARSET      61
  30   CONTINUE                                                         BARSET      62
C                                                                       BARSET      63
       NAMELIST/BARCHK/RALPB,WOB,DEPTH,BORED,XOB,CS,SN,CON1,CON2,       BARSET      64
     $  CON3,CON4,CON5,AREAPB,XP,VP,BOREA,G,RHOP,EORER,BORED8,DTUSQ,    BARSET      65
     $  DT2BD,TOTM                                                      BARSET      66
       IF(IDEBUG(8)) 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), UAENER(100))                                  BARSET      73
       CALL CLEAR(XLB(1), DD1B(100))                                    BARSET      74
C                                                                       BARSET      75
       IF(NX .EQ. 1) GO TO 60                                           BARSET      76
       DO 50 I=2,NX                                                     BARSET      77
       RHOG(I) = RHO0                                                   BARSET      78
       HG(I) = H0                                                       BARSET      79
       PG(I) = P0                                                       BARSET      80
       UG(I) = U0                                                       BARSET      81
       TG(I) = T0                                                       BARSET      82
       AMASS(I) = BOREA                                                 BARSET      83
       AMOM(I) = BOREA                                                  BARSET      84
       AENER(I) = BOREA                                                 BARSET      85
       UAMASS(I) = BOREA                                                BARSET      86
       UAMOM(I) = BOREA                                                 BARSET      87
       UAENER(I) = BOREA                                                BARSET      88
C                                                                       BARSET      89
C     RHODOT AND UP HAVE ALREADY BEEN CLEARED                           BARSET      90
  50   CONTINUE                                                         BARSET      91
C                                                                       BARSET      92
C     SET AREAS AT BARREL GRID 1 TO BORE AREA OF BARREL.                BARSET      93
  60   CONTINUE                                                         BARSET      94
       AMASS(1) = BOREA                                                 BARSET      95
      AMOM(1) = BOREA                                                   BARSET    96
      AENER(1) = BOREA                                                  BARSET    97
      DAMASS(1) = BOREA                                                 BARSET    98
      DAMOM(1) = BOREA                                                  BARSET    99
      DAENER(1) = BOREA                                                 BARSET   100
C                                                                       BARSET   101
C                                                                       BARSET   102
C     SET ALL ENTRIES IN ARRAYS PHI AND OPHI TO 1.0 SO THAT WHEN GRIDS AR BARSET   103
C     THERE WILL BE NO PROPELLANT IN THEM.                              BARSET   104
      DO 60 I=1,100                                                     BARSET   105
         PHI(I) = 1.0                                                   BARSET   106
         OPHI(I) = 1.0                                                  BARSET   107
   60 CONTINUE                                                          BARSET   108
      RETURN                                                            BARSET   109
      END                                                               BARSET   110
      SUBROUTINE MFLOW                                                  MFLOW      2
C                                                                       MFLOW      3
C                                                                       MFLOW      4
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAM2,DIAMPI,BELLEN,BELBEG,IPS1,IPS2,RADPS,BPIGN               CHAMIX     5
      COMMON/GASCON/R0,RHO,CV0,CVH                                      MFLOW      6
      COMMON/CLOCK/TIME,DELT                                            MOD1002    2
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA      2
     $  AH(60),FRACT(60)                                                HOLEA      3
      COMMON/INPUTS/C1,C2,C3,C4,F0,TIGN,DECONS,RHOF,PHI0,TF,CA,RHO0,    INPUTS     2
     $  H0,P0,U0,GTRHOP,HW,DH,TIGNBP,QBCONS,TOTM                        INPUTS     3
      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   DOTMIG(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),COMP(60,5),PHIBP(60,5),       BAG        6
     5  PHIPTD(60,5),TZR(60),TBP(60,5)                                  BAG        7
      DIMENSION DOTM(60)                                                MFLOW     10
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        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 (I,1) AND (I,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 (I,2) AND (I,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
      NGX1 = NGX - 1                                                    MFLOW     29
      ID = 1                                                            MFLOW     30
      IF(CHAM3) ID = 2                                                  MFLOW     31
      J = 1                                                             MFLOW     32
      GO TO 20                                                          MFLOW     33
C                                                                       MFLOW     34
    5 ID = 1                                                            MFLOW     35
      J = 2                                                             MFLOW     36
C                                                                       MFLOW     37
   20 CONTINUE                                                          MFLOW     38
      JP1 = J + 1                                                       MFLOW     39
      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
      DO 60 I=1,NGX                                                     MFLOW     46
C                                                                       MFLOW     47
```

```
          PR = PCH(I,J)
          PA = PCH(I,JP1)
C
C*    IF PR=PA THERE IS NO FLOW
          IF( ABS(PR - PA) .LT. 0.001 ) GO TO 80
C
C*    DETERMINE THE DIRECTION OF THE FLOW.  IF PR IS GREATER THAN PA,
C*    GAS FLOWS OUT OF THE TUBE. OTHERWISE IT FLOWS INTO THE TUBE.
          IF( PR .LT. PA ) GO TO 40
          HRR = HBG(I,J)
C
C*    PCOMP IS AN APPROXIMATE STATIC PRESSURE FOR CHOKED (MACH NO. 1)
C*    FLOW.  ASSUME PCOMP IS NOT LESS THAN PA.
          PCOMP = 0.53*PR
          IF( PCOMP .LT. PA ) PCOMP = PA
C
          CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCOMP,HRR,TDUM,RHODUM,
     $                UBG(I,J),0.0,GMM,CP,3)
          GMMM1 = GMM - 1.0
          CONS1 = 2.0/(GMM + 1.0)
          PWR = GMM/GMMM1
          CONS2 = 2.0/GMMM1
C
C*    PSTAT IS SONIC STATIC PRESSURE
           PSTAT = PR*CONS1**PWR
C
C*    FM IS MACH NUMBER.  IF PA IS LESS THAN PSTAT, FM=1.  OTHERWISE
C*    FM IS NOT 1. AND MUST BE CALCULATEC.
          FM = 1.0
          IF (PA .LT. PSTAT) GO TO 10
          FM = SQRT( ((PR/PA)**(GMMM1/GMM) - 1.0)*CONS2 )
          PSTAT = PA
C
 10       HSTAT = HRR/(1.0 + FM*FM/CONS2)
C
C     SINCE GAS IS FLOWING FROM GRID (I,J) TO GRID (I,J+1), DOTM SHOULD
C     BE NEGATIVE.
C****IS PHIBG(I,2) OKAY
          DOTM(I) = -.203*GMM*PSTAT*FM*CA/SQRT(GMMM1*HSTAT*PHIBG(I,2))
          GO TO 80
C
C
C
C*    THE CODING WHEN PR IS LESS THAN PA IS ESSENTIALLY THE SAME AS ABOVE
C     WITH PR AND PA INTERCHANGED.
 40       CONTINUE
          HAA = HBG(I,JP1)
          PCOMP = 0.53*PA
          IF( PCOMP .LT. PR ) PCOMP = PR
          CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCOMP,HAA,TDUM,RHODUM,
     $                UBG(I,JP1),0.0,GMM,CP,3)
C
          GMMM1 = GMM - 1.0
          CONS1 = 2.0/(GMM + 1.0)
          PWR = GMM/GMMM1
           CONS2 = 2.0/GMMM1
C
```

```
      PSTAT = PA*CONS1**PWR                                             MFLOW    105
C                                                                       MFLOW    106
      FM = 1.0                                                          MFLOW    107
      IF( PR .LT. PSTAT ) GO TO 50                                      MFLOW    108
      FM = SQRT( ((PA/PR)**(GMMM1/GPM) - 1.0)*CONS2 )                   MFLOW    109
      PSTAT = PR                                                        MFLOW    110
 50   HSTAT = HAA/(1.0 + FM*FM/CONS2)                                   MFLOW    111
C                                                                       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*PSTAT*FM*CA/SQRT(GMMP1*HSTAT*PHIBG(I,2))       MFLOW    116
C                                                                       MFLOW    117
 80   CONTINUE                                                          MFLOW    118
      IF(CHAMS .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(1) = 0.5*(DOTM(1) + DOTM(2))*AREAH(1)                      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(1) = 0.5*(DOTM(1) + 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
      SUBROUTINE BPFIR                                                  BPFIR        2
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTCR,DTDR,HMB,TWCGJ,DVAXIS,DVAXIT,   MOD0925     53
     $  DX,DR,RX,GJ,TWODT,HBP                                           MOD0925     54
      COMMON/BPI/TEP(2),DEP(2),CEP(2),FTUBR,PDALST                      MOD0925     55
      COMMON/PRMFLO/DETPRM,DPRM,QPRM,PRPXP,PDSHR,CSPRKS,PRMCOF          MOD0925     56
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX       2
     $   AREACH,AREAC(60),IGNIT,OPEN,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX       3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,DPGAP,AREAGP(60),DAVG,       CHAMIX       4
     $  AREAM,DIAMBT,BELENB,BELBEG,IPS1,IPS2,RADFS,BPIGN                CHAMIX       5
      COMMON/CLOCK/TIME,DELT                                            BPFIR        5
      COMMON/FRD/LF1,LF2,MF1,MF2,BGJAM,BGSTF,NVSP,IBGNP                 MOD0925     57
      COMMON/IGNITE/FDPAL,BFCHG,BRCUT,BGBRN,IMPBG,                      MOD0925     58
     1 BEGIC(5),XLTC(5)                                                 MOD0925     59
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       MOD0605      4
     1 THICK(60),PENET,TYPCL                                            MOD0925     60
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCCRS,RHOF,PHIO,TF,CA,RHOO,     INPUTS       2
     $  HO,PO,UO,GTRHOF,HW,DM,TIGNBP,GBCONS,TOTM                        INPUTS       3
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                       P            2
      COMMON/PRIMV/BPLENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP               PRIMV        2
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), PBG(60,5), UBG(60,5),        BAG          2
     1    VBG(60,5), UPB(60,5), PCH(60,5), T2C(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), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),         BAG          5
     4    VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTPP(60,5),PHIBP(60,5),     BAG          6
     5  PHIPTL(60,5),T2R(60),TBP(60,5)                                  BAG          7
      LOGICAL IGNIT,OPEN,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG      2
      LOGICAL PRI1,IDEBUG                                               PLOG         2
C                                                                       BPFIR       12
C     ARRAY QBAG IS CLEARED IN UPDATE EACH TIME INTERVAL.               BPFIR       13
C     BPIGN WILL REMAIN TRUE ONLY IF AT EACH GRID THERE IS NO BLACK     BPFIR       14
C     POWDER OR THE BLACK POWDER IS IGNITED.                            BPFIR       15
      LOGICAL PENET                                                     MOD0604      4
      LOGICAL BGBRN                                                     MOD0925     61
      STDOFF=BELBEG-TEP(1)                                              MOD0925     62
      IBEGFP=STDOFF/DX + 1.5                                            MOD0925     63
      IF(IBEGFP  .LT. 1) IBEGFP=1                                       MOD0925     64
      IF(TBP(IBEGFP,1) .LT. TIGNBP) GO TO 10                            MOD0925     65
      BPIGN=.TRUE.                                                      MOD0925     66
C  IF THE IGNITER PAD IS PENETRATED BY EITHER BURNING OR PRIMER JET BLAS MOD0925     67
C  PENET IS TRUE AND THE FOLLOWING BLAST EVALUATIONS WILL               MOD0925     66
C  BE BYPASSED.                                                         MOD0925     69
   10 CONTINUE                                                          MOD0925     70
      IF(TEP(1) .EQ. 0. .OR. DEP(1) .EQ. 0.) GO TO 30                   MOD1002      3
      IF(PENET) GO TO 30                                                MOD0605      7
      RAPJT=FTUBR*(1.+(STDOFF/FTUBR)**.815)                             MOD0925     71
      PRMXP=2060.E6*(EXP(-.635*STDOFF/FTUBR))*TIME                      MOD0925     72
      IF(TIME .GE. .0004) PRMXP=1.33E6*(EXP(-.635*STDOFF/FTUBR-1200.    MOD0925     73
     1 *TIME))                                                          MOD0925     74
      PRMXP=PRMCOF*PRMXP                                                MOD0925     75
      PDSHR = TEP(1)*PDALST                                             MOD1611      1
      JTSHR=PRMXP*RAPJT/2.                                              MOD0605     12
      IF(BGBRN) GO TO 20                                                MOD0925     76
      IF(JTSHR .LT. PDSHR) GO TO 30                                     MOD1611      2
   20 CONTINUE                                                          MOD0925     77
      WRITE(6,25) TIME                                                  MOD0925     78
   25 FORMAT(/,4X,6HTIME = E14.7,3X,27HBASE IGNITER PAD PENETRATED)     MOD0925     79
      PENET=.TRUE.                                                      MOD0605     16
      STDOFF=DELBEG                                                     MOD0605    17
      BGBRN=.TRUE.                                                      MOD0925    80
      DWDP=CEP(1)*(DIAMBT/CEP(1))**2.                                   MOD0925    81
      PHIBP(IBEGFP,1) = 1.0                                             MOD0925    82
      PHIBP(IBEGFP +1,1) = 1.                                           MOD1115     3
      PHIBP(IBEGGB,1)=PHIBP(IBEGGB,1)-DWDP/BPDENS*AREAAX*DX             MOD1115     4
      PHIPTO(IBEGFP+1,1) = PHIBP(IBEGFP+1,1)                            MOD0925    86
      PHIPTO(IBEGFP,1)=PHIBP(IBEGFP,1)                                  MOD0925    87
      PHIPTO(IBEGGB,1)=PHIBP(IBEGGB,1)                                  MOD1115     5
   50 CONTINUE                                                          MOD0605    23
      FLAGI=0.                                                          MOD0925    88
      DO 100 J=1,NCR                                                    BPFIR      17
      IF(J .EQ. 2) FLAGJ=0.                                             MOD0925    89
      DO 100 I=1,NGX                                                    BPFIR      18
      IF(PHIBP(I,J) .GE. .999) GO TO 100                                MOD1002     5
          IF(TBP(I,J) .GE. TIGNBP) GO TO 100                            BPFIR      20
          BPIGN = .FALSE.                                               BPFIR      21
C                                                                       BPFIR      22
C    GAS TEMPERATURE                                                    BPFIR      23
          TTX = TBG(I,J)                                                BPFIR      24
          TEMP = 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 = RHOBG(I,J)*ABS(UBG(I,J))*BPRAD(I,J)/TXMU             BPFIR      34
      IF(RETXBP .LT. .00001) RETXBP=.00001                              MOD1017     3
C                                                                       BPFIR      35
C    NUSSELT NUMBER                                                     BPFIR      36
          TXNUBP = 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
          ADVBP = 3.0*(1.0 - PHIBP(I,J))/BPRAD(I,J)                     BPFIR      41
C                                                                       BPFIR      42
C    HEAT FLUX TO BLACK POWDER                                          BPFIR      43
          QCONBP = TXNUBP*TXK/BPRAD(I,J)*(TTX - TBP(I,J))               BPFIR      44
      IBEGC=IBEGB                                                       MOD0925    90
      IF( .NOT. BGBRN .AND. TBP(IBEGFP,1) .GE. TIGNBP) GO TO 50         MOD0925    91
      IF(FLAGI .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)QPMR=3260.*EXP(-30.5*DISPR-600.*TIME)         MOD0925    94
     $ + QSPRKS                                                         MOD0925    95
      IF(TIME .LT. 0.0004) QPMR=1.28E5*SQRT(TIME)*EXP(-30.5*DISPR)      MOD0925    96
     $ + QSPRKS                                                         MOD0925    97
      QPMR=SQRT(PRRCOF)*QPMR                                            MOD0925    98
      IF(QPMR.GT.QCONBP) QCONBP=QPMR                                    BPFIR      54
      IF(J .EQ. 1) FLAGI=1.                                             MOD0925    99
      FLAGJ=1.                                                          MOD0925   100
      IF(IDEBUG(35) .AND. PRI1) WRITE(6,97)QPMR,DISPR,I,J               MOD0925   101
```

```
 97 FORMAT(/,4X,7HOPPR = E14.5,3X,6HDISPR = E14.5,3X,4HI = I2,3X,         MOD0925    102
   $ 4HJ = I2)                                                           MOD0925    103
 50 CONTINUE                                                             BPFIR       55
          IF(GCONBP .LT. 0.001) GO TO 100                                BPFIR       56
          GBAG(I,J) = GCONBP*ADVBP                                       BPFIR       57
      TEMP=2.0*QBCONS*GCONBP                                             MOD0925    104
          TEFFBP = ((TBP(I,J) - TO)/TEMP)**2                             BPFIR       59
          IF(TEFFBP .GT. TIME) TEFFBP = TIME                             BPFIR       60
          TBP(I,J) = TBP(I,J) +                                          BPFIR       61
     $         TEMP*(SQRT(TEFFBP + DELT) - SQRT(TEFFBP))                 BPFIR       62
      IF(PORAD .LE. 0.6*DIAMGT) GO TO 90                                 MOD1215      4
      TBP(IBEGB,2)=TIGNBP                                                MOD1215      5
      TBP(IBEGFP,2)=TIGNBP                                               MOD1215      6
 90 CONTINUE                                                             MOD1215      7
          IF(TBP(I,J) .LT. TIGNBP) GO TO 100                             BPFIR       63
          IF(IDEBUG(9)) WRITE(6,2000) TIME,I,J                           BPFIR       64
100   CONTINUE                                                           BPFIR       65
      IF(PENET) GO TO 150                                                MOD0925    105
      IF(TBP(IBEGFP,J) .LT.   760. ) GO TO 150                           MOD1115      6
      IF(TBP(IBEGFP+1,J) .LT.   760.) GO TO 150                          MOD1115      7
      BGBRN=.TRUE.                                                       MOD1115      8
 150 CONTINUE                                                            MOD0925    116
C                                                                        BPFIR       66
      RETURN                                                             BPFIR       67
C                                                                        BPFIR       68
2000 FORMAT(/,' TIME =',E14.8,' BLACK POWDER AT GRID',I4,I4,             BPFIR       69
   $  ' IS IGNITED')                                                     BPFIR       70
      END                                                                BPFIR       71
      SUBROUTINE PRIMER                                                 PRIMER     2
      COMMON/CALLP/BPLEFT                                               PRIMER     3
      COMMON/IGNTL/PDRAD,BPCHG,BRNOUT,BGBRN,TMPBG,                      MOD1002    6
     $ BEGIC(5),XLIC(5)                                                 MOD1002    7
      COMMON/PRMFLO/DOTMPM,DPRM,QPRM,PRPAP,POSHR,CSPRKS,PRMCOF          MOD0925  118
      COMMON/CLOCK/TIME,DELT                                            PRIMER     5
      COMMON/BPT/TEP(2),DEP(2),CEP(2),FTOGH,PDALST                      MOD0925  119
      COMMON/FAILEL/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       MOD0925  120
     1 THICK(60),PENET,TIFCE                                            MOD0925  121
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $    AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,  CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAB2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,WCONS,RHOF,PHI0,TF,CA,RHO0,     INPUTS     2
     $  H0,P0,U0,GTRHOP,MW,DM,TIGNBP,QBCONS,TOTM                        INPUTS     3
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,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   DOTMIG(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),DDTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),T2R(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
      LOGICAL BPLEFT                                                    PRIMER    12
      LOGICAL PENET                                                     MOD0605   24
      LOGICAL BRNOUT,BGBRN                                              MOD0925  122
      LOGICAL PRI1                                                      MOD0925  123
      DATA FORTPI/4.188790/                                             PRIMER    13
C                                                                       PRIMER    14
C                                                                       PRIMER    15
C                                                                       PRIMER    16
C     CALCULATE PRIMER VELOCITY AND MASS FLOW RATE                      PRIMER    17
C                                                                       PRIMER    18
      IBB2=IBEGB                                                        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
      BPLEFT = .FALSE.                                                  PRIMER    28
C                                                                       PRIMER    29
      DO 60 J = 1,NGR                                                   PRIMER    30
      DO 60 I = 1,NGX                                                   PRIMER    31
      IF(PENET .AND. J .EQ. 1 .AND. I .EQ. IBB2) GO TO 42               MOD1215    8
         IF (PHIBP(I,J) .GE. 0.999) GO TO 55                            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 55                              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
         R = AGENBP*PCH(I,J)**EXPBP + BGENBP                            PRIMER    38
         BURNL = R*DELT                                                 PRIMER    39
         VOLD = FORTPI*BPRAD(I,J)**3                                    PRIMER    40
         BPRAD(I,J) = BPRAD(I,J) - BURNL                                PRIMER    41
```

```
            IF(BPRAD(I,J).GE. 0.001) GO TO 40                           PRIMER      42
            WRITE(6,2000) I,J,BPRAD(I,J)                                PRIMER      43
 35         BPRAD(I,J) = 0.0                                            PRIMER      44
            PHIBP(I,J) = 1.0                                            PRIMER      45
            GO TO 55                                                    PRIMER      46
C                                                                       PRIMER      47
 40         CONTINUE                                                    PRIMER      48
            VNEW = FORTPI*BPRAD(I,J)**3                                 PRIMER      49
            DELTAV = VOLD - VNEW                                        PRIMER      50
            PROV =  (1.0 - PHIBP(I,J))/VOLD                             PRIMER      51
            TEMP = PROV*DELTAV                                          PRIMER      52
            DOTMP(I,J) = TEMP*[illegible]                               PRIMER      53
C   NOW COMPUTE EFFECT OF POROSITY OF FLAME SPREAD                      MOD0925    128
C   AND COMPARE WITH EFFECT OF BURNING RATE                             MOD0925    129
      IF(I .EQ. IBB2 .AND. J .EQ. 1) GO TO 42                           MOD0925    130
      GO TO 48                                                          MOD0925    131
   42 CONTINUE                                                          MOD0925    132
      IF(BRNOUT) GO TO 46                                               MOD0925    133
      PRESC=PCH(IBB2,2)/144.                                            MOD0925    134
      IF(PRESC .LT. 0.) PRESC=0.                                        MOD0925    135
      ABDOT=.00004*(PRESC**2.2)                                         MOD1002      8
      DELTR=0.159*(ABDOT/TEP(1))*DELT                                   MOD0925    137
      PDRAD=DELTR + PDRAD                                               MOD0925    138
      IF(PDRAD .LT. .5*DEP(1)) GO TO 45                                 MOD0925    139
      WRITE(6,430)TIME                                                  MOD0925    140
  430 FORMAT(/,3X,7HTIME = E14.7,3X,23HIGNITER PAD BURNED AWAY)         MOD0925    141
      PHIBP(IBB2,2) = 1.0                                               MOD0925    142
      PHIPTD(IBB2,2)=1.0                                                MOD0925    143
      BRNOUT = .TRUE.                                                   MOD0925    144
      GO TO 46                                                          MOD0925    145
C         THE ROUTE TO STMT 48 MUST BE TAKEN TO                         MOD0925    146
C    CONTINUE COMPUTATION FOR GRID(1,1).  LOGIC                         MOD0925    147
C    FROM STMT 42 ON TO HERE CONCERNS J=2 GRIDS ONLY                    MOD0925    148
   45 CONTINUE                                                          MOD0925    149
   46 PHIIP=PHIBP(IBB2,2)                                               MOD0925    150
   47 IF(PHIIP .GT. .9999) PHIIP=.9999                                  MOD0925    151
      DLVOL=6.283*PDRAD*DELTR                                           MOD0925    152
      IF(DLVOL .LE. 0.)DLVOL=.001                                       MOD0925    153
      DENS=(CEP(1)/TEP(1))/(0.785*(DEP(1)*DEP(1)-DIAMBT**2.))           MOD0925    154
      DIMP=TEP(1)*DENS*DLVOL                                            MOD0925    155
      DOTMP(IBB2,2) = DOTMP(IBB2,2) + DIMP                              MOD0925    156
      PHIBP(IBB2,2)=PHIBP(IBB2,2)+DLVOL                                 MOD0925    157
      PHIPTD(IBB2,2) = PHIBP(IBB2,2)                                    MOD0925    158
   48 CONTINUE                                                          MOD0925    159
            PHIPTD(I,J) = PHIBP(I,J) + TEMP                             PRIMER      54
      GO TO 58                                                          MOD0925    160
C                                                                       PRIMER      56
 55         PHIPTD(I,J) = PHIBP(I,J)                                    PRIMER      57
   58 IF(PHIBP(I,J) .GT. .9999) PHIBP(I,J)=1.0                          MOD0925    161
      IF(PHIPTD(I,J) .GT. .9999) PHIPTD(I,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 IGNITICN.                             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,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONEED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,  CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/CLOCK/TIME,DELT                                            PRPFIR    14
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), UXLB(100), UDOB(100), UDIB(100)   GRAIN      4
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA      2
     $  AH(60),FRACT(60)                                                HOLEA      3
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,QCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        INPUTS     3
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,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   DOTMIG(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),COTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(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,ONEB,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
      LOGICAL PRI1,IDEBUG                                               PLOG       2
      LOGICAL FIRE                                                      PRPFIR    25
C      FIRE REMAINS TRU AFTER THE FIRST PROPELLANT GRID IGNITES         PRPFIR    26
C                                                                       PRPFIR    27
C     ARRAY QBAG IS CLEARED IN UPDATE AT EACH TIME INTERVAL.            PRPFIR    28
C     QBAG IS USED IN CALCULATING UPDATED ENTHALPY IN THE PATH ROUTINES. PRPFIR   29
C     A TERM MAY HAVE ALREADY BEEN ADDED TO QBAG IN SUBROUTINE BPFIR.   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
            TTX = TBG(I,J)                                              PRPFIR    43
            TEMP = TTX*SQRT(TTX)                                        PRPFIR    44
C                                                                       PRPFIR    45
```

```
C     GAS VISCOSITY                                                      PRPFIR    46
              TXMU = C1*TEMP/(TTX + C2)                                  PRPFIR    47
C                                                                        PRPFIR    48
C     THERMAL CONDUCTIVITY OF GAS                                        PRPFIR    49
              TXK = C3*TEMP/(TTX + C4)                                   PRPFIR    50
C                                                                        PRPFIR    51
C     REYNOLDS NUMBER                                                    PRPFIR    52
              TEMP1 = RHOBG(I,J)*DM/TXMU                                 PRPFIR    53
              RETX = TEMP1*ABS(UBG(I,J))                                 PRPFIR    54
       IF(RETX .LT.  .00001) RETX=.00001                                 MOD1017    5
C                                                                        PRPFIR    55
C     NUSSELT NUMBER                                                     PRPFIR    56
              TEMP2 = 0.5*SQRT(PCH(I,J)/144.)                            PRPFIR    57
              TXNUS = 0.3*RETX**0.62 + TEMP2                             PRPFIR    58
C                                                                        PRPFIR    59
C     SURFACE AREA OF PROPELLANT IN GRID PER VOLUME INCREMENT            PRPFIR    60
              ADV = 4.0 *(1.0 - PHIBG(I,J))*( (DO(I,J) + FN*DI(I,J))/    PRPFIR    61
     $             (DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J)) + 0.5/XL(I,J) ) PRPFIR    62
C                                                                        PRPFIR    63
C     HEAT FLUX TO PROPELLANT--BTU/FT**2 - SEC                           PRPFIR    64
              TEMP3 = TXK/DM                                             PRPFIR    65
              QCONVA = TXNUS*TEMP3*(TTX - TZC(I,J))                      PRPFIR    66
              IF(QCONVA .LT. 0.001) GO TO 50                             PRPFIR    67
              TEMP = QCONVA*ADV                                          PRPFIR    68
              IF(ROW2) TEMP = TEMP*FRACT(I)                              PRPFIR    69
              QBAG(I,J) = QBAG(I,J) + TEMP                               PRPFIR    70
              TEMP = QCONS*QCONVA                                        PRPFIR    71
              TEFF = ((TZC(I,J) - T0)/TEMP)**2                           PRPFIR    72
              IF(TEFF .GT. TIME) TEFF = TIME                             PRPFIR    73
C                                                                        PRPFIR    74
C     HEAT TRANSFER CALCULATION USING SEMI-INFINITE HEAT CONDUCTION      PRPFIR    75
C     EQUATION WITH AN EFFECTIVE TIME                                    PRPFIR    76
              TZC(I,J) = TZC(I,J) + TEMP*                                PRPFIR    77
     $             (SQRT(TEFF + DELT) - SQRT(TEFF))                      PRPFIR    78
       IF(TBP(I,J) .GE. TIGNBP) TZC(I,J)=TIGN                            MOD1205    3
              IF(TZC(I,J) .LT. TIGN) GO TO 50                            PRPFIR    79
              FRACT(I) = 1.0                                             PRPFIR    80
              FIRE = .TRUE.                                              PRPFIR    81
              IF(IDEBUG(10)) WRITE(6,2000) TIME,I,J                      PRPFIR    82
C                                                                        PRPFIR    83
C                                                                        PRPFIR    84
C                                                                        PRPFIR    85
C     CALCULATE RISE IN TZR                                              PRPFIR    86
 50           CONTINUE                                                   PRPFIR    87
              IF(.NOT. ROW2) GO TO 95                                    PRPFIR    88
              IF(FRACT(I) .GE. 0.9999) GO TO 95                          PRPFIR    89
              IF(TZR(I) .GE. TIGN) GO TO 95                              PRPFIR    90
              UHOLE(I) = DOTMIG(I)/(0.7*RHOBG(I,1)*AH(I))                PRPFIR    91
C                                                                        PRPFIR    92
C     RESULTANT VELOCITY                                                 PRPFIR    93
          UGAS = UBG(I,J)                                                PRPFIR    94
          IF(I .EQ. 1) UGAS = UBG(I+1,J)/2.                              PRPFIR    95
          IF(I .EQ. NGX .AND. NX .EQ. 1) UGAS = UEG(I-1,J)/2. + UBG(I,J) PRPFIR    96
     *                                  /2.                              PRPFIR    97
          SQRTUH = SQRT(UGAS*UGAS + VBG(I,J)*VBG(I,J))                   PRPFIR    98
              RETXH = TEMP1*SQRTUH                                       PRPFIR    99
              TXNUSH = 0.3*RETXH**0.62 + TEMP2                           PRPFIR   100
```

```
                QCONVH = TXNUSH*TEMP3*(TTX - TZR(I))
                IF(QCONVH .LT. 0.001) GO TO 95
                QBAG(I,J) = QBAG(I,J) + QCONVH*ADV*(1.0 - FRACT(I))
                TEMP = QCONS*QCONVH
                TEFF = ((TZR(I) - TO)/TEMP)**2
                IF(TEFF .GT. TIME) TEFF = TIME
                TZR(I) = TZR(I) + TEMP*(SQRT(TEFF + DELT) - SQRT(TEFF))
                IF(TZR(I) .LT. TIGN) GO TO 95
                IF(IDEBUG(11)) WRITE(6,2001) TIME,I,J
   95        CONTINUE
           IF(TZC(1,J) .LT. TZC(2,J)) TZC(1,J) = TZC(2,J)
  100   CONTINUE
        IF(TZC(1,2).GE.TIGN)FRACT(1)=1.0
C
        RETURN
C
 2000 FORMAT(/,' TIME =',E14.8,' PROPELLANT AT GRID',I4,I4,' IS IGNITED'
     $  )
 2001 FORMAT(/,' TIME =',E14.8,' PROPELLANT UNDER INFLUENCE OF BELL TUBE
     $ HOLES IN GRID',I4,I4,' IS IGNITED')
        END
```

```
      SUBROUTINE REGRES                                                 REGRES     2
      COMMON/BURN/ATPB,CT,PEXP                                          REGRES     3
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN      4
      COMMON/HOLEA/RADHOL(85),NROWH ,NHOLES(85),XCL(85),AREAH(60),      HOLEA      2
     $  AH(60),FRACT(60)                                                HOLEA      3
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,QCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,MW,DM,TIGNOP,QBCONS,TOTM                        INPUTS     3
      COMMON/SPLINT/WHOLEC,WHOLEB                                       REGRES     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   DOTMIG(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),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
      LOGICAL WHOLEC,WHOLEB                                             REGRES    12
      LOGICAL ROW2,PART                                                 REGRES    13
      DATA PIDF/.785398/                                                REGRES    14
C                                                                       REGRES    15
C     SUBROUTINE REGRES CALCULATES UPDATED GRAIN DIMENSIONS AND POROSITY REGRES    16
C     DUE TO BURNING OF M6 PROPELLANT.  PHIBTD 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 DOTMB 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. TZR(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*TWODT                                               REGRES    36
C                                                                       REGRES    37
C     UPDATE GRAIN LENGTH.                                              REGRES    38
          XLTDT(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) .LE. 3.0*DI(I,J) ) GO TO 20                       REGRES    43
C                                                                       REGRES    44
C                                                                       REGRES    45
```

```
C     UPDATE OTHER DIMENSIONS                                          REGRES    46
            DOTDT(I,J) = DO(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 = PIDF*XL(I,J)*( DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J) )  REGRES    51
            VNEW = PIDF*XLTDT(I,J)*( DOTDT(I,J)*DOTDT(I,J) -           REGRES    52
     $                FN*DITDT(I,J)*DITDT(I,J) )                       REGRES    53
            GO TO 30                                                   REGRES    54
C                                                                      REGRES    55
C                                                                      REGRES    56
C     CALCULATE OLD AND NEW GRAIN VOLUMES,                             REGRES    57
C     AFTER THE GRAIN HAS SPLINTERED, VALUES FOR THE CROSS-SECTIONAL   REGRES    58
C     AREA OF THE PARTICLES GO INTO ARRAY DO AND VALUES FOR PERIMETER  REGRES    59
C     GO INTO ARRAY DI.  IF THE GRAIN HAS JUST SPLINTERED, AREA AND    REGRES    60
C     PERIMETER HAVE TO BE INITIALIZED.  IF THE GRAIN HAS JUST         REGRES    61
C     SPLINTERED DO(I,J) IS APPROXIMATELY 3.*DI(I,J) AND IF NOT        REGRES    62
C     DO(I,J) WILL BE LESS THAN DI(I,J).                               REGRES    63
 20         CONTINUE                                                   REGRES    64
            IF( DO(I,J) .LE. DI(I,J) ) GO TO 25                        REGRES    65
            WHOLEC = .FALSE.                                           REGRES    66
            AREA = PIDF*(DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J))         REGRES    67
            DI(I,J) = 3.14*(DO(I,J) + FN*DI(I,J))                      REGRES    68
            DO(I,J) = AREA                                             REGRES    69
C                                                                      REGRES    70
 25         DELR = BURNL*0.5                                           REGRES    71
            DOTDT(I,J) = DO(I,J) - DI(I,J)*DELR                        REGRES    72
C                                                                      REGRES    73
            IF(DOTDT(I,J) .GE. 1.0E-7) GO TO 27                        REGRES    74
            DOTDT(I,J) = 0.0                                           REGRES    75
            DITDT(I,J) = 0.0                                           REGRES    76
            XLTDT(I,J) = 0.0                                           REGRES    77
            PHIBTD(I,J) = 1.0                                          REGRES    78
         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) = SQRT( DI(I,J)*DI(I,J)/DO(I,J)*DOTDT(I,J) )    REGRES    84
C                                                                      REGRES    85
C     VOLUME IS LENGTH TIMES CROSS-SECTIONAL AREA                      REGRES    86
            VOLD = XL(I,J)*DO(I,J)                                     REGRES    87
            VNEW = XLTDT(I,J)*DOTDT(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 - FRACT(I))                      REGRES    97
            TEMP = PNDV*DELTAV                                         REGRES    98
C                                                                      REGRES    99
C     CALCULATE GAS MASS GENERATED BY BURNING PROPELLANT/GRID VOLUME   REGRES   100
            DOTMB(I,J) = TEMP*RHOP                                     REGRES   101
C                                                                      REGRES   102
```

```
C     UPDATE POROSITY
            PHIBTD(I,J) = PHIBG(I,J) + TEMP
            IF(PART) DOTOT(I,J) = DOTOT(I,J)*(1.0 - FRACT(I)) + DO(I,J)*
     $         FRACT(I)
            IF(PART) DITOT(I,J) = DITOT(I,J)*(1.0 - FRACT(I)) + DI(I,J)*
     $         FRACT(I)
            IF(PART) XLTOT(I,J) = XLTOT(I,J)*(1.0 - FRACT(I)) + XL(I,J)*
     $         FRACT(I)
            GO TO 48
C
 45         PHIBTD(I,J) = PHIBG(I,J)
C
 48     CONTINUE
 50     CONTINUE
C
        RETURN
        END
```

```
      SUBROUTINE PRPVEL                                                 MOD0925   164
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), MOD0925   165
     1  AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,    MOD0925   166
     2 AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,       MOD0925   167
     3 AREAM2,DIAMOT,HELLND,BELHEG,IPS1,IPS2,RADPS,BPIGN                MOD0925   168
      COMMON/CLOCK/TIME,DELT                                            MOD0925   169
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   MOD0925   170
     $  DX,DR,RX,GJ,TWODT,HBP                                           MOD0925   171
      COMMON/FORCE/PFORCE(60,5),PFORDT(60,5)                            MOD0925   172
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                       MOD0925   173
      COMMON/IGNTE/PLRAD,BPCHG,BRNOUT,BGBRN,TMPBG,                      MOD0925   174
     $ BEGTC(5),XLTC(5)                                                 MOD0925   175
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       MOD0925   176
     1 THICK(60),PENET,TYFCL                                            MOD0925   177
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP               MOD0925   178
      COMMON/PRO/LF1,LF2,MF1,MF2,BGJAM,BGSTP,NVSP,IBGNP                 MOD0925   179
      COMMON/PRMFLO/DOTMPF,UPRM,QPRM,PRFXP,FDSHR,CSPRKS,PRMCOF          MOD0925   180
      COMMON/BPT/TEP(2),DEP(2),CEP(2),FTOUGH,PDALST                     MOD0925   181
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,WCONS,RHOF,PHI0,TF,CA,RHO0,     MOD0925   182
     $  H0,P0,U0,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        MOD0925   183
      COMMON/TUBE/IENDTC(5),ITUBE,IBEGTC(5),INBP                        MOD1115     9
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),        MOD0925   184
     1  VBG(60,5), UPH(60,5), PCH(60,5), TZC(60,5),                     MOD0925   185
     2  DOTMIG(60), QBAG(60,5), XDRAG(60,5),DOTMB(60,5), UPBDT(60,5),   MOD0925   186
     3  PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),           MOD0925   187
     4  VBGTD(60,5), TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),      MOD0925   188
     5  PHIPTD(60,5),TZR(60),TBP(60,5)                                  MOD0925   189
      DIMENSION PPROP(60,5)                                             MOD0925   190
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        MOD0925   191
      LOGICAL PRI1,IDEBUG                                               MOD0925   192
      LOGICAL BGJAM,BGSTP,PENET                                         MOD0925   193
      DATA GRAV/32.16/                                                  MOD0925   194
      DATA IMOV/0/                                                      MOD1115    10
      DATA KBRN/0/                                                      MOD1115    11
      CALL CLEAR(PFORCE(1,1),PFORDT(60,5))                              MOD0925   195
      CALL CLEAR(PPROP(1,1),PPROP(60,5))                                MOD0925   196
C*                                                                      MOD0925   197
C     SUBROUTINE PRPVEL CALCULATES UPDATED PROPELLANT VELOCITY          MOD0925   198
C     (ASSUMING PROPELLANT IS FREE TO MOVE)                             MOD0925   199
C*                                                                      MOD0925   200
      DO 31 I=1,NGX                                                     MOD0925   201
      DO 31 J=1,NGR                                                     MOD0925   202
C*                                                                      MOD0925   203
C     PHIBTD VALUES SHOULD BE PUT INTO ARRAY PHIBG AND USED HERE        MOD0925   204
C*                                                                      MOD0925   205
       PHIAVE=(4.0-PHIBP(I,J)-PHIBP(I+1,J)-PHIBG(I,J)-PHIBG(I+1,J))/2.0 MOD0925   206
      IF(I.NE.1) PHIAVE=PHIAVE/2.0+(4.0-PHIBP(I,J)-PHIBP(I-1,J)         MOD0925   207
     *-PHIBG(I,J)-PHIBG(I-1,J))/4.0                                     MOD0925   208
      IF(PHIAVE.NE.0.0) XDRAG(I,J)=XDRAG(I,J)*                          MOD0925   209
     * (2.0-PHIBG(I,J)-PHIBP(I,J))/PHIAVE                               MOD0925   210
      PHIBG(I,J) = PHIBTD(I,J)                                          MOD0925   211
      PHIBP(I,J)=PHIPTD(I,J)                                            MOD0925   212
      PHIBG(I,J)=PHIBG(I,J)+PHIBP(I,J)-1.0                              MOD0925   213
C                                                                       MOD0925   214
      IF(PHIBG(I,J) .GE. 0.999) GO TO 31                                MOD0925   215
C*                                                                      MOD0925   216
C     CALCULATE TOTAL PRESSURE GRADIENT                                 MOD0925   217
```

```
C*                                                                      MOD0925  218
      IPA = IPATH(I,J)                                                  MOD0925  219
      GO TO (10,10,10,10,15,15,15,15,20,20,20,20),IPA                   MOD0925  220
   10 DPDX = (PCH(I-1,J) - PCH(I+1,J) )*0.5/DX                          MOD0925  221
      GO TO 25                                                          MOD0925  222
   15 DPDX = (PCH(I-1,J) - PCH(I,J))/DX                                 MOD0925  223
      GO TO 25                                                          MOD0925  224
   20 DPDX = (PCH(I,J) - PCH(I+1,J))/DX                                 MOD0925  225
   25 CONTINUE                                                          MOD0925  226
   30 PPROP(I,J) = XDRAG(I,J)+(DPDX-XDRAG(I,J))*(1.-PHIBG(I,J))         MOD0925  227
      PHIBG(I,J) = PHIBG(I,J) + 1.0 - PHIBP(I,J)                        MOD0925  228
C                                                                       MOD0925  229
C  THE CORRECT VALUES FOR ARRAY PHIBG MUST BE RECCVERED                 MOD0925  230
C  BECAUSE CHANGES ARE GOING TO BE MADE IN PHIBP,PHIBPPPP THAT IS       MOD0925  231
C                                                                       MOD0925  232
   31 CONTINUE                                                          MOD0925  233
  100 DO 900 J=1,NGR                                                    MOD0925  234
      I=1                                                               MOD0925  235
      ACCLL=0.                                                          MOD0925  236
      ACCLM = 0.                                                        MOD0925  237
      FLAG = 0.                                                         MOD0925  238
      LF1=0                                                             MOD0925  239
      LF2=0                                                             MOD0925  240
      MF1=0                                                             MOD0925  241
      MF2=0                                                             MOD0925  242
      SUMFCE = 0.                                                       MOD0925  243
      SUMASS = 0.                                                       MOD0925  244
      TOTFCE=0.                                                         MOD0925  245
      TOTMAS=0.                                                         MOD0925  246
      ICOUNT = 0                                                        MOD0925  247
      ICONT=0                                                           MOD0925  248
      FORCE=0.                                                          MOD0925  249
  101 CONTINUE                                                          MOD0925  250
      STDOFF=DELBEG-TEP(I)                                              MOD0925  253
      IF(ONED) GO TO 110                                                MOD1215    9
      IF(.NOT. PENET .AND. J .EQ. 1) GO TO 800                          MOD1215   10
      IF(J .GT. 1) GO TO 110                                            MOD1215   11
      PRMXP=2060.E6*(EXP(-.635*STDOFF/FTUBR))*TIME                      MOD1205    5
      IF(TIME .GE. .0004)PRMXP=1.33E6*(EXP(-.635*STDOFF/FTUBR-1200.     MOD1205    6
     1 *TIME))                                                          MOD0925  256
      PRMXP=PRMCOF*PRMXP                                                MOD0925  257
      BGFCE = PRMXP                                                     MOD1215   12
      FORCE=AREAAX*(BGFCE-PCH(IBGNP,J))                                 MOD0925  260
      WEGHT=0.                                                          MOD0925  261
      IF(FORCE .LT. 0.) FORCE=0.                                        MOD0925  262
      IF(BGJAM) GO TO 104                                               MOD0925  263
      KBRN = KBRN + 1                                                   MOD1115   21
      IF(KBRN .GT. 150) GO TO 103                                       MOD1115   22
  102 CONTINUE                                                          MOD0925  264
      IF(FORCE .GT. TYFCE) GO TO 103                                    MOD0925  265
      I = I+1                                                           MOD0925  266
      IF ( I .EQ. 5) GO TO 800                                          MOD0925  267
      FORCE=FORCE + PPROP(I,J)*DX*AREAAX                                MOD0925  268
      GO TO 102                                                         MOD0925  269
  103  WEGHT = (1-PHIBP(1,1))*BPDENS*DX/2.*AREAAX                       MOD0925  270
      BGJAM=.TRUE.                                                      MOD0925  271
      WRITE(6,1031) TIME                                                MOD0925  272
```

```
 1031 FORMAT(/,3X,6HTIME = E14.7, X,23HIGNITER BAG TIES FAILED)        MOD0925   273
  104 CONTINUE                                                          MOD0925   274
      DO 1032 I=1,NGX                                                   MOD0925   275
      IBGNP=I                                                           MOD0925   276
      IF(PHIBP(I,J) .LT. .999) GO TO 1033                               MOD0925   277
 1032 CONTINUE                                                          MOD0925   278
 1033 CONTINUE                                                          MOD0925   279
      I=IBGNP                                                           MOD0925   280
 1041 IF (PHIBP(I,J) .GE. .999) GO TO 1044                              MOD0925   281
      WEGHT = (1. - PHIBP(I,J))*BPDENS*DX*AREAAX+WEGHT                  MOD0925   282
      NEND = I                                                          MOD0925   283
      I = I + 1                                                         MOD0925   284
      GO TO 1041                                                        MOD0925   285
 1044 CONTINUE                                                          MOD0925   286
      IF(BGSTP) GO TO 1146                                              MOD0925   287
      IF(.NOT. BGJAM) 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
      IBGNP=IBGNP+1                                                     MOD0925   291
 1045 CONTINUE                                                          MOD0925   292
      LL1=IBGNP+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(BGSTP) NEND=INBP                                               MOD1111    19
      VOLBG=FLOAT(NEND-IBGNP+1)*AREAAX*DX                               MOD0925   295
      PHIPP=1.-WEGHT/(BPDENS*VOLBG)                                     MOD0925   296
      ELTCHG=VOLBG/AREAAX                                               MOD0925   297
      LBP=ELTCHG/DX                                                     MOD0925   298
      FRAT =ELTCHG/DX-FLOAT(LBP)                                        MOD0925   299
C                                                                       MOD1111    20
C   IBGNP IDENTIFIES THE FIRST GRID WITH POWDER AFTER IGNITER           MOD1111    21
C   PAD FAILURE.  LL1 IDENTIFIES THE FIRST PACKED GRID                  MOD1111    22
C                                                                       MOD0925   303
      IF(PHIPP .LT. 0.001) PHIPP=0.001                                  MOD1111    23
      PHIBP(IBGNP,J) = 1.-FRAT*(1.-PHIPP)                               MOD0925   305
      IF(BGSTP) GO TO 1047                                              MOD0925   306
      IF(NEND .LT.  INBP) GO TO 1047                                    MOD1111    24
      BGSTP=.TRUE.                                                      MOD0925   308
      WRITE(6,1046)                                                     MOD1111    25
 1046 FORMAT(5X,13HBGSTP IS TRUE)                                       MOD1111    26
      GO TO 109                                                         MOD0925   309
 1047 CONTINUE                                                          MOD0925   310
  105 CONTINUE                                                          MOD0925   311
      DO 1055 I=IBGNP,NEND                                              MOD1111    27
 1055 FORCE=FORCE + PPROP(I,J)*DX*AREAAX                                MOD0925   313
      IF(FORCE .GT. 350.) FORCE=350.                                    MOD0925   314
      IF(FORCE .LT. 10.) FORCE=10.                                      MOD0925   315
      SMLTC=XLTC(1)+XLTC(2)+XLTC(3)+XLTC(4)+XLTC(5)                     MOD1111    28
      ELT = SMLTC*(1.- .03145*((FORCE-10.)**.545))                      MOD1111    29
      VOLP=AREAAX*ELT                                                   MOD0925   317
      LLBP=ELT/DX                                                       MOD0925   318
      IF(ELT .LT. ELTCHG) GO TO 109                                     MOD0925   319
      IBGNP=NEND-LLBP                                                   MOD1111    30
      FRATN=LLT/DX-FLOAT(LLBP)                                          MOD0925   321
      LL1=IBGNP+1                                                       MOD0925   322
      PHIBP(IBGNP,J)=1.-WEGHT*FRATN/(VOLP*BPDENS)                       MOD0925   323
      PHIPP=1.-WEGHT/(VOLP*BPDENS)                                      MOD0925   324
      IF(PHIPP .LT. 0.001) PHIPP=0.001                                  MOD1111    31
  109 CONTINUE                                                          MOD0925   326
      PHIPTD(IBGNP,J)=PHIBP(IBGNP,J)                                    MOD0925   327
      DO 1090 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*PHIO) GO TO 534            MOD1215    13
      IF(PHIBG(I,J) .LE. .2*PHIO) 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-PHIO)*(PHIO/PHIBG(I-1,J)-1.0))/DX    MOD1215    16
      PFORCE(I+1,J) = (1.45E3/(1.0-PHIO)*(PHIO/PHIBG(I+1,J)-1.0))/DX    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
      FORCR = PPROP(I,J) + PFORCE(I-1,J) - PFORCE(I+1,J)                MOD1215    20
      IF(PHIBG(I,J) .GT. PHIO) FORCR = PPROP(I,J)                       MOD1215    21
  111 CONTINUE                                                          MOD1215    22
      GMASS=RHOP*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 = FORCR/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 600                                          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=PFROP(I,J)/GMASS                                             MOD1017   17
  211 DELUP=ACCL*DELT*GRAV                                              MOD1017   18
      UPBOT(I,J)=UPB(I,J)+DELUP                                         MOD1017   19
C                                                                       MOD1017   20
C   NOW TEST FOR SEPARATION OF PACKED GRILS INTO TWO GROUPS             MOD1017   21
C                                                                       MOD1017   22
      IF(I .EQ. LF1) GO TO 212                                          MOD1017   23
      IF(UPBOT(I,J) .GT. UPPO) GO TO 525                                MOD1017   24
  212 CONTINUE                                                          MOD1017   25
      SUMFCE = SUMFCE + PFROP(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
      UPPO = USUM/N + DELUP                                             MOD1017   34
      N = N + 1                                                         MOD1017   35
      IF(LF1 .EQ. 1 .AND. UPPO .LT. 0.) GO TO 480                       MOD1017   36
      GO TO 520                                                         MOD1017   37
  480 DO 500 I=LF1,LF2                                                  MOD1017   38
  500 UPBOT(I,J) = 0.                                                   MOD1017   39
      IF(LF2 .EQ. NGX) GO TO 800                                        MOD1017   40
  520 I=I+1                                                             MOD0925  417
      IF(PHIBG(I,J) .GT. 0.2*PHI0) FLAG=0.                              MOD1215   24
      IF(I .GT. NGX) GO TO 530                                          MOD0925  418
      IF(PHIBG(I,J) .GT. 0.2*PHI0) 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=I                                                             MOD0925  423
      GO TO 202                                                         MOD0925  424
  525 FLAG=1.                                                           MOD0925  425
  530 DO 533 I=LF1,LF2                                                  MOD0925  426
  533 UPBOT(I,J) =UPPO                                                  MOD1115   29
      IF(I .GT. NGX) GO TO 800                                          MOD0925  428
      IF(FLAG) 700,700,534                                              MOD0925  429
  534 MF1=I                                                             MOD0925  430
      M = 1                                                             MOD1017   41
      USUM = 0.                                                         MOD1017   42
  535 TOTFCE=TOTFCE + FPROP(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.-PHIBG(I,J)                                                MOD0925  435
      IF( FCPH .LT. .001)  FCPH = 0.                                    MOD1017   43
      TOTMAS=TOTMAS+RHOP*FFPH                                           MOD1002   11
      ACCLM = 0.                                                        MOD0925  438
      IF(TOTMAS .EQ. 0.) GO TO 541                                      MOD0925  439
      ACCLM = TOTFCE/TOTMAS                                             MOD0925  440
  541 DELUP=ACCLM*DELT*GRAV                                             MOD0925  441
      MF2 = I                                                           MOD1017   44
      USUM=UPB(I,J) + USUM                                              MOD1017   45
      UPPO = 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 UPBOT(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(PHIBG(I,J) .GT. 0.2*PHI0) GO TO 665                            MOD1215     26
      GO TO 535                                                         MOD0925    472
  665 DO 667 I=MF1,MF2                                                  MOD0925    473
  667 UPBOT(I,J) = UPPO                                                 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(PRI1) WRITE(6,731)LF1,LF2,J                                    MOD0925    480
  731 FORMAT(/,8X,4HLF1=I5,3X,4HLF2=I5,3X,2HJ=I3)                       MOD0925    481
      IF(PRI1) WRITE(6,733)MF1,MF2,J                                    MOD0925    482
  733 FORMAT(/,8X,4HMF1=I5,3X,4HMF2=I5,3X,2HJ=I3)                       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,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     2  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IUPGAP,AREAGP(60),DAVG,      CHAMIX     4
     3  AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/CLOCK/TIME,DELT                                            PROPEL     6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     1  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN      4
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOF,PHI0,TF,CA,RHO0,     INPUTS     2
     1  H0,P0,U0,GTRHOP,MW,DM,TIGNBP,QBCONS,TOTM                        INPUTS     3
      COMMON/SPLINT/WHOLEC,WHOLEB                                       PROPEL    10
      COMMON/PRO/LF1,LF2,MF1,MF2,BGJAM,BGSTP,NVSP,IBGNP                 MOD0925  487
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                       MOD0925  488
      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    DOTMIG(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),DDTMP(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
      LOGICAL WHOLEC,WHOLEB                                             PROPEL    13
      LOGICAL BPLEFT                                                    MOD0925  489
      LOGICAL PRI1,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,NGR                                                     PROPEL    21
      DO 10 I=1,NGX                                                     PROPEL    22
      PHIBG(I,J) = PHIBTD(I,J)                                          PROPEL    23
      XL(I,J) = XLTDT(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,NGR                                                     PROPEL    29
      DO 70 I=1,NGX                                                     PROPEL    30
      INCRE = 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)INCRE = -1                                   PROPEL    34
      IF(UPBDT(I,J).EQ.0.0)INCRE = 0                                    PROPEL    35
      ITWO = I + 2 * INCRE                                              PROPEL    36
      IONE = I + INCRE                                                  PROPEL    37
      UP1 = UPBDT(I,J)                                                  PROPEL    38
      IF (I .EQ. 1 ) GO TO 15                                           PROPEL    39
      IF ( J .EQ. 2 ) GO TO 12                                          PROPEL    40
      AREAX = AREAC (I)                                                 PROPEL    41
      ARONE = AREAC (IONE)                                              PROPEL      42
      ARTWO = AREAC (ITWO)                                              PROPEL      43
      AREAP1 = AREAC (I+1)                                              PROPEL      44
      AREAM1 = AREAC (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
      AREAM1 = AREAR (I-1)                                              PROPEL      51
      GO TO 18                                                          PROPEL      52
   15 IF (J .EQ. 2 ) GO TO 16                                           PROPEL      53
      AREAX = AREAC (I)                                                 PROPEL      54
      ARONE = AREAC (IONE)                                              PROPEL      55
      ARTWO = AREAC (ITWO)                                              PROPEL      56
      AREAP1 = AREAC (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
      DMPI1 = (1.0 - PHIBG(IONE,J))*DELT*ARONE*ABS(UPBDT(IONE,J))       PROPEL      65
      DMPI = ( 1.0 - PHIBG (I,J))*DELT*AREAX*ABS (UPBDT(I,J))           PROPEL      66
      DMPI2 = 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) DMPI2=(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
      PHII1 = PHIBG (IONE,J) + (DMPI1 - DMPI2 - DMPI) / (ARONE*DX)      PROPEL      73
      GO TO 30                                                          PROPEL      74
   20 PHII1 = PHIBG(IONE,J) + 2.0*(DMPI1 - DMPI)/ (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
      DMPIP1 =0.0                                                       PROPEL      84
      DMPI = ( 1.0 - PHIBG(I,J))*DELT*AREAX*ABS ( UPBDT(I,J))           PROPEL      85
      DMPIM1 = 0.0                                                      PROPEL      86
      XLU = XL(I+1,J)                                                   PROPEL      87
      XLL = XL(I-1,J)                                                   PROPEL      88
      DDU = DD(I+1,J)                                                   PROPEL      89
      DDL = DD(I-1,J)                                                   PROPEL      90
      DIU = DI(I+1,J)                                                   PROPEL      91
      DIL = DI(I-1,J)                                                   PROPEL      92
      UPU = UPB(I+1,J)                                                  PROPEL      93
      UPL = UPB(I-1,J)                                                  PROPEL      94
      IZCU=IZC(I+1,J)                                                   PROPEL      95
      IZCL=IZC(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) DMPIP1 = (1.0 - PHIBG(I+1,J))           PROPEL     100
     1   *DELT*AREAP1*ABS(UPBDT(I+1,J))                                 PROPEL     101
      IF(UPBDT(I-1,J) .GT. 0.0) DMPIM1 = (1.0 - PHIBG(I-1,J))           PROPEL     102
     1   *DELT*AREAM1*ABS(UPBDT(I-1,J))                                 PROPEL     103
      GO TO 50                                                          PROPEL     104
  40  IF(I .EQ. 1  .AND.  UPBDT(2,J) .LT. 0.0) DMPIP1 = 2.0*            PROPEL     105
     1  (1.0 - PHIBG(2,J))*DELT*AREAP1*ABS(UPBDT(2,J))                  PROPEL     106
      DMPI = 2.0 * DMPI                                                 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(I .EQ. NGX .AND. NX .EQ. 1 .AND. UPBDT(I,J) .GT. 0.)           MOD1215     29
     *    DMPI = 0.                                                     MOD1215     30
      IF(I .EQ. NGX .AND. NX .EQ. 1) DMPIP1 = 0.                        MOD1215     31
      PHII = PHIBG(I,J) + (DMPI-DMPIP1-DMPIM1)/(AREAX*DX)               MOD1215     32
      IF(I.NE.1) GO TO 62                                               PROPEL     111
      XLL = XLU                                                         PROPEL     112
      DOL = DOU                                                         PROPEL     113
      DIL = DIU                                                         PROPEL     114
      UPL = 0.0                                                         PROPEL     115
      TZCL=TZCU                                                         PROPEL     116
   62 IF(PHII .GT. .99999) GO TO 63                                     MOD1205      8
      UPBDT(I,J) = ( UPBDT(I,J)*(1.0 - PHIBG(I,J))*AREAX*DX             PROPEL     118
     1          + UPU*DMPIP1 + UPL*DMPIM1 - UPBDT(I,J)*DMPI )/          PROPEL     119
     2          ((1.0 - PHII)*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
      XLTDT(I,J) = (XL(I,J)*(1.0 - PHIBG(I,J))*AREAX*DX    + XLU        PROPEL     121
     1    *DMPIP1 + XLL*DMPIM1 - XL(I,J)*DMPI) / ((1.0 - PHII)*AREAX*DX)  PROPEL     122
      IF(WHOLEC) DOTDT(I,J) = (DO(I,J)*(1.0-PHIBG(I,J))*AREAX*DX + DOU  PROPEL     123
     1    *DMPIP1 + DOL*DMPIM1 - DO(I,J)*DMPI) / ((1.0 - PHII)*AREAX*DX)  PROPEL     124
      IF(WHOLEC) DITDT(I,J) = (DI(I,J)*(1.0-PHIBG(I,J))*AREAX*DX + DIU  PROPEL     125
     1   *DMPIP1 + DIL*DMPIM1 - DI(I,J)*DMPI) / ((1.0 - PHII)*AREAX*DX)   PROPEL     126
      IF(TZC(I,J) .LT. 0.001) TZC(I,J) = TZCL                           PROPEL     127
      IF(TZC(I,J).LT.TIGN) TZC(I,J)=(TZC(I,J)*(1.0-PHIBG(I,J))          PROPEL     128
     .                         *AREAX*DX + TZCU*DMPIP1 + TZCL*DMPIM1    PROPEL     129
     *                                -TZC(I,J)*DMPI)/((1.0-PHII)*AREAX*DX)  PROPEL     130
   63 CONTINUE                                                          MOD0620      1
      PHIBTD(I,J)=PHII                                                  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
JLD HAVE RESULTED IN BETTER OPTIMIZATION
```

```
      SUBROUTINE DRAG(DRAGX,INB,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 UTDT AND VTDT.                 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/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL      2
     1  PG(100), TG(100),  PMDOT(100), GL(100), UDRAG(100), FRICT(100), BARRL      3
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),BARRL      4
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),     BARRL      5
     4  UAENER(100)                                                     BARRL      6
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           DRAG      10
     $  UTDT,VBGAVE,VTDT                                                DRAG      11
      COMMON/CLOCK/TIME,DELT                                            DRAG      12
      COMMON/DRGCON/VTSG,FRCTF                                          MOD1215   33
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTD(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,            GRAIN      3
     2  XLB(100), DOB(100), DIB(100), UXLB(100), UDOB(100), UDIB(100)   GRAIN      4
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENER,BGENER,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), TZC(60,5),                    BAG        3
     2   DOTMIG(60), DBAG(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),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),IZR(60),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)*DO(I,J)*XL(I,J))                                  DRAG      29
      IF(DO(I,J).LT.DI(I,J)) 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)*BPRAD(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
      VTSGG=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.DIB(I)) 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
      VTSGG=32.16*DM                                                    DRAG      46
    7 CONTINUE                                                          DRAG      47
```

```
      IF (VTSGG.GT.VTSG)VTSGG=VTSG
      IF(PHIAVE .LT. 0.1) PHIAVE=0.1
      SHAPE=0.7
      IF(J .EQ. 1 .OR. DO(I,J) .LE. 0. .OR. XL(I,J) .LE. 0.) GO TO 9
      DEFF =(1.5*DO(I,J)*DO(I,J)*XL(I,J))**0.333
      SHAPE=DEFF*DEFF*(DO(I,J)*(XL(I,J)+.5*DO(I,J)))
      IF(SHAPE .LT. 0.1) SHAPE = 0.1
    9 CONTINUE
      CONST = 2.0*FRCTF*(1.-PHIAVE)/(SHAPE*PHIAVE*VTSG)
C
C
      A = CONST*RHOTOT
      B = 2.0*PHIRHO/(DELT*GRAV)
C
C     THE FOLLOWING CALCULATIONS INCORPORATE THE SIMULTANEOUS SOLUTIONS
C     OF THE DRAG EQUATION AND FINITE DIFFERENCE EQUATIONS FOR UBG
C     TO CALCULATE THE UPDATED QUANTITY FOR UBG EXPLICITLY,
C     CALCULATIONS ARE DONE IN THE FOLLOWING SEQUENCE ( DRXT AND
C     DRXTDT ARE THE CURRENT AND UPDATED DRAG IN THE AXIAL DIRECTION)
C               1)  COMPUTE DRXT
C               2)  COMPUTE UTDT - UPBAVE, (UMU), WHICH IS THE
C                   SOLUTION OF A QUADRATIC AND THEN GET UTDT
C               3)  COMPUTE DRXTDT
C               4)  COMPUTE DRAGX
C
      DIFFU = UBGAVE - UPBAVE
      IF(ABS(DIFFU).GT..0001)GO TO 10
      DRAGX = 0.0
      GO TO 40
C
   10 DRXT = CONST*RHOAVE*DIFFU*ABS(DIFFU)
      C = DRXT - B*(UTDT-UPBAVE)
      DISCRM = B*B - 4.0*A*C
      IF(DISCRM.LT.0.0)GO TO 20
      UMU = (-B + SQRT(DISCRM))/(A+A)
      GO TO 30
C
   20 DISCRP = B*B + 4.0*A*C
      UMU = (B - SQRT(DISCRP))/(A+A)
C
   30 UTDT = UMU + UPBAVE
      DRXTDT = A*UMU*ABS(UMU)
      DRAGX = (DRXT + DRXTDT)*0.5
C
C
   40 IF(INB) RETURN
C
C    UPDATE VTDT.  DRAGR DOES NOT NEED TO BE CALCULATED SINCE IT IS
C    NOT USED EXPLICITLY IN THE FINITE DIFFERENCE EQUATIONS.
      IF( ABS(VBGAVE) .LE. 0.0001) RETURN
C
      DRRT = CONST*RHOAVE*VBGAVE*ABS(VBGAVE)
      C = DRRT - B*VTDT
      DISCRM = B*B - 4.0*A*C
      IF(DISCRM.LT.0.0)GO TO 60
      VTDT = (-B + SQRT(DISCRM))/(A+A)
      RETURN
```

```
C                                                                       DRAG      98
 60   DISCRP = B*B + 4.0*A*C                                            DRAG      99
      VTOT = (B - SQRT(DISCRP))/(A+A)                                   DRAG     100
      RETURN                                                            DRAG     101
C                                                                       DRAG     102
      END                                                               DRAG     103
      SUBROUTINE GSPROP(RO,RRO,R,CVO,CVH,CV,P,H,T,RHO,U,V,GAM,CP,IPROP)  GSPROP       2
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,     EQNS         2
     $  DX,DR,NX,GJ,TWODT,HBP                                             EQNS         3
      DATA XJUL/778.0/                                                   GSPROP       4
C                                                                        GSPROP       5
C     IPROP = 1 - GIVEN T AND P                                          GSPROP       6
C     IPROP = 2 - GIVEN H AND RHO                                        GSPROP       7
C     IPROP = 3 - GIVEN H AND P                                          GSPROP       8
C                                                                        GSPROP       9
C     R, CV,CP, AND GAM ARE TO BE CALCULATED.  ALSO, OF T, P, H, AND RHO GSPROP      10
C     THE TWO THAT ARE NOT GIVEN ARE TO BE CALCULATED.                   GSPROP      11
C                                                                        GSPROP      12
      GO TO (10,20,30),IPROP                                             GSPROP      13
C                                                                        GSPROP      14
C     THE EQUATIONS FOR IPROP=1 ARE VALID ASSUMING GAS VELOCITY IS 0.    GSPROP      15
   10 R = 0.5*RO + SQRT(0.25*RO*RO + RRO*P/T)                            GSPROP      16
      XTEMP = T*CVH                                                      GSPROP      17
      CV = (CVO + XTEMP*R/XJUL)/(1.0 - XTEMP)                            GSPROP      18
      CP = R/XJUL + CV                                                   GSPROP      19
      GAM = CP/CV                                                        GSPROP      20
      H = T*CP                                                           GSPROP      21
      RHO=P/(R*T)                                                        GSPROP      22
      RETURN                                                             GSPROP      23
C                                                                        GSPROP      24
   20 IF(RHO .LE. 0.) RHO = .00001                                       MOD0925    501
      R=RO + RRO*RHO                                                     MOD0925    502
      IF(H .LE. 0.) H=.1                                                 MOD0925    503
      CV = CVO + CVH*H                                                   GSPROP      26
      CP = R/XJUL + CV                                                   GSPROP      27
      GAM = CP/CV                                                        GSPROP      28
      T = H/CP                                                           GSPROP      29
      P = XJUL*(GAM - 1.0)/GAM*RHO*(H - (U*U+V*V)/TWOGJ)                 GSPROP      30
      IF(P .LE. 0.1) P=0.1                                               MOD0925    504
      RETURN                                                             GSPROP      31
C                                                                        GSPROP      32
   30 IF(H .LE. 0.) H=.1                                                 MOD0925    505
      CV=CVO + CVH*H                                                     MOD0925    506
      XTEMP = RRO*P/(H - (U*U+V*V)/TWOGJ)                                GSPROP      34
      YTEMP = 0.5*(RO + XTEMP/XJUL)                                      GSPROP      35
      R = YTEMP + SQRT(YTEMP*YTEMP + XTEMP*CV)                           GSPROP      36
      CP = CV + R/XJUL                                                   GSPROP      37
      GAM = CP/CV                                                        GSPROP      38
      T = H/CP                                                           GSPROP      39
      RHO=P/(R*T)                                                        GSPROP      40
      RETURN                                                             GSPROP      41
      END                                                                GSPROP      42
      SUBROUTINE AXIS                                                   AXIS         2
C     IX IS I HERE, IR IS J                                             AXIS         3
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBEAVE,UPBAVE,           AXIS         4
     $  UTDT,VBGAVE,VTDT                                                AXIS         5
      COMMON/FAILED/THICKT,PHOOP,PCOMP,ETUB,XNTUB,FAIL,MFAIL(60),       MOD0925    507
     1 THICK(60),PENET,TYFCE                                            MOD0925    508
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI        2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI        3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMI        4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI        5
      COMMON/CLOCK/TIME,DELT                                            AXIS         7
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS         3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      AXIS         9
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,CCONS,RHOF,PHIO,TF,CA,RHOO,     MOD0925    509
     $  HO,PO,UO,GTRHOP,HW DM,TIGNBP,QBCONS,TOTM                        MOD0925    510
      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   DOTMIG(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),DCTMP(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
      LOGICAL PENET                                                     MOD0925    511
      DATA GRAV/32.16/                                                  MOD0925    512
C                                                                       AXIS        13
      RB = 0.0                                                          AXIS        14
      XB=XB+DX                                                          AXIS        15
C                                                                       AXIS        16
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           AXIS        17
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      AXIS        18
      BUGGER = (GAM - 1.0)/TWOGJ                                        AXIS        19
      IP1 = I+1                                                         AXIS        20
      IM1 = I -1                                                        AXIS        21
C                                                                       AXIS        22
C                                                                       AXIS        23
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST  AXIS        24
C     POROSITY OF THE PROPELLANT.                                       AXIS        25
      F5 = PHIBG(I,J)                                                   AXIS        26
      A5 = AREAC(I)                                                     AXIS        27
      G2 = RHOBG(IM1,J)                                                 AXIS        28
      G4 = RHOBG(IP1,J)                                                 AXIS        29
      G5 = RHOBG(I,J)                                                   AXIS        30
      E2 = G2*UBG(IM1,J)                                                AXIS        31
      E4 = G4*UBG(IP1,J)                                                AXIS        32
      E5 = G5*UBG(I,J)                                                  AXIS        33
      H2 = PHIBG(IM1,J) * AREAC(IM1) * E2                               AXIS        34
      H4 = PHIBG(IP1,J) * AREAC(IP1) * E4                               AXIS        35
      C2 = G2 * HBG(IM1,J)                                              AXIS        36
      C4 = G4 * HBG(IP1,J)                                              AXIS        37
      C5 = G5 * HBG(I,J)                                                AXIS        38
      B2 = BUGGER* UBG(IM1,J) * E2                                      AXIS        39
      B4 = BUGGER * UBG(IP1,J) * E4                                     AXIS        40
      B5 = BUGGER * UBG(I,J) * E5                                       AXIS        41
C                                                                       AXIS        42
      PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25             AXIS        43
      RHOAVE = (G2 + G5 + G5 + G4)*0.25                                 AXIS        44
      UBGAVE = (UBG(IM1,J) + UBG(I,J) + UBG(I,J) + UBG(IP1,J))*0.25    AXIS      45
      VBGAVE = 0.0                                                     AXIS      46
      UPBAVE = (UPB(IM1,J) + UPB(I,J) + UPB(I,J) + UPB(IP1,J))*0.25    AXIS      47
C                                                                      AXIS      48
      PHITDT = PHIBTD(I,J) + PHIPTD(I,J) - 1.0                         AXIS      49
      RHOTDT = ( F5*RHOAVE - T2DX*(H4 - H2)/A5 + DELT*DOTMIG(I)/DVAXIS AXIS      50
     $        + DOTMB(I,J) + DOTMP(I,J) )/PHITDT                       AXIS      51
      PHIRHO = PHITDT*RHOTDT                                           AXIS      52
      TERM=0.0                                                         AXIS      53
      IF(DOTMIG(I).LT.0.0)TERM=1.0                                     AXIS      54
C                                                                      AXIS      55
      UTDT = ( F5*(E2 + E5 + E5 + E4)*0.25                             AXIS      56
     1        - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J)                    AXIS      57
     2           + GRAV*AREAC(I)*PCH(IP1,J)                            AXIS      58
     3           - GRAV*AREAC(I)*PCH(IP1,J))/A5                        AXIS      59
     *  +DELT*DOTMIG(I)*UBG(I,J)/DVAXIS*TERM                           AXIS      60
     4        + DOTMB(I,J)*UPB(I,J) )/PHIRHO                           AXIS      61
      IF (  ABS(UTDT).LT. 0.1  )UTDT = 0.0                             AXIS      62
C                                                                      AXIS      63
      VTDT = 0.0                                                       AXIS      64
C                                                                      AXIS      65
      IF(PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)             AXIS      66
C                                                                      AXIS      67
      HIGN = HBG(I,1)                                                  AXIS      68
      IF ( DOTMIG(I) .GT. 0.00001 ) HIGN = HBG(I,2)                    AXIS      69
C                                                                      AXIS      70
      HBGTD(I,J) = GAM*( F5*(C2+B2+C5+C5+H5+B5+C4+B4)/(4.0*GAM)         AXIS      71
     1        - T2DX*(H4*HBG(IP1,J) - H2*HBG(IM1,J))/A5                AXIS      72
     2        + DELT*(DOTMIG(I)/DVAXIS*HIGN - QBAG(I,J))               AXIS      73
     3        + DOTMB(I,J)*HMB + DOTMP(I,J)*HBP                        AXIS      74
     * -XDRAG(I,J)*UPB(I,J)*DELT/778.                                  AXIS      75
     4        - BUGGER*PHIRHO*UTDT*UTDT/GAM )/PHIRHO                   AXIS      76
C                                                                      AXIS      77
      RHOBTD(I,J) = RHOTDT                                             AXIS      78
      UBGTD(I,J) = UTDT                                                AXIS      79
      VBGTD(I,J) = VTDT                                                AXIS      80
      RETURN                                                           AXIS      81
      END                                                              AXIS      82
      SUBROUTINE AXIT                                                   AXIT          2
C     IX IS I HERE, IR IS J                                             AXIT          3
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHCAVE,UBGAVE,UPBAVE,           AXIT          4
     $  UTDT,VBGAVE,VTDT                                                AXIT          5
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI         2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI         3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IDPGAP,AREAGP(60),DAVG,      CHAMI         4
     $  AREAH2,DIAMBT,BELENG,BELBFG,IPS1,IPS2,RADFS,BPIGN               CHAMI         5
      COMMON/CLOCK/TIME,DELT                                            AXIT          7
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS          2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS          3
      COMMON/GASCON/RG,RRO,CVG,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), TZC(60,5),                    BAG           3
     2   DOTMIG(60), UBAG(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),COTMP(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/                                                  AXIT         12
C                                                                       AXIT         13
      RB = RB + DR                                                      AXIT         14
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           AXIT         15
     $        RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                    AXIT         16
      BUGGER = (GAM - 1.0)/TWOGJ                                        AXIT         17
      IP1 = I+1                                                         AXIT         18
      IM1 = I -1                                                        AXIT         19
      JP1 = J+1                                                         AXIT         20
      F1 = PHIBG(I,JP1)                                                 AXIT         21
      F2 = PHIBG(IM1,J)                                                 AXIT         22
      F4 = PHIBG(IP1,J)                                                 AXIT         23
      F5 = PHIBG(I,J)                                                   AXIT         24
      G1 = F1*RHOBG(I,JP1)                                              AXIT         25
      G2 = F2*RHOBG(IM1,J)                                              AXIT         26
      G4 = F4*RHOBG(IP1,J)                                              AXIT         27
      G5 = F5*RHOBG(I,J)                                                AXIT         28
      E1 = G1*UBG(I,JP1)                                                AXIT         29
      E2 = G2*UBG(IM1,J)                                                AXIT         30
      E4 = G4*UBG(IP1,J)                                                AXIT         31
      E5 = G5*UBG(I,J)                                                  AXIT         32
      D1 = G1*VBG(I,JP1)                                                AXIT         33
      C1 = G1*HBG(I,JP1)                                                AXIT         34
      C2 = G2*HBG(IM1,J)                                                AXIT         35
      C4 = G4*HBG(IP1,J)                                                AXIT         36
      C5 = G5*HBG(I,J)                                                  AXIT         37
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    AXIT         38
      B2 = BUGGER*( UBG(IM1,J)**2 + VBG(IM1,J)**2 )*G2                  AXIT         39
      B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2 )*G4                   AXIT         40
      B5 = BUGGER*(UBG(I,J)**2 + VBG(I,J)**2)*G5                        AXIT         41
C*                                                                      AXIT         42
      DENOM = F1 + F2 + F4 + F5                                         AXIT         43
      SIGG = G1 + G2 + G4 + G5                                          AXIT         44
      SIGE = E1 + E2 + E4 + E5                                          AXIT         45
      PHIAVE = DENOM*0.25                                               AXIT         46
      RHOAVE = SIGG/DENOM                                               AXIT         47
      UBGAVE = SIGE/SIGG                                                AXIT         48
      VBGAVE = 0.0                                                      AXIT         49
      UPBAVE = (UPB(I,J) + UPB(I,JP1) + UPB(IM1,J) + UPB(IP1,J))*0.25
C
      PHITDT = PHIBTD(I,J)
      TERM=0.0
      IF(DOTMIG(I).GT.0.0)TERM=-1.0
      RHOTDT = ( F5*RHOAVE - T2DX*(E4-E2) - DTDR*C1
     $        - DELT*DOTMIG(I)/DVAXIT + DOTMB(I,J)   )/PHITDT
      PHIRHO = PHITDT*RHOTDT
      UTDT = (F5*SIGE/DENOM - T2DX*( E4*UBG(IP1,J) - E2*UBG(IM1,J)
     $              + GRAV   *PCH(IP1,J) - GRAV   *PCH(IM1,J) )
     *  +DELT*DOTMIG(I)*UBG(I,J)/DVAXIS*TERM
     $        - DTDR*D1*UBG(I,JP1) - DOTMB(I,J)*UPE(I,J) )/PHIRHO
      IF( ABS(UTDT) .LT. 0.1 ) UTDT = 0.0
C*
C*
      VTDT = 0.0
      IF(PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)
C*
      HIGN = HBG(I,1)
C*
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)
      HBGTD(I,J) = GAM*( F5*(C1+B1+C2+B2+C4+B4+C5+B5)/(DENOM*GAM)
     $        -T2DX*( C4*UBG(IP1,J) - C2*UBG(IM1,J) )
     $        - DTDR*C1*VBG(I,JP1)
     $        - DELT*( DOTMIG(I)/DVAXIT*HIGN + QBAE(I,J) )
     *  -XDRAG(I,J)*UPB(I,J)*DELT/788.
     $        + DOTMB(I,J)*HMB - BUGGER*UTDT*UTDT*PHIRHO/GAM )/PHIRHO
C*
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE INTER                                                  INTER      2
      COMMON/AVGDT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           INTER      3
     $ UTOT,VBGAVE,VTOT                                                 INTER      4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREMAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAH2,DIAPB1,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            INTER      6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTOR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/RO,RRO,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), TZC(60,5),                    BAG        3
     2   DOTMIG(60), QDAG(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),DOTMP(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 GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           INTER     14
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     INTER     15
      BUGGER = (GAM - 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
      D1 = G1*VBG(I,JP1)                                                INTER     34
      D2 = G2*VBG(IM1,J)                                                INTER     35
      D3 = G3*VBG(I,JM1)                                                INTER     36
      D4 = 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
      DRP = (RB + DR)/RB                                                INTER     46
      DRM = (RB - DR)/RB                                                INTER     47
C*                                                                      INTER     48
      DENOM = F1 + F2 + F3 + F4                                         INTER     49
      SIGG = G1 + G2 + G3 + G4
      SIGD = D1 + D2 + D3 + D4
      SIGE = E1 + E2 + E3 + E4
      PHIAVE = DENOM*0.25
      RHOAVE = SIGG/DENOM
      UBGAVE = SIGE/SIGG
      VBGAVE = SIGD/SIGG
      UPBAVE = (UPB(I,JP1) + UPB(IM1,J) + UPB(I,JM1) + UPB(IP1,J))*0.25
      AVPHI = F5/DENOM
C*
      PHITOT = PHIBTD(I,J)
      RHOTOT = ( F5*RHOAVE   -  T2DX*(E4 - E2)
     $                       -  T2DR*( DRP*D1 - DRM*D3 )
     $       + DOTMB(I,J)   )/PHITOT
      PHIRHO = PHITOT*RHOTOT
      UTOT = ( AVPHI*SIGE      -  T2DX*(E4*UBG(IP1,J) - E2*UBG(IM1,J)
     $            + GRAV   *PCH(IP1,J) - GRAV   *FCH(IM1,J) )
     $   -  T2DR*( DRP*E1*VBG(I,JP1) - DRM*E3*VBG(I,JM1) )
     $       + DOTMB(I,J)*UPB(I,J) )/PHIRHO
      IF( ABS(UTOT) .LT. 0.1 ) UTOT = 0.0
C*
      VTOT = ( AVPHI*SIGD
     $       - T2DX*( E4*VBG(IP1,J) - E2*VBG(IM1,J) )
     $      - T2DR*( DRP*( D1*VBG(I,JP1) + GRAV   *PCH(I,JP1) )
     $               - DRM*( D3*VBG(I,JM1) + GRAV   *PCH(I,JM1) ) )
     $      + DELT*GRAV*PCH(I,J)/RB    )/PHIRHO
      IF( ABS(VTOT) .LT. 0.1 ) VTOT = 0.0
C*
C     ARRAY XDRAG IS CLEARED IN MAIN AT THE BEGINNING OF EACH TIME
C     INTERVAL SO IF SUBROUTINE DRAG IS NOT CALLED, XDRAG(I,J) IS 0.0.
      IF( PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J), .FALSE.)
C*
C*
      HBGTD(I,J) = GAM*(AVPHI*(C1+B1+C2+B2+C3+B3+C4+B4)/GAM
     $     -  T2DX*( C4*UBG(IP1,J) - C2*UBG(IM1,J))
     $     -  T2DR*( DRP*C1*VBG(I,JP1) - DRM*C3*VEG(I,JM1) )
     $       - DELT*XDRAG(I,J) + DOTMB(I,J)*HMB
     $       - BUGGER*(UTOT*UTOT + VTOT*VTOT)*PHIRHO/GAM ) / PHIRHO
C*
C
      RHOBTD(I,J) = RHOTOT
      UBGTD(I,J) = UTOT
      VBGTD(I,J) = VTOT
      RETURN
      END
      SUBROUTINE BNDY                                                   BNDY       2
      COMMON/AVGOT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBEAVE,UPBAVE,           BNDY       3
     $ UTOT,VBGAVE,VTOT                                                 BNDY       4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAMX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAR2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            BNDY       6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GO,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      BNDY       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   DOTMIG(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),CCTMP(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/                                                  BNDY      11
C                                                                       BNDY      12
      RB=RB+DR                                                          BNDY      13
C                                                                       BNDY      14
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           BNDY      15
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     BNDY      16
      BUGGER = (GAM - 1.0)/TWOGJ                                        BNDY      17
      IP1 = I+1                                                         BNDY      18
      IM1 = I -1                                                        BNDY      19
      JM1 = J-1                                                         BNDY      20
      F2 = PHIBG(IM1,J)                                                 BNDY      21
      F3 = PHIBG(I,JM1)                                                 BNDY      22
      F4 = PHIBG(IP1,J)                                                 BNDY      23
      F5 = PHIBG(I,J)                                                   BNDY      24
      G2 = F2*RHOBG(IM1,J)                                              BNDY      25
      G3 = F3*RHOBG(I,JM1)                                              BNDY      26
      G4 = F4*RHOBG(IP1,J)                                              BNDY      27
      E2 = G2*UBG(IM1,J)                                                BNDY      28
      E3 = G3*UBG(I,JM1)                                                BNDY      29
      E4 = G4*UBG(IP1,J)                                                BNDY      30
      D3 = G3*VBG(I,JM1)                                                BNDY      31
      C2 = G2*HBG(IM1,J)                                                BNDY      32
      C3 = G3*HBG(I,JM1)                                                BNDY      33
      C4 = G4*HBG(IP1,J)                                                BNDY      34
      B2 = BUGGER*UBG(IM1,J)**2*G2                                      BNDY      35
      B3 = BUGGER*( UBG(I,JM1)**2 + VBG(I,JM1)**2)*G3                   BNDY      36
      B4 = BUGGER*UBG(IP1,J)**2*G4                                      BNDY      37
C*                                                                      BNDY      38
      DENOM = F2 + F3 + F3 + F4                                         BNDY      39
      SIGG = G2 + G3 + G3 + G4                                          BNDY      40
      SIGE = E2 + E3 + E3 + E4                                          BNDY      41
      PHIAVE = DENOM*0.25                                               BNDY      42
      RHOAVE = SIGG/DENOM                                               BNDY      43
      UBGAVE = SIGE/SIGG                                                BNDY      44
      VBGAVE = 0.0                                                      BNDY      45
      UPBAVE = (UPB(IM1,J) + 2.0*UPB(I,JM1) + UPB(IP1,J))*0.25          BNDY      46
C*                                                                      BNDY      47
      PHITOT = PHIBTD(I,J)                                              BNDY      48
      RHOTOT = ( F5*RHOAVE - T2DX*(E4-E2) + DTDR*C3                     BNDY      49
     $        + DOTMB(I,J)   )/PHITOT
      PHIRHO  = PHITOT*RHOTOT
C*
      UTOT = (F5*SIGE/DENOM - T2DX*(E4*UBG(IP1,J) - E2*UBG(IM1,J)
     $             + GRAV   *FCH(IP1,J) - GRAV   *FCH(IM1,J) )
     $        +DTDR*E3*VBG(I,JM1)
     $        + DOTMB(I,J)*UPB(I,J) )/PHIRHO
      IF( ABS(UTOT) .LT. 0.1 ) UTOT = 0.0
      VTOT=0.0
C*
      IF( PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J), .FALSE.)
C*
C*
      HBGTD(I,J) = GAM*( F5*(C2+B2+C3+C3+B3+B3+C4+B4)/(DENOM*GAM)
     $        -T2DX*(C4*UBG(IP1,J) - C2*UBG(IM1,J))
     $        + DTDR*C3*VBG(I,JM1)
     $        - DELT*QBAG(I,J) + DOTMB(I,J)*HMB
     $        - BUGGER*UTOT*UTOT*PHIRHO/GAM )/PHIRHO
C*
C
      RHOBTD(I,J) = RHOTOT
      UBGTD(I,J) = UTOT
      VBGTD(I,J) = VTOT
      RETURN
      END
      SUBROUTINE BSURFA                                                 BSURFA     2
C     IX IS I HERE, IR IS J                                             BSURFA     3
      COMMON/AVGOT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           BSURFA     4
     $  UTOT,VBGAVE,VTOT                                                BSURFA     5
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAM2,DIAMBT,DELEND,DELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            BSURFA     7
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/R0,RR0,CV0,CVH                                      BSURFA     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), IZC(60,5),                    BAG        3
     2   DOTMIG(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),COTMP(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
C                                                                       BSURFA    12
      RB=0.0                                                            BSURFA    13
      XB = XB + DX                                                      BSURFA    14
C                                                                       BSURFA    15
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           BSURFA    16
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     BSURFA    17
      BUGGER = (GAM - 1.0)/TWOGJ                                        BSURFA    18
      IP1 = I+1                                                         BSURFA    19
      JP1 = J+1                                                         BSURFA    20
      F1 = PHIBG(I,JP1)                                                 BSURFA    21
      F4 = PHIBG(IP1,J)*AREAC(IP1)                                      BSURFA    22
      F5 = PHIBG(I,J)*AREAC(I)                                          BSURFA    23
      G1 = F1*RHOBG(I,JP1)                                              BSURFA    24
      G4 = F4*RHOBG(IP1,J)                                              BSURFA    25
      E4 = G4*UBG(IP1,J)                                                BSURFA    26
      D1 = G1*VBG(I,JP1)                                                BSURFA    27
      C1 = G1*HBG(I,JP1)                                                BSURFA    28
      C4 = G4*HBG(IP1,J)                                                BSURFA    29
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    BSURFA    30
      B4 = BUGGER*UBG(IP1,J)**2*G4                                      BSURFA    31
C*                                                                      BSURFA    32
      DENOM = F1 + F4                                                   BSURFA    33
      PHIAVE = DENOM/(AREAC(I) + AREAC(IP1))                            BSURFA    34
      RHOAVE = (G1+G4)/DENOM                                            BSURFA    35
      UBGAVE = 0.0                                                      BSURFA    36
      VBGAVE = 0.0                                                      BSURFA    37
      UPBAVE = (UPB(I,JP1) + UPB(IP1,J))*0.5                            BSURFA    38
C*                                                                      BSURFA    39
      PHITOT = PHIBTD(I,J)                                              BSURFA    40
      RHOTOT = (F5*RHOAVE - DTDX*E4 - TWOTDR*D1                         BSURFA    41
     $        + DELT*DOTMIG(I)/DVAXIS + DOTMB(I,J)*AREAC(I) )/          BSURFA    42
     $            (PHITOT*AREAC(I))                                     BSURFA    43
      PHIRHO = PHITOT*RHOTOT*AREAC(I)                                   BSURFA    44
C                                                                       BSURFA    45
      UTOT = 0.0                                                        BSURFA    46
      VTOT = 0.0                                                        BSURFA    47
C*                                                                      BSURFA    48
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)               BSURFA    49
```

```
C*                                                                      BSURFA    50
      HIGN = HBG(I,1)                                                   BSURFA    51
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)                        BSURFA    52
C                                                                       BSURFA    53
      HBGTD(I,J) = GAM*( F5*(C1+B1+C4+H4)/(DENOM*GAM)                   BSURFA    54
     $       - DTDX*C4*UBG(IP1,J) - TWOTDR*C1*HBG(I,JP1)                BSURFA    55
     $        + DELT*( DOTMIG(I)/DVAXIS*HIGN - QBAG(I,J) )              BSURFA    56
     $        + DOTMB(I,J)*HMB*AREAC(I) )/PP*RHO                        BSURFA    57
C*                                                                      BSURFA    58
C                                                                       BSURFA    59
      RHOBTD(I,J) = RHOTDT                                              BSURFA    60
      UBGTD(I,J) = UTDT                                                 BSURFA    61
      VBGTD(I,J) = VTDT                                                 BSURFA    62
      RETURN                                                            BSURFA    63
      END                                                               BSURFA    64
      SUBROUTINE BSURFT                                                 BSURFT     2
C     IX IS I HERE, IR IS J                                             BSURFT     3
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           BSURFT     4
     $  UTDT,VBGAVE,VTDT                                                BSURFT     5
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBLGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAL(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAH2,DIAMB1,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            BSURFT     7
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/R0,RR0,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   DOTMIG(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),DDTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),TZF(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
C                                                                       BSURFT    12
      RB=RB+DR                                                          BSURFT    13
C                                                                       BSURFT    14
      CALL GSPROP(R0,RR0,R,CV0,CVR,CV,PCH(I,J),HBG(I,J),TDUM,           BSURFT    15
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     BSURFT    16
      BUGGER = (GAM - 1.0)/TWOGJ                                        BSURFT    17
      IP1 = I+1                                                         BSURFT    18
      JP1 = J+1                                                         BSURFT    19
      F1 = PHIBG(I,JP1)                                                 BSURFT    20
      F4 = PHIBG(IP1,J)                                                 BSURFT    21
      F5 = PHIBG(I,J)                                                   BSURFT    22
      G1 = F1*RHOBG(I,JP1)                                              BSURFT    23
      G4 = F4*RHOBG(IP1,J)                                              BSURFT    24
      E4 = G4*UBG(IP1,J)                                                BSURFT    25
      D1 = G1*VBG(I,JP1)                                                BSURFT    26
      C1 = G1*HBG(I,JP1)                                                BSURFT    27
      C4 = G4*HBG(IP1,J)                                                BSURFT    28
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    BSURFT    29
      B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2 )*G4                   BSURFT    30
      DENOM = F1 + F4                                                   BSURFT    31
      PHIAVE = DENOM*0.5                                                BSURFT    32
      RHOAVE = (G1+G4)/DENOM                                            BSURFT    33
      UBGAVE = 0.0                                                      BSURFT    34
      VBGAVE = 0.0                                                      BSURFT    35
      UPBAVE = (UPB(I,JP1) + UPB(IP1,J))*0.5                            BSURFT    36
C*                                                                      BSURFT    37
      PHITDT = PHIBTD(I,J)                                              BSURFT    38
      RHOTDT = ( F5*RHOAVE - DTDX*E4 - CTDR*D1                          BSURFT    39
     $         - DELT*DOTMIG(I)/DVAXIT + DOTMB(I,J) )/PHITDT            BSURFT    40
      PHIRHO = PHITDT*RHOTDT                                            BSURFT    41
      UTDT = 0.0                                                        BSURFT    42
      VTDT = 0.0                                                        BSURFT    43
C*                                                                      BSURFT    44
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)               BSURFT    45
C*                                                                      BSURFT    46
      HIGN = HBG(I,1)                                                   BSURFT    47
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)                        BSURFT    48
C                                                                       BSURFT    49
```

```
      HBGTD(I,J) = GAM*( F5*(C1+B1+C4+B4)/(DENOM*EAM)
     $       - DTDX*C4*UBG(IP1,J) - DTDR*C1*VBG(I,JP1)
     $       - DELT*( DOTMIG(I)/DVAXIT*HIGA + QBAG(I,J) )
     $       + DOTMB(I,J)*HMB  )/PHIRHO
C*
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE BSURFI                                                 BSURFI     2
      COMMON/AVGDT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           BSURFI     3
     $ UTOT,VBGAVE,VTOT                                                 BSURFI     4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAAA,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAH2,DIAMB1,BELENG,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            BSURFI     6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      BSURFI     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   DOTMIG(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),CLTMP(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/                                                  BSURFI    11
C                                                                       BSURFI    12
      RB=RB+DR                                                          BSURFI    13
C*                                                                      BSURFI    14
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           BSURFI    15
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     BSURFI    16
      BUGGER = (GAM - 1.0)/TWOGJ                                        BSURFI    17
      IP1 = I+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(IP1,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
      E4 = 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
      C1 = G1*HBG(I,JP1)                                                BSURFI    32
      C3 = G3*HBG(I,JM1)                                                BSURFI    33
      C4 = G4*HBG(IP1,J)                                                BSURFI    34
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    BSURFI    35
      B3 = BUGGER*( UBG(I,JM1)**2 + VBG(I,JP1)**2 )*G3                  BSURFI    36
      B4 = BUGGER*( UBG(IP1,J)**2 + VBG(IP1,J)**2 )*G4                  BSURFI    37
      DRP = ( RB + DR )/RB                                              BSURFI    38
      DRM = ( RB - DR )/RB                                              BSURFI    39
C*                                                                      BSURFI    40
      DENOM = F1 + F3 + F4 + F4                                         BSURFI    41
      SIGG = G1 + G3 + G4 + G4                                          BSURFI    42
      SIGD = D1 + D3 + D4 + D4                                          BSURFI    43
      PHIAVE = DENOM*0.25                                               BSURFI    44
      RHOAVE = SIGG/DENOM                                               BSURFI    45
      UBGAVE = 0,0                                                      BSURFI    46
      VBGAVE = SIGD/SIGG                                                BSURFI    47
      UPBAVE = (UPB(I,JP1) + UPB(I,JM1) + 2,0*UPB(IP1,J))*0.25          BSURFI    48
C*                                                                      BSURFI    49
```

```
      PHITDT = PHIBTD(I,J)
      RHOTDT = ( F5*RHOAVE - DTDX*E4 - T2DR*(DRP*C1 - DRM*D3)
     $        + DDTMB(I,J)   )/PHITDT
      PHIRHO = PHITDT*RHOTDT
C*
      UTDT = 0.0
      VTDT = (F5*SIGD/DENOM - DTDX*E4*VBG(IP1,J)
     $       - T2DR*( DRP*( D1*VBG(I,JP1) + GRAV    *PCH(I,JP1) )
     $                  - DRM*( D3*VBG(I,JP1) + GRAV    *PCH(I,JM1) ) )
     $       + DELT*GRAV*PCH(I,J)/RB ) / PHIRHO
      IF( ABS(VTDT) .LT. 0.1 ) VTDT = 0.0
C*
C     ARRAY XDRAG IS CLEARED IN MAIN AT THE BEGINNING OF EACH TIME
C     INTERVAL SO IF SUBROUTINE DRAG IS NOT CALLED, XDRAG(I,J) IS 0.0.
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)
C*
      HBGTD(I,J) = GAM*( F5*(C1+B1+C3+B3+C4+C4+B4+B4)/(DENOM*GAM)
     $      - DTDX*C4*UBG(IP1,J)
     $       - T2DR*(DRP*C1*VBG(I,JP1) - DRP*C3*VBG(I,JM1) )
     $        - DELT*QBAG(I,J) + DDTMB(I,J)*HMB
     $         - BUGGER*VTDT*VTDT*PHIRHO/GAM )/PHIRHO
C*
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE BSURFB                                                 BSURFB     2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,          BSURFB     3
     $ UTDT,VBGAVE,VTDT                                                 BSURFB     4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            BSURFB     6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $ DX,DR,NX,GJ,TWODT,HBP                                            EQNS       3
      COMMON/GASCON/RO,RRO,CVO,CVR                                      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   DOTMIG(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),DCTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),T2R(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
C                                                                       BSURFB    11
      RB=RO+DR                                                          BSURFB    12
C*                                                                      BSURFB    13
      CALL GSPROP(RO,RRO,R,CVO,CVR,CV,PCH(I,J),HBG(I,J),TDUM,           BSURFB    14
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      BSURFB    15
      BUGGER = (GAM - 1.0)/TWOGJ                                        BSURFB    16
      IP1 = I+1                                                         BSURFB    17
      JM1 = J-1                                                         BSURFB    18
      F3 = PHIBG(I,JM1)                                                 BSURFB    19
      F4 = PHIBG(IP1,J)                                                 BSURFB    20
      F5 = PHIBG(I,J)                                                   BSURFB    21
      G3 = F3*RHOBG(I,JM1)                                              BSURFB    22
      G4 = F4*RHOBG(IP1,J)                                              BSURFB    23
      E4 = G4*UBG(IP1,J)                                                BSURFB    24
      D3 = G3*VBG(I,JM1)                                                BSURFB    25
      C3 = G3*HBG(I,JM1)                                                BSURFB    26
      C4 = G4*HBG(IP1,J)                                                BSURFB    27
      B3 = BUGGER*VBG(I,JM1)**2*G3                                      BSURFB    28
      B4 = BUGGER*(UBG(IP1,J)**2 + VBG(IP1,J)**2 )*G4                   BSURFB    29
C*                                                                      BSURFB    30
      DENOM = F3 + F4                                                   BSURFB    31
      PHIAVE = DENOM*0.5                                                BSURFB    32
      RHOAVE = (G3+G4)/DENOM                                            BSURFB    33
      UBGAVE = 0.0                                                      BSURFB    34
      UPBAVE = (UPB(I,JM1) + UPB(IP1,J))*0.5                            BSURFB    35
      VBGAVE = 0.0                                                      BSURFB    36
C*                                                                      BSURFB    37
      PHITDT = PHIBTD(I,J)                                              BSURFB    38
      RHOTDT = ( F5*RHOAVE - DTDX*E4 + DTDR*D3                          BSURFB    39
     $         + DOTMB(I,J)   )/PHITDT                                  BSURFB    40
      PHIRHO = PHITDT*RHOTDT                                            BSURFB    41
      UTDT = 0.0                                                        BSURFB    42
      VTDT = 0.0                                                        BSURFB    43
C*                                                                      BSURFB    44
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)               BSURFB    45
C*                                                                      BSURFB    46
      HBGTD(I,J) = GAM*( F5*(B3+C3+B4+C4)/(DENOM*GAM)                   BSURFB    47
     $        - DTDX*C4*UBG(IP1,J) + DTDR*C3*HBG(I,JM1)                 BSURFB    48
     $         - DELT*GBAG(I,J) + DOTMB(I,J)*HMB  )/PHIRHO              BSURFB    49
```

```
C*
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE FSURFA                                                 FSURFA       2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           FSURFA       3
     $  UTDT,VBGAVE,VTDT                                                FSURFA       4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI        2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI        3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,UPGAP,AREAGP(60),DAVG,       CHAMI        4
     $  AREAH2,DIAPBI,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI        5
      COMMON/CLOCK/TIME,DELT                                            FSURFA       6
      COMMON/EQNS/DTDX,I2DR,I2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     $  DX,DR,NX,GG,TWODT,HBP                                           EQNS         3
      COMMON/GASCON/RO,RRO,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), TZC(60,5),                    BAG          3
     2   DOTMIG(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),DCTMP(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
C                                                                       FSURFA      11
      RB=0.0                                                            FSURFA      12
      XB=XB+DX                                                          FSURFA      13
C                                                                       FSURFA      14
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           FSURFA      15
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      FSURFA      16
      BUGGER = (GAM - 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
      E2 = G2*UBG(IM1,J)                                                FSURFA      25
      D1 = G1*VBG(I,JP1)                                                FSURFA      26
      C1 = G1*HBG(I,JP1)                                                FSURFA      27
      C2 = G2*HBG(IM1,J)                                                FSURFA      28
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    FSURFA      29
      B2 = 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
      PHITDT = PHIBTD(I,J)                                              FSURFA      39
      RHOTDT = ( F5*RHOAVE + DTDX*E2 - TWOTDR*D1                        FSURFA      40
     $         + DELT*DOTMIG(I)/DVAXIS + DOTPB(I,J)*AREAC(I) )/         FSURFA      41
     $          (PHITDT*AREAC(I))                                       FSURFA      42
      PHIRHO = PHITDT*RHOTDT*AREAC(I)                                   FSURFA      43
      UTDT = 0.0                                                        FSURFA      44
      VTDT = 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
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)
C
      HBGTD(I,J) = GAM*( F5*(B1+C1+B2+C2)/(LENGM*GAM)
     $        + DTDX*C2*UBG(IM1,J) - TWOTDR*L1*HBG(I,JP1)
     $        + DELT*( DOTMIG(I)/DVAXIS*HIGN - QBAG(I,J) )
     $        + DOTMB(I,J)*HMB*AREAC(I) )/PHIRHO
C*
C
      RHOBTD(I,J) = RHOTOT
      UBGTD(I,J) = UTOT
      VBGTD(I,J) = VTOT
      RETURN
      END
      SUBROUTINE FSURFT                                                 FSURFT     2
      COMMON/AVGDT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           FSURFT     3
     $  UTOT,VBGAVE,VTOT                                                FSURFT     4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI      2
     $    AREACH,AREAL(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,  CHAMI      3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            FSURFT     6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/RO,RRO,CVO,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   DOTMIG(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),COTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),T2R(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
C*                                                                      FSURFT    11
      RB=RB+DR                                                          FSURFT    12
C                                                                       FSURFT    13
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           FSURFT    14
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      FSURFT    15
      BUGGER = (GAM - 1.0)/TWOGJ                                        FSURFT    16
      IM1 = I -1                                                        FSURFT    17
      JP1 = J+1                                                         FSURFT    18
      F1 = PHIBG(I,JP1)                                                 FSURFT    19
      F2 = PHIBG(IM1,J)                                                 FSURFT    20
      F5 = PHIBG(I,J)                                                   FSURFT    21
      G1 = F1*RHOBG(I,JP1)                                              FSURFT    22
      G2 = F2*RHOBG(IM1,J)                                              FSURFT    23
      E2 = G2*UBG(IM1,J)                                                FSURFT    24
      D1 = G1*VBG(I,JP1)                                                FSURFT    25
      C1 = G1*HBG(I,JP1)                                                FSURFT    26
      C2 = G2*HBG(IM1,J)                                                FSURFT    27
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    FSURFT    28
      B2 = BUGGER*( 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(I,JP1) + UPB(IM1,J))*0.5                            FSURFT    36
C                                                                       FSURFT    37
      PHITOT = PHIBTD(I,J)                                              FSURFT    38
      RHOTOT = ( F5*RHOAVE + DTDX*E2 - CTDR*D1                          FSURFT    39
     $         - DELT*DOTMIG(I)/DVAXIT + DOTPB(I,J) )/PHITDT            FSURFT    40
      PHIRHO = PHITDT*RHOTOT                                            FSURFT    41
      UTOT = 0.0                                                        FSURFT    42
      VTOT = 0.0                                                        FSURFT    43
C                                                                       FSURFT    44
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)               FSURFT    45
C*                                                                      FSURFT    46
      HIGN = HBG(I,1)                                                   FSURFT    47
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)                        FSURFT    48
C                                                                       FSURFT    49
```

```
      HBGTD(I,J) = GAM*( F5*(B1+C1+B2+C2)/(DENOM*GAM)                  FSURFT      50
     $        + DTDX*C2*UBG(IM1,J) - DTDY*C1*VBG(I,JP1)                 FSURFT      51
     $        - DELT*( DOTMIG(I)/DVAXIT*HIGN + QBAG(I,J) )              FSURFT      52
     $        + DOTMB(I,J)*HMB  )/PHIRHO                                FSURFT      53
C*                                                                      FSURFT      54
C                                                                       FSURFT      55
      RHOBTD(I,J) = RHOTOT                                              FSURFT      56
      UBGTD(I,J) = UTOT                                                 FSURFT      57
      VBGTD(I,J) = VTOT                                                 FSURFT      58
      RETURN                                                            FSURFT      59
      END                                                               FSURFT      60
      SUBROUTINE FSURFI                                                 FSURFI     2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           FSURFI     3
     $  UTDT,VBGAVE,VTDT                                                FSURFI     4
      COMMON/CHAM/I  ,J  ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),  CHAMI    2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMI      4
     $  AREA[illegible],DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN      CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            FSURFI     6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/R0,RRO,CV0,CVH                                      FSURFI     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   DOTMIG(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),DCTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),T2R(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
      DATA GRAV/32.16/                                                  FSURFI    11
C*                                                                      FSURFI    12
      RB=RB+DR                                                          FSURFI    13
C                                                                       FSURFI    14
      CALL GSPROP(R0,RRO,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           FSURFI    15
     $        RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                    FSURFI    16
      BUGGER = (GAM - 1.0)/TWOGJ                                        FSURFI    17
      IM1 = I -1                                                        FSURFI    18
      JP1 = J+1                                                         FSURFI    19
      JM1 = J-1                                                         FSURFI    20
      F1 = PHIBG(I,JP1)                                                 FSURFI    21
      F2 = PHIBG(IM1,J)                                                 FSURFI    22
      F3 = PHIBG(I,JM1)                                                 FSURFI    23
      F5 = PHIBG(I,J)                                                   FSURFI    24
      G1 = F1*RHOBG(I,JP1)                                              FSURFI    25
      G2 = F2*RHOBG(IM1,J)                                              FSURFI    26
      G3 = F3*RHOBG(I,JM1)                                              FSURFI    27
      E2 = G2*UBG(IM1,J)                                                FSURFI    28
      D1 = G1*VBG(I,JP1)                                                FSURFI    29
      D2 = G2*VBG(IM1,J)                                                FSURFI    30
      D3 = G3*VBG(I,JM1)                                                FSURFI    31
      C1 = G1*HBG(I,JP1)                                                FSURFI    32
      C2 = G2*HBG(IM1,J)                                                FSURFI    33
      C3 = G3*HBG(I,JM1)                                                FSURFI    34
      B1 = BUGGER*(UBG(I,JP1)**2 + VBG(I,JP1)**2)*G1                    FSURFI    35
      B2 = BUGGER*(UBG(IM1,J)**2 + VBG(IM1,J)**2)*G2                    FSURFI    36
      B3 = BUGGER*( UBG(I,JM1)**2 + VBG(I,JM1)**2 )*G3                  FSURFI    37
      DRP = (RB + DR)/RB                                                FSURFI    38
      DRM = (RB - DR)/RB                                                FSURFI    39
C*                                                                      FSURFI    40
      DENOM = F1 + F2 + F2 + F3                                         FSURFI    41
      SIGG = G1 + G2 + G2 + G3                                          FSURFI    42
      SIGD = D1 + D2 + D2 + D3                                          FSURFI    43
      PHIAVE = DENOM*0.25                                               FSURFI    44
      RHOAVE = SIGG/DENOM                                               FSURFI    45
      UBGAVE = 0.0                                                      FSURFI    46
      VBGAVE = SIGD/SIGG                                                FSURFI    47
      UPBAVE = (UPB(I,JP1) + 2.0*UPB(IM1,J) + UPB(I,JM1))*0.25          FSURFI    48
C*                                                                      FSURFI    49
```

```
      PHITDT = PHIBTD(I,J)
      RHOTDT = ( F5*RHOAVE + DTDX*E2
     $        - T2DR*(DRP*D1 - DRM*D3)
     $        + DOTMB(I,J)   )/PHITDT
      PHIRHO = PHITDT*RHOTDT
C*
      UTDT = 0.0
      VTDT = ( F5*SIGD/DENOM + DTDX*E2*VBG(IM1,J)
     $     - T2DR*( DRP*( D1*VBG(I,JP1) + GRAV   *PCH(I,JP1) )
     $              - DRM*( D3*VBG(I,JP1) + GRAV   *PCH(I,JM1) ) )
     $     + DELT*GRAV*PCH(I,J)/RB )/PHIRHO
      IF( ABS(VTDT) .LT. 0.1 ) VTDT = 0.0
C*
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)
      RBGTD(I,J) = GAM*(F5*(B1+C1+B2+C2+B2+C2+B3+C3)/(DENOM*GAM)
     $        + DTDX*C2*UBG(IM1,J)
     $        - T2DR*( DRP*C1*VBG(I,JP1) - DRM*C3*VBG(I,JM1) )
     $        - DELT*DRAG(I,J) + DOTMB(I,J)*RMB
     $          - BUGGER*VTDT*VTDT*PHIRHO/GAM )/PHIRHO
C*
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE FSURFB                                                 FSURFB     2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           FSURFB     3
     1  UTOT,VBGAVE,VTOT                                                FSURFB     4
      COMMON/CHAMI/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI     2
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI      3
     2  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMI      4
     3  AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI      5
      COMMON/CLOCK/TIME,DELT                                            FSURFB     6
      COMMON/CONS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   CONS       2
     1  DX,DR,NX,GG,TWODT,HBP                                           CONS       3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      FSURFB     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), TZC(60,5),                    BAG        3
     2   DOTMIG(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),CLTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIFIL(60,5),TZR(60),TBP(60,5)                                  BAG        7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG    2
C*                                                                      FSURFB    11
      RB=RB+DR                                                          FSURFB    12
C*                                                                      FSURFB    13
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           FSURFB    14
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     FSURFB    15
      BUGGER = (GAM - 1.0)/TWOGJ                                        FSURFB    16
      IM1 = I -1                                                        FSURFB    17
      JM1 = J-1                                                         FSURFB    18
      F2 = PHIBG(IM1,J)                                                 FSURFB    19
      F3 = PHIBG(I,JM1)                                                 FSURFB    20
      F5 = PHIBG(I,J)                                                   FSURFB    21
      G2 = F2*RHOBG(IM1,J)                                              FSURFB    22
      G3 = F3*RHOBG(I,JM1)                                              FSURFB    23
      E2 = G2*UBG(IM1,J)                                                FSURFB    24
      D3 = G3*VBG(I,JM1)                                                FSURFB    25
      C2 = G2*HBG(IM1,J)                                                FSURFB    26
      C3 = G3*HBG(I,JM1)                                                FSURFB    27
      B2 = BUGGER*UBG(IM1,J)**2*G2                                      FSURFB    28
      B3 = BUGGER*( UBG(I,JM1)**2 + VBG(I,JM1)**2 )*G3                  FSURFB    29
C*                                                                      FSURFB    30
      DENOM = F2 + F3                                                   FSURFB    31
      PHIAVE = DENOM*0.5                                                FSURFB    32
      RHOAVE = (G2+G3)/DENOM                                            FSURFB    33
      UBGAVE = 0.0                                                      FSURFB    34
      VBGAVE = 0.0                                                      FSURFB    35
      UPBAVE = (UPB(IM1,J) + UPB(I,JM1))*0.5                            FSURFB    36
C*                                                                      FSURFB    37
      PHITDT = PHIBTD(I,J)                                              FSURFB    38
      RHOTDT = ( F5*RHOAVE + DTDX*E2 + DTDR*D3                          FSURFB    39
     $        + DOTMB(I,J)   )/PHITDT                                   FSURFB    40
      PHIRHO = PHITDT*RHOTDT                                            FSURFB    41
      UTOT = 0.0                                                        FSURFB    42
      VTOT = 0.0                                                        FSURFB    43
C                                                                       FSURFB    44
C*****IF(PHIAVE .LT. 0.999) CALL DRAG(XDRAG(I,J),.FALSE.)               FSURFB    45
      HBGTD(I,J) = GAM*( F5*(B2+C2+B3+C3)/(DENOM*GAM)                   FSURFB    46
     $        + DTDX*E2*HBG(IM1,J) + DTDR*D3*HBG(I,JM1)                 FSURFB    47
     $        - DELT*GBAG(I,J) + DOTMB(I,J)*HMB   )/PHIRHO              FSURFB    48
C*                                                                      FSURFB    49
```

```
C                                                                      FSURFB     50
      RHOBTD(I,J) = RHOTDT                                             FSURFB     51
      UBGTD(I,J) = UTDT                                                FSURFB     52
      VBGTD(I,J) = VTDT                                                FSURFB     53
      RETURN                                                           FSURFB     54
      END                                                              FSURFB     55
      SUBROUTINE DIMIN
C
C     SUBROUTINE  DIMIN IS THE BARREL ROUTINE EQUIVALENT TO REGRESS IN
C     THE CHAMBER,  WE ASSUME ONLY IGNITED PROPELLANT IS IN THE BARREL.
C
      COMMON/BURN/ATPB,CT,PEXP
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GO,TWODT,HEP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTOT(60,5), DOTOT(60,5), DITOT(60,5), XL0,  DO0, DI0,
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOF,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM
      COMMON/SPLINT/WHOLEC,WHOLEB
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100),  PMDOT(100), GL(100), DDRAG(100), FRICT(100),
     2  QCONV(100), DUP(100), DPHI(100), DRHOG(100), DHG(100), DUG(100),
     3  AMASS(100), AMOM(100), AENER(100), DAMASS(100), DAMOM(100),
     4  DAENER(100)
      LOGICAL WHOLEC,WHOLEB
      DATA PIDF/.785398/
C
C     SUBROUTINE DIMIN CALCULATES DPHI FOR USE IN RHOUH AND IN PROPMO.
C     IT UPDATES GRAIN DIMENSIONS FOR USE IN PROPMO.
C
C     ARRAY PMDOT IS CLEARED IN MAIN AT EACH TIME INTERVAL.
C
      DO 50 I=2,NX
      IF(PHI(I).GE.0.99999) GO TO 45
C
         R = ATPB*PG(I)**PEXP + CT
         BURNL = R*TWODT
         DXLB(I) = XLB(I) - BURNL
C
         IF( DOB(I) .LE. 3.0*DIB(I) ) GO TO 20
C
C     CODING FOR CYLINDRICAL PROPELLANT GRAINS
         DDOB(I) = DOB(I) - BURNL
         DDIB(I) = DIB(I) + BURNL
         VOLD = PIDF*XLB(I)*( DOB(I)*DOB(I) - FN*DIB(I)*DIB(I) )
         VNEW = PIDF*DXLB(I)*( DDOB(I)*DDOB(I) -
     $            FN*DDIB(I)*DDIB(I) )
         GO TO 30
C
C
C     CODING FOR SPLINTERED PROPELLANT GRAINS
 20      CONTINUE
         IF( DOB(I) .LE. DIB(I) ) GO TO 25
         WHOLEB = .FALSE.
         AREA = PIDF*(DOB(I)*DOB(I) - FN*DIB(I)*DIB(I))
         DIB(I) = 3.14*(DOB(I) + FN*DIB(I))
         DOB(I) = AREA
C
 25      DELR = BURNL*0.5
         DDOB(I) = DOB(I) - DIB(I)*DELR
         IF(DDOB(I) .GE. 1.0E-7) GO TO 27
         DDOB(I) = 0.0
```

```
            UDIB(I) = 0.0
            UXLB(I) = 0.0
            UPHI(I) = 1.0
            GO TO 50
C
 27         CONTINUE
            UDIB(I) = SQRT( DIB(I)*DIB(I)/DOB(I)*UDOB(I) )
            VOLD = XLB(I)*DOB(I)
            VNEW = UXLB(I)*UDOB(I)
C
C
C
 30         IF(VNEW .LE. 0.0) GO TO 45
            DELTAV = VOLD - VNEW
            PNDV = (1.0 - PHI(I))/VOLD
            TEMP = PNDV*DELTAV
            PMDOT(I) = TEMP*RHOP
            UPHI(I) = PHI(I) + TEMP
            GO TO 50
C
C
 45         UPHI(I) = PHI(I)
C
C
 50   CONTINUE
C
      RETURN
      END
C
C
C
C
C
```

```
      SUBROUTINE RHOUH                                                  RHOUH      2
C                                                                       RHOUH      3
C     SUBROUTINE RHOUH PERFORMS THE FINITE DIFFERENCE CALCULATIONS AT   RHOUH      4
C     THE BARREL GRIDS.                                                 RHOUH      5
C                                                                       RHOUH      6
      COMMON/AVGOT/ORPOT,PHIRHO,PHIAVE,RHOAVE,UGAVE,UPAVE,UUT,          RHOUH      7
     $  VRGAVE,VTOT                                                     RHOUH      8
      COMMON/BARRL2/BOREA,AP,VP,BORED,PCHER,BORED2,DT2BU,DTUSQ,XLBAR    RHOUH      9
      COMMON/CLOCK/TIME,DELT                                            RHOUH     10
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTER,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GC,TWODT,HBP                                           EQNS       3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      RHOUH     12
      COMMON/GRIDNX/DXPRIM                                              MOD0301   26
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL      2
     1  PG(100), TG(100),  PMDOT(100), GL(100), UDRAG(100), FRICT(100), BARRL      3
     2  QCONV(100), DUP(100), DPHI(100), DRHOG(100), UHG(100), UUG(100),BARRL      4
     3  AMASS(100), AMOM(100), AENER(100), DAMASS(100), DAMOM(100),     BARRL      5
     4  DAENER(100)                                                     BARRL      6
C                                                                       RHOUH     14
      DATA GRAV/32.16/                                                  RHOUH     15
C                                                                       RHOUH     16
C                                                                       RHOUH     17
      NM1 = NX-1                                                        RHOUH     18
      NP1 = NX+1                                                        RHOUH     19
      PHI(NP1) = PHI(NM1)                                               RHOUH     20
      RHOG(NP1) = RHOG(NM1)                                             RHOUH     21
      UG(NP1) = 2.0*VP - UG(NM1)                                        RHOUH     22
      AMASS(NP1) = AMASS(NM1)                                           RHOUH     23
      AMOM(NP1) = AMOM(NM1)                                             RHOUH     24
      AENER(NP1) = AENER(NM1)                                           RHOUH     25
      HG(NP1) = HG(NM1)                                                 RHOUH     26
      UP(NP1) = 2*VP- UP(NM1)                                           MOD1205    9
      PG(NP1)=PG(NM1)                                                   RHOUH     28
C                                                                       RHOUH     29
C***RHOUH SHOULD NOT BE CALLED IF NX IS 1.                              RHOUH     30
C                                                                       RHOUH     31
      DO 100 I=2,NX                                                     RHOUH     32
         IM1=I-1                                                        RHOUH     33
         IP1=I+1                                                        RHOUH     34
C                                                                       RHOUH     35
         CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PG(I),HG(I),TDUM,RHOG(I),      RHOUH     36
     $             UG(I),0.0,GAM,CP,2)                                  RHOUH     37
         BUGGER = (GAM - 1.0)/TWOGJ                                     RHOUH     38
C                                                                       MOD0301   27
C     WHEN I = NX  - 1, GRID I + 1 SHOULD BE THE SAME DISTANCE AWAY AS  MOD0301   28
C     GRID NX - 2 (GRID NX WILL PROBABLY BE FARTHER AWAY), SO USE       MOD0301   29
C     LINEAR INTERPOLATION TO GET THE PROPERTIES AT I + 1.  LET PHI     MOD0301   30
C     REMAIN THE SAME.                                                  MOD0301   31
      IF(I .NE. NX - 1) GO TO 30                                        MOD0301   32
      RHOS = RHOG(NX)                                                   MOD0301   33
      UGS = UG(NX)                                                      MOD0301   34
      HGS = HG(NX)                                                      MOD0301   35
      PGS = PG(NX)                                                      MOD0301   36
      UPS = UP(NX)                                                      MOD0301   37
      RATIO = DX/DXPRIM                                                 MOD0301   38
      RHOG(IP1) = RHOG(I) + RATIO*(RHOG(IP1) - RHCG(I))                 MOD0301   39
      UG(IP1) = UG(I) + RATIO*(UG(IP1) - UG(I))                         MOD0301   40
      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                                                                       MOD0301     44
C     AT GRID NX DXPRIM SHOULD BE USED RATHER THAN DX, SO T2DX MUST BE  MOD0301     45
C     CHANGED.                                                          MOD0301     46
 30   CONTINUE                                                          MOD0301     47
      IF(I .NE. NX) GO TO 40                                            MOD0301     48
      T2DXS = T2DX                                                      MOD0301     49
      T2DX = DELT/(2.0*DXPRIM)                                          MOD0301     50
C                                                                       MOD0301     51
 40   CONTINUE                                                          MOD0301     52
C                                                                       RHOUH       39
           F1 = PHI(IM1)                                                RHOUH       40
           F2 = PHI(IP1)                                                RHOUH       41
           F5 = PHI(I)                                                  RHOUH       42
           G1 = RHOG(IM1)                                               RHOUH       43
           G2 = RHOG(IP1)                                               RHOUH       44
           G5 = RHOG(I)                                                 RHOUH       45
           E1 = G1*UG(IM1)                                              RHOUH       46
           E2 = G2*UG(IP1)                                              RHOUH       47
           E5 = G2*UG(I)                                                RHOUH       48
           C1 = G1*HG(IM1)                                              RHOUH       49
           C2 = G2*HG(IP1)                                              RHOUH       50
           C5 = G5*HG(I)                                                RHOUH       51
           B1 = BUGGER*UG(IM1)*E1                                       RHOUH       52
           B2 = BUGGER*UG(IP1)*E2                                       RHOUH       53
           B5 = BUGGER*UG(I)*E5                                         RHOUH       54
C                                                                       RHOUH       55
           PHIAVE = (F1 + F5 + F5 + F2)*0.25                            RHOUH       56
           RHOAVE = (G1 + G5 + G5 + G2)*0.25                            RHOUH       57
           UGAVE = (UG(IM1) + UG(I) + UG(I) + UG(IF1))*0.25             RHOUH       58
      IF(UP(I).EQ.0.0.AND.PHIAVE.NE.0.)UP(I)=UGAVE                      RHOUH       59
      UPAVE=(UP(IM1)+UP(I)+UP(I)+UP(IP1))*0.25                          RHOUH       60
C                                                                       RHOUH       61
           UPHIT = UPHI(I)                                              RHOUH       62
C                                                                       RHOUH       63
           URHOT = ( F5*AMASS(I)*RHOAVE                                 RHOUH       64
     1          - T2DX*(F2*AMASS(IP1)*E2 - F1*AMASS(IM1)*E1)            RHOUH       65
     2          + PMDOT(I)*AMASS(I) )/(UPHI(I*DAMASS(I))                RHOUH       66
C                                                                       RHOUH       67
           PHIRHO = UPHIT*URHOT                                         RHOUH       68
C                                                                       RHOUH       69
           GRAVA = GRAV*AMOM(I)                                         RHOUH       70
           UUT = ( F5*AMOM(I)*(E1 + E5 + E5 + E2)*0.25                  RHOUH       71
     1          - T2DX*(F2*AMOM(IP1)*E2*UG(IP1) - F1*AMOM(IM1)*E1*UG(IM1)  RHOUH    72
     2             + GRAVA*PG(IP1) - GRAVA*PG(IM1))                     RHOUH       73
     3          - DELT*FRICT(I) + AMOM(I)*PMDOT(I)*UP(I) )/             RHOUH       74
     4          (PHIRHO*DAMOM(I))                                       RHOUH       75
C                                                                       RHOUH       76
           IF(ABS(UUT) .LT. 0.1) UUT = 0.0                              RHOUH       77
      IF(PHIAVE.LT.0.999) CALL DRAG(UDRAG(I),.TRUE.,1.0)                RHOUH       78
C                                                                       RHOUH       79
           UHG(I) = GAM*( F5*AENER(I)*(C1+B1+C5+B5+C5+B5+C2+B2)/        RHOUH       80
     1          (4.0*GAM) -  T2DX*(F2*AENER(IP1)*E2*HG(IP1) -           RHOUH       81
     2             F1*AENER(IM1)*E1*HG(IP1))                            RHOUH       82
     3          - QCONV(I) + PMDOT(I)*AENER(I)*HMB                      RHOUH       83
     *   -DDRAG(I)*UP(I)*DELT*AENER(I)/786.                              RHOUH      84
     4         - BUGGER*UUT*UUT*PHIRHO*DAENER(I)/GAM )/                  RHOUH      85
     5         (PHIRHO*DAENER(I))                                        RHOUH      86
C                                                                        RHOUH      87
C                                                                        RHOUH      88
          DRHOG(I) = DRHOT                                               RHOUH      89
          UUG(I) = UUT                                                   RHOUH      90
C     THE SAVED PROPERTIES AT GRID NX SHOULD BE PUT BACK INTO THE        MOD0301    53
C     APPROPRIATE ARRAYS BEFORE I IS SET TO NX.                          MOD0301    54
          IF(I .NE. NX - 1) GO TO 100                                    MOD0301    55
      RHOG(NX) = RHOS                                                    MOD0301    56
      UG(NX) = UGS                                                       MOD0301    57
      HG(NX) = HGS                                                       MOD0301    58
      PG(NX) = PGS                                                       MOD0301    59
      UP(NX) = UPS                                                       MOD0301    60
C                                                                        MOD0301    61
C                                                                        MOD0301    62
C                                                                        RHOUH      91
  100 CONTINUE                                                           RHOUH      92
C                                                                        MOD0301    63
C     REPLACE T2DX BY ITS SAVED VALUE.                                   MOD0301    64
      T2DX = T2DXS                                                       MOD0301    65
C                                                                        RHOUH      93
      RETURN                                                             RHOUH      94
C                                                                        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,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IUPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAR2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GJ,TWODT,HGP                                           EQNS       3
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTOT(60,5), DOTOT(60,5), DITOT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), UXLB(100), UDOB(100), UDIB(100)   GRAIN      4
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,GCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,HW,CM,TIGNBP,GBCONS,TOTM                        INPUTS     3
      COMMON/SPLINT/WHOLEC,WHOLEB                                       PROPMO    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), TZC(60,5),                    BAG        3
     2   DOTMIG(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),TBC(60,5),DOTMBG(60),CCTMP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),TZR(50),TBP(60,5)                                  BAG        7
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL      2
     1  PG(100), TG(100),  PMDOT(100), GL(100), UDRAG(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), UAMASS(100), UAMOM(100),     BARRL      5
     4  UAENER(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 PRPVEL.                        PROPMO    17
      IF(ONED) GO TO 10                                                 PROPMO    18
      PHI(1) = 0.0                                                      PROPMO    19
      XLB(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
          XLB(1) = XLB(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. 0.99999) GO TO 8                                   MOD0620    3
      TEMP = (1.0 - PHI(1))*AREACH                                      PROPMO    36
      XLB(1) = XLB(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
```

```
  5   CONTINUE
      XLB(1) = 0.0
      DOB(1) = 0.0
      DIB(1) = 0.0
      UP(1) = 0.0
      GO TO 15
C
 10   PHI(1) = PHIBG(NGX,1)
      XLB(1) = XL(NGX,1)
      DOB(1) = DO(NGX,1)
      DIB(1) = DI(NGX,1)
      UP(1) = UPB(NGX,1)
C
 15   CONTINUE
      IF(.NOT. WHOLEB  .AND.  DOB(1) .LE. DIB(1)) WHOLEB = .FALSE.
C
C     ARRAY UUP IS CLEARED IN MAIN AT EACH TIME INTERVAL.
C
C     PUT UPDATED POROSITY AND GRAIN DIMENSIONS CALCULATED IN DIMIN INTO
C     ARRAYS PHI, DOB, DIB, XLB.
      DO 20 I=2,NX
          PHI(I) = UPHI(I)
          DOB(I) = UDOB(I)
          DIB(I) = UDIB(I)
          XLB(I) = UXLB(I)
 20   CONTINUE
C
      DO 50 I=2,NX
C
C     UPDATE PROPELLANT VELOCITY.  PUT UPDATED VALUES IN ARRAY UP.
C     USE AN AVERAGE POROSITY IN UPDATING UP.
C     GTRHOP IS GRAV*DELT/RHOP
          PHIAVE = (PHI(I-1) + PHI(I) + PHI(I) + PHI(I+1))*0.25
          IF(I .EQ. NX) PHIAVE = (PHI(I-1) + PHI(I))*0.5
          IF(PHIAVE .GE. 0.99999) GO TO 30
          DELUP = GTRHOP*UDRAG(I)/(1.0 - PHIAVE)
          UP(I) = UP(I) + DELUP
C
C     AMOUNT OF PROPELLANT MOVED FROM ITH GRID INTO I+1ST GRID.
 30       CONTINUE
          DMPN = (1.0 - PHI(I))*DTDX*UP(I)
C
C     AMOUNT OF PROPELLANT MOVED FROM I-1ST GRID INTO ITH GRID.
C****THIS STATEMENT WILL GET PROPELLANT INTO THE BARREL ORIGINALLY.
          DMPN1 = (1.0 - PHI(I-1))*DTDX*UP(I-1)
C
C     UPDATED POROSITY AS A RESULT OF MOVEMENT
          PHIN = PHI(I) + DMPN - DMPN1
       IF(PHIN.LT.0.99999) GO TO 40
       UUP(I)=UG(I)
          UPHI(I) = 1.0
          UXLB(I) = XLB(I)
          UDOB(I) = DOB(I)
          UDIB(I) = DIB(I)
          GO TO 50
C
C
```

```
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)*DMPM1                   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) UDOB(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) UDIB(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                                                                       MOTION       3
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED2,DT2BD,DTDSQ,XLBAR    MOTION       4
      COMMON/CLOCK/TIME,DELT                                            MOTION       5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS         3
      COMMON/GASCON/R0,RHO,CV0,CVH                                      MOTION       7
      COMMON/GRAIN/ XL(60,5), DO(60,5), DI(60,5), FN,                   MOD0301     66
     1 XLTD(60,5), DOTDT(60,5), DITDT(60,5), XLO, DOO, DIO,             MOD0301     67
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   MOD0301     68
      COMMON/GRIDNX/DXPRIM                                              MOD0301     69
      COMMON/MOCON/CON1,CON2,CON3,CON4,CON5,AREAPB,Z0,WOB,XOB,ROTK,     MOTION       8
     $  FOMAX,CON6,XINT,XLO,PINT,PLO                                    MOD1111     33
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)                       P            2
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL        2
     1  PG(100), TG(100),  PMOUT(100), GL(100), DDRAG(100), FRICT(100), BARRL        3
     2  DCONV(100), DUP(100), DPHI(100), DRHOG(100), DHG(100), DUG(100), BARRL       4
     3  AMASS(100), AMOM(100), AENER(100), DAMASS(100), DAMOM(100),     BARRL        5
     4  DAENER(100)                                                     BARRL        6
      LOGICAL PRI1,IDEBUG                                               PLOG         2
      LOGICAL NXGE2                                                     MOTION      13
      DATA NXGE2/.FALSE./                                               MOTION      14
      DATA BPRES/2116./                                                 MOD1111     34
      DATA CSPD/1100./                                                  MOD1111     35
      DATA DELO/0./                                                     MOD1111     36
C                                                                       MOTION      15
      DATA XJOUL/778./                                                  MOTION      16
C                                                                       MOTION      17
C     DETERMINE PROPULSIVE FORCE ACTING ON THE PROJECTILE               MOTION      18
       F = PG(NX)*AREAPB                                                MOTION      19
C                                                                       MOTION      20
C     DETERMINE FOPRIM, THE ENGRAVING FORCE AND SLIDING RESISTANCE      MOTION      21
C     XOB IS THE INITIAL PROJECTILE POSITION--POSITION WHERE ENGRAVING  MOTION      22
C     BEGINS.  WOB IS THE LENGTH OF THE ENGRAVING BAND.                 MOTION      23
      TEMP = XP - XOB                                                   MOTION      24
      IF(TEMP .GT. WOB) GO TO 30                                        MOTION      25
      FOPRIM = Z0 + CON3*TEMP                                           MOTION      26
      GO TO 50                                                          MOD1111     37
C                                                                       MOTION      28
   30 IF(XP .GT. (XINT-XOB)) GO TO 40                                   MOD1111     38
      FOPRIM=FOMAX-CON4*(TEMP-WOB)                                      MOD1111     39
      GO TO 50                                                          MOD1111     40
   40 CONTINUE                                                          MOD1111     41
      FOPRIM=PINT*AREAPB-CON6*(TEMP-XINT)                               MOD1111     42
   45 DELC=0.2*DELO                                                     MOD1111     43
      CSPD=CSPD + DELC                                                  MOD1111     44
      DELP= 1.4*BPRES*DELO/(CSPD-0.6*DELO)                              MOD1111     45
      BPRES = BPRES + DELP                                              MOD1111     46
       FPRIM=BPRES*AREAPB                                               MOD1111     47
      IF(FOPRIM .LT. FPRIM) FOPRIM=FPRIM                                MOD1111     48
   50 CONTINUE                                                          MOD1111     49
C                                                                       MOTION      32
C     DETERMINE PROJECTILE AXIAL ACCELERATION                           MOTION      33
      ACC = CON2*(F*CON1 - FOPRIM)                                      MOTION      34
C                                                                       MOTION      35
C     IF THERE IS NO MOVEMENT, NX AND DXPRIM REMAIN THE SAME.           MOTION      36
C****IF ACC IS USED ELSEWHERE, MAKE IT NONNEGATIVE BEFORE RETURNING     MOTION      37
```

```
      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 = ROTR*VP*9.5493                                           MOTION     45
C                                                                      MOTION     46
C     PROJECTILE AXIAL POSITION                                        MOTION     47
      XP = XP + VP0*DELT + ACC*CON5                                    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
      XNG = XP/DX + 1.0                                                MOTION     55
C                                                                      MOTION     56
C     NUMBER OF TRUE GRIDS                                             MOTION     57
      NX = XNG                                                         MOTION     58
C                                                                      MOTION     59
C                                                                      MOD0301    70
C                                                                      MOD0301    71
C      SAVE THE PREVIOUS DXPRIM                                        MOD0301    72
      DXPS = DXPRIM                                                    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
           NXGE2 = .TRUE.                                              MOTION     72
           UHG(1) = HG(1)                                              MOTION     73
           URHOG(1) = RHOG(1)                                          MOTION     74
           UUG(1) = UG(1)                                              MOTION     75
 80    CONTINUE                                                        MOTION     76
       CALL GSPROP(R0,RRC,R,CV0,CVH,CV,PG(NGC),UHG(NGC),TOUM,URHOG(NGC),  MOTION     77
     $           UUG(NGC),0,0,GAM,CP,2)                                MOTION     78
      GAM1 = GAM - 1.0                                                 MOTION     79
      DELVP = VP - VP0                                                 MOTION     80
      C = SQRT(GAM1*(GJ*UHG(NGC) - VP0*VP0*0.5))                       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
C                                                                       MOTION    91
      TEMP = VP*VP/TWOGJ                                                MOTION    92
C                                                                       MOD1002   15
      UHG(NX) = CPRIM*CPRIM/(GAM1*GJ) + TEMP                            MOTION    93
C                                                                       MOTION    94
      URHOG(NX) = GAM/GAM1*PRES/(XJOUL*(UHG(NX) - TEMP))                MOTION    95
C                                                                       MOTION    96
      IF(NX .GT. NGC  .AND.  NX .EQ. 2) UUP(NX) = UP(1)                 MOD0301   78
      IF(NX .GT. NGC  .AND.  NX .GT. 2) UUP(NX) = UUP(NX - 1)           MOD0301   79
      IF(UUP(NX) .GT. UUG(NX)) UUP(NX) = UUG(NX)                        MOTION    98
      IF(NX .EQ. 1) GO TO 65                                            MOD0301   80
C****THE FOLLOWING MAY NOT BE NECESSARY                                 MOTION    99
      UAMASS(NX) = UAMASS(NX-1)                                         MOTION   100
      UAMOM(NX) = UAMOM(NX-1)                                           MOTION   101
      UAENER(NX) = UAENER(NX-1)                                         MOTION   102
C                                                                       MOTION   103
   65 CONTINUE                                                          MOD0301   81
C                                                                       MOD0301   82
      IF(NX .GT. NGC) GO TO 90                                          MOD0301   83
      UPHI(NX) = 1.0 - (1.0 - UPHI(NX))*(DXPS - DX*0.5)/                MOD0301   84
     *  (DXPRIM - DX*0.5)                                               MOD0301   85
      GO TO 100                                                         MOD0301   86
C                                                                       MOD0301   87
 90   CONTINUE                                                          MOD0301   88
      IF(NX .GT. 2) GO TO 95                                            MOD0301   89
      UPHI(NX) = PHI(1)                                                 MOD0301   90
      UDOB(NX) = DOB(1)                                                 MOD0301   91
      UDIB(NX) = DIB(1)                                                 MOD0301   92
      UXLB(NX) = XLB(1)                                                 MOD0301   93
      GO TO 100                                                         MOD0301   94
C                                                                       MOD0301   95
 95   CONTINUE                                                          MOD0301   96
      UPHI(NX) = 1.0 - (1.0 - UPHI(NGC))*(DXPS - DX*0.5)/               MOD0301   97
     *  (DXPRIM + DX*0.5)                                               MOD0301   98
      UDOB(NX) = UDOB(NX - 1)                                           MOD0301   99
      UDIB(NX) = UDIB(NX - 1)                                           MOD0301  100
      UXLB(NX) = UXLB(NX - 1)                                           MOD0301  101
C                                                                       MOD0301  102
C     IF A NEW GRID HAS BEEN ADDED, GRID NX - 1 DOES NOT HAVE THE PROPER MOD0301  103
C     VALUES EXCEPT FOR UAMASS, UAMOM, AND UAENER (IN BNDLYR AREAS      MOD0301  104
C     AT GRID NX ARE SET TO THOSE OF NX - 1).  IN THIS CASE NGC IS NOW  MOD0301  105
C     NX - 1.                                                           MOD0301  106
      FRACT = DX/(DX + DXPRIM)                                          MOD0301  107
      URHOG(NX - 1) = URHOG(NX - 2) + FRACT*(URHOG(NX) - URHOG(NX - 2)) MOD0301  108
      UUG(NX - 1) = UUG(NX - 2) + FRACT*(UUG(NX) - UUG(NX - 2))         MOD0301  109
      UHG(NX - 1) = UHG(NX - 2) + FRACT*(UHG(NX) - UHG(NX - 2))         MOD0301  110
      UUP(NX - 1) = UUP(NX - 2) + FRACT*(UUP(NX) - UUP(NX - 2))         MOD0301  111
      UPHI(NX - 1) = UPHI(NX)                                           MOD0301  112
C                                                                       MOTION   104
 100  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(6,5000) 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
     *  E14.8)                                                         MOTION     122
 4000 FORMAT(1H0,* TIME IS*,E14.8)                                     MOTION     123
      RETURN                                                           MOTION     124
      END                                                              MOTION     125
      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/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       UPDATE      7
     1 THICK(60),PENET,TYPCE                                            MOD0925   516
      COMMON/BARRL2/BOREA,XP,VP,BORED,BCHER,BORED2,DT2BD,DTDSQ,XLBAR    UPDATE      9
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX      2
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX      3
     2  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX      4
     3  AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX      5
      COMMON/CLOCK/TIME,DELT                                            UPDATE     11
      COMMON/IGNTE/PDRAD,BPCHG,BRNOUT,BGBRN,TMPBG,                      MOD1002    14
     + BEGTC(5),XLTC(5)                                                 MOD1002    15
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS        2
     +  DX,DR,NX,GJ,TWODT,HBP                                           EQNS        3
      COMMON/GASCON/RO,RHO,CVO,CVH                                      UPDATE     13
      COMMON/GRIDNX/DXPRIM                                              MOD0301   113
      COMMON/GASES/ROM6,RROM6,CVOM6,CVHM6,ROBP,RRCBP,CVOBP,CVHBP        UPDATE     14
      COMMON/BPT/TEP(2),DEP(2),CEP(2),FTUBR,PDALST                      MOD0925   517
      COMMON/PRMFLO/DOTMPM,UPRM,QPRM,PRMXP,FDSHR,CSPRKS,PRMCOF          MOD0925   518
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN       2
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,           GRAIN       3
     2  XLB(100), DOB(100), DIB(100), UXLB(100), UDOB(100), UDIB(100)   GRAIN       4
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,GLCNS,RHOF,PHIO,TF,CA,RHOO,     INPUTS      2
     1  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        INPUTS      3
      COMMON/P/IPRINT,MOUCH,MOUGR,PRI1,IDEBUG(35)                       P           2
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,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   DOTMIG(60), UBAG(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),CCTMP(60,5),PHIBP(60,5),      BAG         6
     5  PHIPTD(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),  PMDOT(100), GL(100), UDRAG(100), FRICT(100), BARRL       3
     2  QCONV(100), DUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100), BARRL      4
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),     BARRL       5
     4  UAENER(100)                                                     BARRL       6
      LOGICAL IGNIT,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,PMDOTS/0.0,0.0,0.0/                            UPDATE     25
      DATA  I24/0/                                                      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
      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.
      VAR = .TRUE.
C
      DO 100 I=1,NGX
         PRI2 = .FALSE.
         IF(MOD(I,MODGR) .EQ. 0  .OR.  I .EQ. 1  .OR.  I .EQ. NGX)
     $       PRI2 = .TRUE.
C
         DO 50 J=1,NGR
             PHIBG(I,J) = PHIBTD(I,J)
             PHIBP(I,J) = PHIPTD(I,J)
             RHOBG(I,J) = RHOBTD(I,J)
             UBG(I,J) = UBGTD(I,J)
             VBG(I,J) = VBGTD(I,J)
             HBG(I,J) = HBGTD(I,J)
             UPB(I,J) = UPBDT(I,J)
C
C     UPDATE TBG AND PCH BY CALLING GSPRCP WITH HBG AND RHOBG
             CALL GSPROP(RO,RHO,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),
     $            TBG(I,J),RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
C
             XL(I,J) = XLTDT(I,J)
             DO(I,J) = DOTDT(I,J)
             DI(I,J) = DITDT(I,J)
      IF(PCH(I,J).LE.0.0.OR.HBG(I,J).LE.0.0.OR.RHCBG(I,J).LE.0.0)
     *GO TO 10
C
             IF(.NOT. PRI1) GO TO 40
             IF(.NOT. IDEBUG(15)) GO TO 40
             IF(.NOT. PRI2) GO TO 40
   10 CONTINUE
             PRES = PCH(I,J)/144.
             WRITE(6,2001) I,J,PHIBG(I,J),RHOBG(I,J),UBG(I,J),VBG(I,J),
     1            HBG(I,J),TBG(I,J),PRES,PHIBP(I,J),UPB(I,J),TZC(I,J),
     2            UBAG(I,J),XDRAG(I,J),COMPC(I,J),DOTMP(I,J),TBP(I,J),
     3            XL(I,J),DO(I,J),DI(I,J)
 40          CONTINUE
C
 50      CONTINUE
C
C     POROSITY TEST
         IF(ONED) GO TO 100
         IF(.NOT. VAR) GO TO 100
      IF(I.LE.IBEGB)GO TO 100
         IF(I .GT. IENDB) GO TO 100
         IF(PHIBP(I,1) .LT. 0.999) VAR = .FALSE.
 100  CONTINUE
C
      NAMELIST/DOT/DOTMIG,DOTMBG,TZR
      IF(ONED) GO TO 110
      IF(IDEBUG(16)  .AND.  PRI1) WRITE(6,DOT)
 110  CONTINUE
C
C
C*********************************************************************
C     DETERMINE WHETHER THE CHAMBER SHOULD BE MADE 1-DIMENSIONAL
C*********************************************************************
```

```
C
      IF(UNED) GO TO 150
      IF(.NOT. VAR) GO TO 150
      IF(.NOT. IGNIT) GO TO 150
          UNED = .TRUE.
          CALL UNEDIM
          IF(IDEBUG(17)) WRITE(6,2006) TIME
          IF(IDEBUG(17)) WRITE(6,2009) IPRINT
          IF(.NOT. IDEBUG(18)) GO TO 150
          WRITE(6,2007)
          DO 130 I=1,NGX
              PRES = PBG(I,1)/144.
              WRITE(6,2008) I,PHIBG(I,1),RHOBG(I,1),UBG(I,1),HBG(I,1),
     $              TBG(I,1),PRES     ,UPB(I,1),XL(I,1),DO(I,1),DI(I,1)
 130      CONTINUE
C
 150  CONTINUE
C
C
C*****************************************************************
C     UPDATE BARREL ARRAYS
C*****************************************************************
C
      IF(.NOT. PRI1) GO TO 170
      IF(.NOT. IDEBUG(19)) GO TO 170
          WRITE(6,2004)
          I = 1
          PRES = PG(1)/144.
          WRITE(6,2005) I,PHI(I),RHOG(I),UG(I),HG(I),TG(I),PRES,UP(I),
     $         UL(I),UDRAG(I),PMDOT(I),AMASS(I),AMOM(I),AENER(I),
     $         XLB(I),DOB(I),DIB(I)
 170  CONTINUE
C
      IF(NX .LT. 2) GO TO 220
      DO 200 I=2,NX
          PHI(I) = UPHI(I)
          RHOG(I) = URHOG(I)
          UG(I) = UUG(I)
          HG(I) = UHG(I)
          UP(I) = UUP(I)
C
C     UPDATE TG AND PG BY CALLING GSPROP WITH HG AND RHOG
          CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PG(I),HG(I),TG(I),RHOG(I),
     *          UG(I),0.0,GAM,CP,2)
C
          XLB(I) = UXLB(I)
          DOB(I) = UDOB(I)
          DIB(I) = UDIB(I)
C
          AMASS(I) = UAMASS(I)
          AMOM(I) = UAMOM(I)
          AENER(I) = UAENER(I)
      IF(PG(I).LE.0.0.OR.HG(I).LE.0.0.OR.RHOG(I).LE.0.0) GO TO 160
C
          IF(.NOT. PRI1) GO TO 200
      IF(.NOT. IDEBUG(19)) GO TO 200
 160  CONTINUE
```

```
         PRES = PG(I)/144.
         WRITE(6,2005) I,PHI(I),RHOG(I),QG(I),HG(I),TG(I),PRES,
     $      UP(I),QL(I),DDRAG(I),PMDOT(I),AMASS(I),AMOM(I),AENER(I),
     1       XLB(I),DOF(I),DIB(I)
  200 CONTINUE
  220 CONTINUE
C
C***********************************************************************
C     PRINT OUT CERTAIN PRESSURES
C***********************************************************************
C
      IF(.NOT. PRI1) GO TO 240
      IF(.NOT. IDEFBG(22)) GO TO 240
      P1 = PCH(1,1)/144.
      WRITE(6,2014) P1
C
  240 CONTINUE
C
C
C***********************************************************************
C     COMPUTE MASS OF GAS IN THE SYSTEM AND THE MASS OF BLACK POWDER
C     AND PROPELLANT IN THE SYSTEM
C***********************************************************************
C
C     CALCULATIONS WILL BE DONE EACH TIME INTERVAL AND DENSITIES WILL BE
C     ADJUSTED IF MASS IS LOST.
C
C     LOGIC IS NOT WRITTEN FOR THE CASE WHERE CHAM1 IS TRUE
      IF(.NOT. CHAM1) GO TO 250
      WRITE(6,2010)
      GO TO 500
C
  250 CONTINUE
      IF(ONED) GO TO 300
      VOL = DVAXIS*0.5
      GASMAS = (PHIBG(1,1) + PHIBP(1,1) - 1.0)*RHOBG(1,1)*VOL
      PROMAS = ( (1.0 - PHIBG(1,1))*RHOP + (1.0 - PHIBP(1,1))*BPDENS )*
     $   VOL
      VOL = DVAXIS
      VOLROP = VOL*RHOP
      VOLBPD = VOL*BPDENS
      DO 260 I=2,NGX
         GASMAS = GASMAS + (PHIBG(I,1) + PHIBP(I,1) - 1.0)*RHOBG(I,1)*
     $      VOL
         PROMAS = PROMAS + (1.0 - PHIBG(I,1))*VOLROP +
     $       (1.0 - PHIBP(I,1))*VOLBPD
  260 CONTINUE
C
      VOL = AREAR(1)*DX*0.5
      GASMAS = GASMAS + (PHIBG(1,2) + PHIBP(1,2) - 1.0)*RHOBG(1,2)*VOL
      PROMAS = PROMAS + ( (1.0 - PHIBG(1,2))*RHOP +
     $     (1.0 - PHIBP(1,2))*BPDENS )*VOL
      DO 270 I=2,NGX
         VOL = AREAR(I)*DX
         GASMAS = GASMAS + (PHIBG(I,2) + PHIBP(I,2) - 1.0)*RHOBG(I,2)*
     $      VOL
         PROMAS = PROMAS + ( (1.0 - PHIBG(I,2))*RHOP +
     $        (1.0 - PHIBP(I,2))*BPDENS )*VOL
  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     CHAM3 IS TRUE)                                                    UPDATE     212
      VOL = AREAGP(1)*DX*0.5                                            UPDATE     213
      GASMAS = GASMAS + RHOBG(1,3)*VOL                                  UPDATE     214
      DO 280 I=2,NGX                                                    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)*RHOBG(1,1)*VOL           UPDATE     224
      PROMAS = ( (1.0 - PHIBG(1,1))*RHOP + (1.0 - PHIBP(1,1))*BPDENS )* UPDATE     225
     $     VOL                                                          UPDATE     226
      DO 310 I=2,NGX                                                    UPDATE     227
          VOL = AREAC(I)*DX                                             UPDATE     228
          GASMAS = GASMAS + (PHIBG(I,1) + PHIBP(I,1) - 1.0)*RHOBG(I,1)* UPDATE     229
     $         VOL                                                      UPDATE     230
          PROMAS = PROMAS + ( (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
          PROMAS = PROMAS + (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
      PROMAS = PROMAS + (1.0 - PHI(NX))*VOLROP                          MOD0301    117
C                                                                       UPDATE     250
  360 IF(PRI1 .AND. IDEBUG(20)) WRITE(6,2011) GASMAS,PROMAS             UPDATE     251
C                                                                       UPDATE     252
C     ADJUST DENSITIES                                                  UPDATE     253
      ACTGAS = TOTM - PROMAS                                            UPDATE     254
      ADJUST = ACTGAS/GASMAS                                            UPDATE     255
      DO 370 J=1,NGR                                                    UPDATE     256
      DO 370 I=1,NGX                                                    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                                                                         UPDATE      263
      IF(PRT1 .AND. IDEBUG(20)) WRITE(6,2013) ACTEAS,ADJUST               UPDATE      264
C                                                                         UPDATE      265
C                                                                         UPDATE      266
C************************************************************************ UPDATE      267
C     KEEP A RUNNING SUM OF DOTMB,DOTMP, ANC 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 SKIPPEC.                             UPDATE      273
  400  CONTINUE                                                           UPDATE      274
       IF(ONED) GO TO 420                                                 UPDATE      275
       DOTMBS = DOTMBS + (DOTMB(1,1)*DVAXIS + DOTME(1,2)*AREAR(1)*DX)*0.5 UPDATE      276
       DOTMPS = DOTMPS + (DOTMP(1,1)*DVAXIS + DOTMF(1,2)*AREAR(1)*DX)*0.5 UPDATE      277
       DO 410 I=2,NGX                                                     UPDATE      278
           DOTMBS = DOTMBS + DOTMB(I,1)*DVAXIS + DCTMB(I,2)*AREAR(I)*DX   UPDATE      279
           DOTMPS = DOTMPS + DOTMP(I,1)*DVAXIS + DCTMP(I,2)*AREAR(I)*DX   UPDATE      280
  410  CONTINUE                                                           UPDATE      281
       GO TO 440                                                          UPDATE      282
C                                                                         UPDATE      283
C     CALCULATIONS WHEN THE CHAMBER IS ONE DIMENSICNAL                    UPDATE      284
  420  CONTINUE                                                           UPDATE      285
       DOTMBS = DOTMBS + DOTMB(1,1)*AREAC(1)*DX*0.5                       UPDATE      286
       DOTMPS = DOTMPS + DOTMP(1,1)*AREAC(1)*DX*0.5                       UPDATE      287
       DO 430 I=2,NGX                                                     UPDATE      288
           DOTMBS = DOTMBS + DOTMB(I,1)*AREAC(I)*DX                       UPDATE      289
           DOTMPS = DOTMPS + DOTMP(I,1)*AREAC(I)*DX                       UPDATE      290
  430  CONTINUE                                                           UPDATE      291
C                                                                         UPDATE      292
C                                                                         UPDATE      293
C     BARREL CALCULATIONS                                                 UPDATE      294
  440  CONTINUE                                                           UPDATE      295
       IF(NX .EQ. 1) GO TO 470                                            UPDATE      296
       VOL = BOREA*DX                                                     UPDATE      297
       IF(NX .EQ. 2) GO TO 460                                            UPDATE      298
       NX1 = NX - 1                                                       UPDATE      299
       DO 450 I=2,NX1                                                     UPDATE      300
  450  PMDOTS = PMDOTS + PMDOT(I)*VOL                                     UPDATE      301
C                                                                         MOD0301     118
  460  PMDOTS = PMDOTS + PMDOT(NX)*BOREA*(DXPRIM - DX*0.5)                 MOD0301     119
C                                                                         UPDATE      303
  470  CONTINUE                                                           UPDATE      304
       PROPS = DOTMBS + PMDOTS                                            UPDATE      305
       DENOM = PROPS + DOTMPS                                             UPDATE      306
       IF(DENOM .LT. 0.0001) GO TO 480                                    UPDATE      307
       FRACM6 = PROPS/DENOM                                               UPDATE      308
       FRACBP = DOTMPS/DENOM                                              UPDATE      309
       RO = FRACM6*ROM6 + FRACBP*ROBP                                     UPDATE      310
       RRO = FRACM6*RROM6 + FRACBP*RROBP                                  UPDATE      311
       CVO = FRACM6*CVOM6 + FRACBP*CVOBP                                  UPDATE      312
       CVM = FRACM6*CVMM6 + FRACBP*CVMBP                                  UPDATE      313
C                                                                         UPDATE      314
  480  CONTINUE                                                           UPDATE      315
       IF(PRT1 .AND. IDEBUG(21)) WRITE(6,2012) DOTMBS,DOTMPS,PMDOTS,RO,   UPDATE      316
     $     RRO,CVO,CVM                                                    UPDATE      317
C************************************************************************ UPDATE      318
```

```
C     GET READY FOR THE NEXT TIME INTERVAL                                UPDATE     319
C*********************************************************************** UPDATE     320
C                                                                         UPDATE     321
  500  CONTINUE                                                           UPDATE     322
C                                                                         UPDATE     323
C     CLEAR ARRAYS DOTMIG AND DOTMBG FOR SUBROUTINE MFLOW, ARRAY DOTMP    UPDATE     324
C     FOR PRIMER, ARRAY GBAG FOR PRPFIR, ARRAY DOTMB FOR REGRES, AND      UPDATE     325
C     ARRAY UPBDT FOR PRPVEL.                                             UPDATE     326
       CALL CLEAR( DOTMIG(1),GBAG( 60,5))                                 UPDATE     327
       CALL CLEAR( DOTMB(1,1),UPBDT( 60,5))                               UPDATE     328
       CALL CLEAR(DOTMBG(1),DOTMP(60,5))                                  UPDATE     329
C                                                                         UPDATE     330
C                                                                         UPDATE     331
C     CLEAR ARRAY PMDOT FOR SUBROUTINE DIMIN, ARRAY QL FOR BNDLYR,        UPDATE     332
C     ARRAY UDRAG FOR RHOUH, AND ARRAY UGP FOR PROFMO.                    UPDATE     333
       CALL CLEAR(PMDOT(1), UGP(100))                                     UPDATE     334
       P11=PCH(1,1)/144.0                                                 UPDATE     335
       P12=PCH(1,2)/144.0                                                 UPDATE     336
       PN1=PCH(NGX,1)/144.0                                               UPDATE     337
       PN2=PCH(NGX,2)/144.0                                               UPDATE     338
       PGN=PG(NX)/144.0                                                   UPDATE     339
       XPPP=XP*12.0                                                       UPDATE     340
       I24=I24+1                                                          UPDATE     341
       I124=MOD(I24,10)                                                   UPDATE     342
       IF(IDEBUG(24).AND.I124.EQ.0)WRITE(6,2015)IPRINT,TIME,P11,          UPDATE     343
      1  P12,PN1,PN2,PGN,XPPP,VP                                          UPDATE     344
       IF(IDEBUG(34) .AND. TIME .LT. 0.005) WRITE(6,2016)                 MOD0605     28
      *  TIME,DOTMPM,GPMP,PRMXP,PDSHR                                     MOD0604      6
C                                                                         UPDATE     345
       RETURN                                                             UPDATE     346
C                                                                         UPDATE     347
C                                                                         UPDATE     348
C                                                                         UPDATE     349
C*********************************************************************** UPDATE     350
C                                                                         UPDATE     351
 2001 FORMAT(/,2I5,7(2X,E14.8),/,10X,7(2X,E14.8),/,                       UPDATE     352
     $  10X,4(2X,E14.8) )                                                 UPDATE     353
 2003 FORMAT(//,' IPRINT = ',I6,                                          UPDATE     354
     $          /,' TIME =',E14.8,//,3X,'I',4X,'J',6X,'PHIBG',11X,        UPDATE     355
     1  'RHOBG',12X,'UBG',13X,'VBG',13X,'HBG',13X,'TBG',13X,'PCH',        UPDATE     356
     2  /,15X,'PHIBP',11X,'UPB',13X,'TZC',13X,'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,'RHOG',13X,'LG',14X,'HG',14X,        UPDATE     359
     1    'TG',14X,'PG',14X,'UP',/,29X,'GL',12X,'UDRAG',11X,'PMDOT',      UPDATE     360
     2      11X,'AMASS',12X,'AMOM',11X,'AENER',                           UPDATE     361
     3    /,28X,'XLB',13X,'DOB',13X,'DIB')                                UPDATE     362
 2005 FORMAT(//,I5, 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-DIMENSIONA UPDATE     364
     $L')                                                                 UPDATE     365
 2007 FORMAT(////,3X,'I',6X,'PHIBG',11X,'RHOBG',12X,'UBG',13X,'HBG',      UPDATE     366
     $  13X,'TBG',13X,'PCH',13X,'UPB',/,28X,'XL',14X,'DO',14X,'DI')       UPDATE     367
 2008 FORMAT(/,I5,7(2X,E14.8),/,21X,3(2X,E14.8))                          UPDATE     368
 2009 FORMAT(//,' IPRINT = ',I5)                                          UPDATE     369
 2010 FORMAT('  LOGIC FOR SUMMING GAS MASS AND PROPELLANT MASS HAS NOT B UPDATE     370
     $EEN WRITTEN FOR CHAM1 TRUE')                                        UPDATE     371
 2011 FORMAT('  THE MASS OF GAS IN THE SYSTEM IS',F10.4,//,               UPDATE     372
     $  '  THE MASS OF PROPELLANT AND BLACK POWDER IN THE SYSTEM IS',     UPDATE     373
     $  F10.4)                                                          UPDATE     374
 2012 FORMAT(//,'  DUTMBS = ',F10.4,'  DUTPPS = ',F10.4,'  PMDOTS = ',  UPDATE     375
     $  F10.4,//,' FOR THE NEXT TIME INTERVAL  RO =',E10.4,'  RRO =',   UPDATE     376
     $  E10.4,'  CVU =',E10.4,'  CVM =',E10.4,/)                        UPDATE     377
 2013 FORMAT(' ACTUAL GAS IN THE SYSTEM IS', F10.4,                     UPDATE     378
     $  '  ADJUSTING FRACTION IS',F10.4,/)                              UPDATE     379
 2014 FORMAT(1H0,*PRESSURE AT GRID (1,1) IS*, E20.10)                   UPDATE     380
 2015 FORMAT(2X, I5, 3X, 8(E13.6,2X))                                   UPDATE     381
 2016 FORMAT(//,2X,5HTIME=,E14.8,3X,7HDCTMPF=,E14.5,                    MOD0611      6
     + 3X,5HGMPR=,E14.5,3X,6HPRMXP=,E14.5,3X,6HPDSHR=,E14.5)            MOD0611      7
      END                                                               UPDATE     382
      SUBROUTINE ONEDIM                                                 ONEDIM     2
C                                                                       ONEDIM     3
      COMMON/BARREL2/BOREA,XP,VP,BORED,BOREH,BORED2,DT2BD,DTDSQ,XLBAR   ONEDIM     4
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAM2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $  DX,DR,NX,GO,TWODT,HEP                                           EQNS       3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      ONEDIM     7
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTOT(60,5), DOTOT(60,5),  ITOT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN      4
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        INPUTS     3
      COMMON/PRIMV/BPLENS,BPRAD(60,5),AGENBP,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), TZC(60,5),                    BAG        3
     2   DOTMIG(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),DCTMP(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
C                                                                       ONEDIM    13
C     THIS SUBROUTINE PUTS THE CORRECT CHAMBER AREAS INTO ARRAY AREAC.  ONEDIM    14
C     PUTS VOLUME-WEIGHTED AVERAGES OF THE GAS ARRAYS INTO THE ARRAY    ONEDIM    15
C     ELEMENTS AT GRID 1. CLEARS THE GAS ARRAYS AT THE OTHER GRIDS.     ONEDIM    16
C     AND SETS NGR TO 1.                                                ONEDIM    17
C                                                                       ONEDIM    18
      DATA PIDF/.785398/                                                ONEDIM    19
C                                                                       ONEDIM    20
C                                                                       ONEDIM    21
C                                                                       ONEDIM    22
C********************************************************************** ONEDIM    23
C     FILL ARRAY AREAC                                                  ONEDIM    24
C********************************************************************** ONEDIM    25
C                                                                       ONEDIM    26
      IF(CHAM3) GO TO 30                                                ONEDIM    27
      DO 20 I=1,NGX                                                     ONEDIM    28
          AREAC(I) = AREAR(I) + AREAAX                                  ONEDIM    29
 20   CONTINUE                                                          ONEDIM    30
      GO TO 50                                                          ONEDIM    31
C                                                                       ONEDIM    32
 30   CONTINUE                                                          ONEDIM    33
      DO 40 I=1,NGX                                                     ONEDIM    34
          AREAC(I) = AREAR(I) + AREAAX + AREAGP(I)                      ONEDIM    35
 40   CONTINUE                                                          ONEDIM    36
C                                                                       ONEDIM    37
 50   AREAC(NGX+1) = BOREA                                              ONEDIM    38
C                                                                       ONEDIM    39
C                                                                       ONEDIM    40
C********************************************************************** ONEDIM    41
C     FILL EACH CHAMBER ARRAY AT (I,1) WITH THE VOLUME-WEIGHTED AVERAGE ONEDIM    42
C     OF THE ARRAY VALUES AT I.                                         ONEDIM    43
C********************************************************************** ONEDIM    44
C                                                                       ONEDIM    45
      IF(CHAM2) GO TO 400                                               ONEDIM    46
      IF(CHAM3) GO TO 500                                             ONEDIM    47
      DO 500 I=1,NGX                                                  ONEDIM    48
C                                                                     ONEDIM    49
C     MAKE TZC GREATER THAN TIGN SO THAT REGRES  CALCULATIONS ARE DONE. ONEDIM  50
         TZC(I,1) = TIGN + 1.0                                        ONEDIM    51
C                                                                     ONEDIM    52
         RHOBG(I,1) = RHOBG(I,1)*AREAG(1)                             ONEDIM    53
         HBG(I,1) = HBG(1,1)*AREAG(1)                                 ONEDIM    54
         UBG(I,1) = UBG(1,1)*AREAG(1)                                 ONEDIM    55
         PCH(I,1) = PCH(I,1)*AREAG(1)                                 ONEDIM    56
         TEMP = (1.0 - PHIBG(1,1))*AREAG(1)                           ONEDIM    57
         UPB(I,1) = UPB(I,1)*TEMP                                     ONEDIM    58
         XL(I,1) = XL(I,1)*TEMP                                       ONEDIM    59
         DO(I,1) = DO(I,1)*TEMP                                       ONEDIM    60
         DI(I,1) = DI(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
            RHOBG(I,1) = RHOBG(I,1) + RHOBG(I,J)*AREAG(J)             ONEDIM    68
            HBG(I,1) = HBG(1,1) + HBG(I,J)*AREAG(J)                   ONEDIM    69
            UBG(I,1) = UBG(1,1) + UBG(I,J)*AREAG(J)                   ONEDIM    70
            PCH(I,1) = PCH(1,1) + PCH(I,J)*AREAG(J)                   ONEDIM    71
         PHIBG(1,1) = PHIBG(1,1) + 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(1,1) = BPRAD(I,1) + BPRAD(I,J)*(1.0 - PHIBP(I,J))*   ONEDIM    78
     $          AREAG(J)                                              ONEDIM    79
           PHIBP(1,1) = PHIBP(I,1) + PHIBP(I,J)*AREAG(J)              ONEDIM    80
 250     CONTINUE                                                     ONEDIM    81
C                                                                     ONEDIM    82
         RHOBG(I,1) = RHOBG(I,1)/AREACH                               ONEDIM    83
         HBG(I,1) = HBG(1,1)/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(1,1))*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(1,1)/AREACH                               ONEDIM    97
         TEMP = (1.0 - PHIBG(1,1))*AREACH                             ONEDIM    98
      IF(TEMP .LT. .000001) GO TO 260                                 MOD1205   11
         UPB(I,1) = UPB(1,1)/TEMP                                     ONEDIM   100
         XL(I,1) = XL(I,1)/TEMP                                       ONEDIM   101
         DO(1,1) = DO(I,1)/TEMP                                       ONEDIM   102
         DI(I,1) = DI(I,1)/TEMP                                       ONEDIM   103
```

```
          GO TO 300                                                     ONEDIM   104
C                                                                       ONEDIM   105
 200      DPB(I,1) = 0.0                                                ONEDIM   106
          XL(I,1) = 0.0                                                 ONEDIM   107
          DO(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
      DO 450 I=1,NGX                                                    ONEDIM   118
          TZC(I,1) = TIGN + 1.0                                         ONEDIM   119
          PTEMP1 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAAX               ONEDIM   120
          PTEMP2 = (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)             ONEDIM   121
          BPRAD(I,1) = BPRAD(I,1)*(1.0 - PHIBP(I,1))*AREAAX +           ONEDIM   122
     $        BPRAD(I,2)*(1.0 - PHIBP(I,2))*AREAR(I)                    ONEDIM   123
          PHIBP(I,1) = (PHIBP(I,1)*AREAAX + 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))*AREAAX                             ONEDIM   135
          TEMP2 = (1.0 - PHIBG(I,2))*AREAR(I)                           ONEDIM   136
          PHIBG(I,1) = (PHIBG(I,1)*AREAAX + 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
          DPB(I,1) = (DPB(I,1)*TEMP1 + DPB(I,2)*TEMP2)/TEMP3            ONEDIM   141
          XL(I,1) = (XL(I,1)*TEMP1 + XL(I,2)*TEMP2)/TEMP3               ONEDIM   142
          DO(I,1) = (DO(I,1)*TEMP1 + DO(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. DO(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      DPB(I,1) = 0.0                                                ONEDIM   146
          XL(I,1) = 0.0                                                 ONEDIM   147
          DO(I,1) = 0.0                                                 ONEDIM   148
          DI(I,1) = 0.0                                                 ONEDIM   149
      IF(XL(I,1) .LE. 0. .OR. DO(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
          PTEMP3 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAC(I)             ONEDIM   152
C                                                                       ONEDIM   153
C****PTEMP3 SHOULD NEVER BE 0.0                                         ONEDIM   154
          RHOSAV = (RHOBG(I,1)*PTEMP1 + RHOBG(I,2)*PTEMP2)/PTEMP3       ONEDIM   155
          HBG(I,1) = (HBG(I,1)*PTEMP1*RHOBG(I,1) +                      ONEDIM   156
     $        HBG(I,2)*PTEMP2*RHOBG(I,2))/(PTEMP3*RHOSAV)
          UBG(I,1) = (UBG(I,1)*PTEMP1*RHOBG(I,1) +
     $        UBG(I,2)*PTEMP2*RHOBG(I,2))/(PTEMP3*RHOSAV)
          RHOBG(I,1) = RHOSAV
C
C     CALL GSPROP TO UPDATE PCH
          CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,1),HBG(I,1),TDUM,
     $        RHOBG(I,1),UBG(I,1),0.0,GAM,CP,2)
  450  CONTINUE
       GO TO 600
C
C
C
C     CALCULATIONS FOR CHAM3 TRUE
  500  CONTINUE
       DO 550 I=1,NGX
          TZC(I,1) = TIGN + 1.0
C
C     SAVE TOTAL POROSITY TIMES AREA
          PTEMP1 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAAX
          PTEMP2 = (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)
          PTEMP3 = AREAGP(I)
C
C     NOTE THAT (1.0 - PHIBP(I,3))*AREAGP(I) IS 0.0
          BPRAD(I,1) = BPRAD(I,1)*(1.0 - PHIBP(I,1))*AREAAX +
     $        BPRAD(I,2)*(1.0 - PHIBP(I,2))*AREAR(I)
          PHIBP(I,1) = (PHIBP(I,1)*AREAAX + PHIBP(I,2)*AREAR(I) +
     $        PHIBP(I,3)*AREAGP(I))/AREAC(I)
          TEMP = (1.0 - PHIBP(I,1))*AREAC(I)
       IF(TEMP .LT. .000001) GO TO 510
          BPRAD(I,1) = BPRAD(I,1)/TEMP
          GO TO 520
C
  510     CONTINUE
          BPRAD(I,1) = 0.0
C
  520     CONTINUE
          TEMP1 = (1.0 - PHIBG(I,1))*AREAAX
          TEMP2 = (1.0 - PHIBG(I,2))*AREAR(I)
C
C     NOTE THAT (1.0 - PHIBG(I,3))*AREAGP(I) IS 0.0
          PHIBG(I,1) = (PHIBG(I,1)*AREAAX + PHIBG(I,2)*AREAR(I) +
     $        PHIBG(I,3)*AREAGP(I))/AREAC(I)
          TEMP4 = (1.0 - PHIBG(I,1))*AREAC(I)
       IF(TEMP4 .LT. .000001) GO TO 530
          UPB(I,1) = (UPB(I,1)*TEMP1 + UPB(I,2)*TEMP2)/TEMP4
          XL(I,1) = (XL(I,1)*TEMP1 + XL(I,2)*TEMP2)/TEMP4
          DO(I,1) = (DO(I,1)*TEMP1 + DO(I,2)*TEMP2)/TEMP4
          DI(I,1) = (DI(I,1)*TEMP1 + DI(I,2)*TEMP2)/TEMP4
          GO TO 540
  530     UPB(I,1) = 0.0
          XL(I,1) = 0.0
          DO(I,1) = 0.0
          DI(I,1) = 0.0
C
  540 CONTINUE
          PTEMP4 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAC(I)
```

```
C
C****PTEMP4 SHOULD NOT BE 0.0
            RHOSAV = (RHOBG(I,1)*PTEMP1 + RHOBG(I,2)*PTEMP2 +
     $          RHOBG(I,3)*PTEMP3)/PTEMP4
            HBG(I,1) = (HBG(I,1)*PTEMP1*RHOBG(I,1) +
     $         HBG(I,2)*PTEMP2*RHOBG(I,2) + HBG(I,3)*PTEMP3*RHOBG(I,3))/
     $          (PTEMP4*RHOSAV)
            UBG(I,1) = (UBG(I,1)*PTEMP1*RHOBG(I,1) +
     $          UBG(I,2)*PTEMP2*RHOBG(I,2) + UBG(I,3)*PTEMP3*RHOBG(I,3))/
     $          (PTEMP4*RHOSAV)
            RHOBG(I,1) = RHOSAV
C
C     CALL GSPROP TO UPDATE PRESSURE
            CALL GSPROP(R0,RHO,R,CV0,CVH,CV,PCH(I,1),HBG(I,1),TDUM,
     $          RHOBG(I,1),UBG(I,1),0.0,GAM,CP,2)
  550 CONTINUE
C
C***********************************************************************
C     CLEAR ARRAY VBG AND OTHER CHAMBER ARRAYS AT GRIDS NOT ON THE AXIS
C***********************************************************************
C
  600 CONTINUE
      CALL CLEAR(VBG(1,1),VBG(60,5))
      CALL CLEAR(RHOBG(1,2),RHOBG(60,5))
      CALL CLEAR(PHIBG(1,2),PHIBG(60,5))
      CALL CLEAR(HBG(1,2),HBG(60,5))
      CALL CLEAR(UBG(1,2),UBG(60,5))
      CALL CLEAR(UPB(1,2),UPB(60,5))
      CALL CLEAR(PCH(1,2),PCH(60,5))
      CALL CLEAR(TBG(1,2),TBG(60,5))
      CALL CLEAR(XL(1,2),XL(60,5))
      CALL CLEAR(DO(1,2),DO(60,5))
      CALL CLEAR(DI(1,2),DI(60,5))
      CALL CLEAR(PHIBP(1,2),PHIBP(60,5))
      CALL CLEAR(BPRAD(1,2),BPRAD(60,5))
C
C
C
      NGR = 1
C
      RETURN
C
      END
      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 CUT 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/BARRL2/BOREA,XP,VP,BORED,BOREH,BORED2,DT2BD,DTDSQ,XLBAR    NEWDX        9
      COMMON/CAREAS/ARROW1,ARROW2,ARROW3,ARTOT                          MOD0301    120
      COMMON/GRIDNX/DXPRIM                                              MOD0301    121
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX       2
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX       3
     1  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMIX       4
     1  AREAH2,DIAMB1,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX       5
      COMMON/CLOCK/TIME,DELT                                            NEWDX       11
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     1  DX,DR,NX,GJ,TWODT,HBP                                           EQNS         3
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN        2
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,           GRAIN        3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), UDIB(100)   GRAIN        4
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60)        NEWDX       14
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,QCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS       2
     1  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        INPUTS       3
      COMMON/MOCON/CON1,CON2,CON3,CON4,CON5,AREAPE,ZO,WOB,XOB,ROTK,     NEWDX       16
     1  FDMAX,CON6,XINT,XLO,PINT,PLO                                    MOD1111     51
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,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), T2C(60,5),                    BAG          3
     2   DOTMIG(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),DOTMP(60,5),PHIBP(60,5),      BAG          6
     5  PHIPTD(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),  PMDOT(100), GL(100), UDRAG(100), FRICT(100), BARRL        3
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100), BARRL       4
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),     BARRL        5
     4  UAENER(100)                                                     BARRL        6
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG      2
      LOGICAL PRI1,IDEBUG                                               PLOG         2
      DATA GRAV/32.16/                                                  NEWDX       23
C                                                                       NEWDX       24
      NX = 11                                                           NEWDX       25
C                                                                       NEWDX       26
C     PUT THE BARREL ARRAY VALUES AT THE ODD-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
         URHOG(I) = URHOG(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
```

```
               DAMASS(I) = DAMASS(J)                                    NEWDX      41
               DAMOM(I) = DAMOM(J)                                      NEWDX      42
               DAENER(I) = DAENER(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 ODD-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
                    UPBOT(I,K) = UPBOT(J,K)                             NEWDX      67
                    XLTOT(I,K) = XLTOT(J,K)                             NEWDX      68
                    DOTOT(I,K) = DOTOT(J,K)                             NEWDX      69
                    DITOT(I,K) = DITOT(J,K)                             NEWDX      70
   30          CONTINUE                                                 NEWDX      71
                    AREAR(I) = AREAR(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*BOREA                                          MOD0301   124
          IF(ONED) GO TO 46                                             MOD0301   125
          IF(.NOT. CHAM2) GO TO 44                                      MOD0301   126
          SUM = AREAR(NGX) + AREAAX                                     MOD0301   127
          ARROW2 = ARROW2 + (AREAR(NGX)/SUM)*VLOST                      MOD0301   128
          ARROW1 = ARROW1 + (AREAAX/SUM)*VLOST                          MOD0301   129
          ARTOT = 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) + AREAAX                       MOD0301   135
          ARROW1 = ARROW1 + (AREAAX/SUM)*VLOST                          MOD0301   136
          ARROW2 = ARROW2 + (AREAR(NGX)/SUM)*VLOST                      MOD0301   137
          ARROW3 = ARROW3 + (AREAGP(NGX)/SUM)*VLOST                     MOD0301   138
          ARTOT = ARROW1 + ARROW2 + ARROW3                              MOD0301   139
          GO TO 50                                                      MOD0301   140
C                                                                       MOD0301   141
   46     CONTINUE                                                      MOD0301   142
          ARTOT = ARTOT + 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
       TWODT = 2.0*DELT                                                 NEWDX       80
       DTDX = DELT/DX                                                   NEWDX       81
       DTDR = DELT/DR                                                   NEWDX       82
       T2DR = 0.5*DTDR                                                  NEWDX       83
       T2DX = 0.5*DTDX                                                  NEWDX       84
       TWOTDR = 2.0*DTDR                                                NEWDX       85
C                                                                       NEWDX       86
C                                                                       NEWDX       87
       CON5 = 0.5*DELT*DELT                                             NEWDX       88
       DTDSQ = DELT*BORED*BORED                                         NEWDX       89
       DT2BD = -0.5*DELT/BORED                                          NEWDX       90
       GTRHOP = GRAV*DELT/RHOP                                          NEWDX       91
C                                                                       MOD0301    147
C      CALCULATE THE CHAMBER VOLUME.  IF IT HAS CHANGED ADJUST THE AREAS. MOD0301  148
       IF(ONED) GO TO 150                                               MOD0301    149
       ARR1 = (FLOAT(NGX) - 0.5)*DX*AREAAX                              MOD0301    150
C                                                                       MOD0301    151
       ARR2 = AREAR(1)*DX*0.5                                           MOD0301    152
       IF(NGX .EQ. 1) GO TO 80                                          MOD0301    153
       DO 70 I = 2,NGX                                                  MOD0301    154
       ARR2 = ARR2 + AREAR(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 = AREAGP(1)*DX*0.5                                          MOD0301    161
       IF(NGX .EQ. 1) GO TO 100                                         MOD0301    162
       DO 90 I = 2,NGX                                                  MOD0301    163
       ARR3 = ARR3 + AREAGP(I)*DX                                       MOD0301    164
   90  CONTINUE                                                         MOD0301    165
C                                                                       MOD0301    166
  100  CONTINUE                                                         MOD0301    167
C                                                                       MOD0301    168
C      ADJUST AREAAX AND ARRAY AREAC                                    MOD0301    169
       ADJUST = ARROW1/ARR1                                             MOD0301    170
       AREAAX = AREAAX*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 ARRAY AREAR                                               MOD0301    176
       ADJUST = ARROW2/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 ARRAY AREAGP                                              MOD0301    183
       ADJUST = ARROW3/ARR3                                             MOD0301    184
```

```
      DO 120 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
  150 CONTINUE                                                          MOD0301   192
      ART = AREAC(1)*DX*0.5                                             MOD0301   193
      IF(NGX .EQ. 1) GO TO 170                                          MOD0301   194
      DO 160 I = 2,NGX                                                  MOD0301   195
      ART = ART + AREAC(I)*DX                                           MOD0301   196
  160 CONTINUE                                                          MOD0301   197
C                                                                       MOD0301   198
  170 CONTINUE                                                          MOD0301   199
      ADJUST = ARTOT/ART                                                MOD0301   200
      DO 180 I = 1,NGX                                                  MOD0301   201
      AREAC(I) = AREAC(I)*ADJUST                                        MOD0301   202
  180 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
      DVAXIS = 3.141593*DR*DR*DX                                        NEWDX      95
      DVAXIT = 1.25*DVAXIS                                              NEWDX      96
      GO TO 220                                                         MOD0301   208
C                                                                       NEWDX      98
  210 CONTINUE                                                          MOD0301   209
      AREAG(2) = AREAR(NGX)                                             NEWDX     100
      IF(CHAM3) AREAG(3) = AREAGP(NGX)                                  NEWDX     101
      AREACH = AREAG(1) + AREAG(2)                                      NEWDX     102
      IF(CHAM3) AREACH = AREACH + AREAG(3)                              NEWDX     103
      DVAXIS = AREAAX*DX                                                NEWDX     104
C                                                                       NEWDX     105
  220 CONTINUE                                                          MOD0301   210
      IF(ONED) AREAC(NGP1) = BOREA                                      NEWDX     107
      AREAR(NGP1) = BOREA - AREAAX                                      NEWDX     108
      IF(CHAM3) AREAR(NGP1) = AREAR(NGP1) - AREAGF(NGP1)                NEWDX     109
      IF(IDEBUG(12)) WRITE(6,2000) IPRINT,NGX                           NEWDX     110
      NAMELIST/NEWCHK/DX,DELT,TWODT,DTDX,DTDR,T2DR,T2DX,TWOTDR,         NEWDX     111
     +  CONS,DTDSQ,DT2BD,GTRHOP,DVAXIS,DVAXIT,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) (AREAC(I),I=1,NGP1)                                 NEWDX     120
      WRITE(6,2005) (AREAG(I),I=1,NGR)                                  NEWDX     121
 2000 FORMAT('1 NEWDX CALLED, IPRINT =',I5,/,' NGX =',I5)               NEWDX     122
 2002 FORMAT(///,' ARRAY AREAR',//)                                     NEWDX     123
 2003 FORMAT(9X,10F11.7,/)                                              NEWDX     124
 2004 FORMAT(///,' ARRAY AREAC',//)                                     NEWDX     125
 2005 FORMAT(///,' ARRAY AREAG',//,20X,5F11.7)                          NEWDX     126
 2000 FORMAT(///,' ARRAY AREAGP',//)                                  NEWDX      127
      RETURN                                                          NEWDX      128
      END                                                             NEWDX      129
      SUBROUTINE AXIT2                                                  AXIT2        2
C     SUBROUTINE AXIT2 IS CALLED FOR GRIDS IN THE 2ND RADIAL ROW WHEN   AXIT2        3
C     THE CHAMBER HAS 2 SEPARATE ONE DIMENSIONAL ROWS.                  AXIT2        4
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           AXIT2        5
     1  UTDT,VBGAVE,VTDT                                                AXIT2        6
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBLGB,IENDB,IPATH(60,5),AREAG(5), CHAMI        2
     *   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI        3
     *  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,      CHAMI        4
     1  AREAR2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI        5
      COMMON/CLOCK/TIME,DELT                                            AXIT2        8
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS         2
     *  DX,DR,NX,GJ,TWODT,HBP                                           EQNS         3
      COMMON/GASCON/R0,RR0,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), T2C(60,5),                    BAG          3
     2   DOTMIG(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),DOTMP(60,5),PHIBP(60,5),      BAG          6
     5  PHIPTD(60,5),T2R(60),TBP(60,5)                                  BAG          7
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                        CHAMLOG      2
      DATA GRAV/32.16/                                                  AXIT2       13
C                                                                       AXIT2       14
C                                                                       AXIT2       15
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           AXIT2       16
     *      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      AXIT2       17
      BUGGER = (GAM - 1.0)/TWOGJ                                        AXIT2       18
      IP1 = I+1                                                         AXIT2       19
      IM1 = I - 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(I,J)                                                   AXIT2       25
      A5 = AREAR(I)                                                     AXIT2       26
      G2 = RHOBG(IM1,J)                                                 AXIT2       27
      G4 = RHOBG(IP1,J)                                                 AXIT2       28
      G5 = RHOBG(I,J)                                                   AXIT2       29
      E2 = G2*UBG(IM1,J)                                                AXIT2       30
      E4 = G4*UBG(IP1,J)                                                AXIT2       31
      E5 = G5*UBG(I,J)                                                  AXIT2       32
      H2 = PHIBG(IM1,J)*AREAR(IM1)*E2                                   AXIT2       33
      H4 = PHIBG(IP1,J)*AREAR(IP1)*E4                                   AXIT2       34
      C2 = G2 * HBG(IM1,J)                                              AXIT2       35
      C4 = G4 * HBG(IP1,J)                                              AXIT2       36
      C5 = G5 * HBG(I,J)                                                AXIT2       37
      B2 = BUGGER* UBG(IM1,J) * E2                                      AXIT2       38
      B4 = BUGGER * UBG(IP1,J) * E4                                     AXIT2       39
      B5 = BUGGER * UBG(I,J) * E5                                       AXIT2       40
      VOL = AREAR(I)*DX                                                 AXIT2       41
C                                                                       AXIT2       42
      PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25             AXIT2       43
      RHOAVE = (G2 + G5 + G5 + G4)*0.25                                 AXIT2       44
      UBGAVE = (UBG(IM1,J) + UBG(I,J) + UBG(I,J) + UBG(IP1,J))*0.25     AXIT2       45
      VBGAVE = 0.0                                                      AXIT2       46
      UPBAVE = (UPB(IM1,J) + UPB(I,J) + UPB(I,J) + UPB(IP1,J))*0.25     AXIT2       47
C                                                                       AXIT2       48
      PHITDT = PHIBTD(I,J) + PHIPTD(I,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
      TERM1=0.0                                                         AXIT2    54
      IF(DOTMIG(I).GT.0.0)TERM1=1.0                                     AXIT2    55
      TERM2=0.0                                                         AXIT2    56
      IF(DOTMBG(I).LT.0.0)TERM2=1.0                                     AXIT2    57
C                                                                       AXIT2    58
      UTDT = ( F5*(E2 + E5 + E5 + E4)*0.25                              AXIT2    59
     1         - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J)                    AXIT2    60
     2              + GRAV*AREAR(I)*PCH(IP1,J)                          AXIT2    61
     3              - GRAV*AREAR(I)*PCH(IP1,J))/A5                      AXIT2    62
     *  -DELT*(DOTMIG(I)*UBG(I,J)*TERM1-(DOTMBG(I)*UBG(I,J)*TERM2))/VOL AXIT2    63
     4         + DOTMB(I,J)*UPB(I,J) )/PHIRHO                           AXIT2    64
      IF (  ABS(UTDT).LT. 0.1  )UTDT = 0.0                              AXIT2    65
C                                                                       AXIT2    66
      VTDT = 0.0                                                        AXIT2    67
C                                                                       AXIT2    68
      IF(PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)              AXIT2    69
C                                                                       AXIT2    70
      HIGN = HBG(I,1)                                                   AXIT2    71
      IF ( DOTMIG(I) .GT. 0.00001 ) HIGN = HBG(I,2)                     AXIT2    72
      HBGN = HBG(I,2)                                                   AXIT2    73
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)                        AXIT2    74
C                                                                       AXIT2    75
      HBGTD(I,J) = GAM*( F5*(C2+B2+C5+C5+B5+B5+C4+B4)/(4.0*GAM)         AXIT2    76
     1         - T2DX*(H4*HBG(IP1,J) - H2*HBG(IM1,J))/A5                AXIT2    77
     2         + DELT*((DOTMBG(I)*HBGN - DOTMIG(I)*HIGN)/VOL            AXIT2    78
     3         - QBAG(I,J)) + DOTMB(I,J)*HMB + DOTMP(I,J)*HBP           AXIT2    79
     *  -XDRAG(I,J)*UPB(I,J)*DELT/788.                                  AXIT2    80
     4         - BUGGER*PHIRHO*UTDT*UTDT/GAM )/PHIRHO                   AXIT2    81
C                                                                       AXIT2    82
      RHOBTD(I,J) = RHOTDT                                              AXIT2    83
      UBGTD(I,J) = UTDT                                                 AXIT2    84
      VBGTD(I,J) = VTDT                                                 AXIT2    85
      RETURN                                                            AXIT2    86
      END                                                               AXIT2    87
      SUBROUTINE AXIT3
C     SUBROUTINE AXIT3 IS CALLED FOR GRIDS IN THE 2ND RADIAL ROW WHEN     AXIT3      2
C     THE CHAMBER HAS 3 SEPARATE ONE DIMENSIONAL ROWS.                    AXIT3      3
      COMMON/AVGDT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,             AXIT3      4
     $  UTOT,VBGAVE,VTOT                                                  AXIT3      5
      COMMON/CHAMI/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),  AXIT3      6
     +   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,     CHAMI      2
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,        CHAMI      3
     $  AREAH ,DIAMHT,BELENL,BEEBEG,IPS1,IPS2,RADFS,BPIGN                 CHAMI      4
      COMMON/CLOCK/TIME,DELT                                              CHAMI      5
      COMMON/EQNS/GTDX,T2DR,T2DX,TROTOR,DTCR,HMB,TWOGJ,DVAXIS,DVAXIT,     AXIT3      8
     $  DX,DR,NX,GO,TROUT,HBP                                             EQNS       2
      COMMON/GASCON/RO,RRO,CVO,CVR                                        EQNS       3
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),          AXIT3     10
     1    VBG(60,5), UPB(60,5), PCH(60,5), T2C(60,5),                     BAG        2
     2    DOTMIG(60), GBAG(60,5), DRAG(60,5), DOTMB(60,5), UPBDT(60,5),   BAG        3
     3    PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),           BAG        4
     4    VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),       BAG        5
     5   PHIPTD(60,5),T2R(60),TBP(60,5)                                   BAG        6
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                          BAG        7
      DATA GRAV/32.16/                                                    CHAMLOG    2
C                                                                         AXIT3     13
C                                                                         AXIT3     14
      CALL GSPROP(RO,RRO,R,CVO,CVR,CV,PCH(I,J),HBG(I,J),TDUM,             AXIT3     15
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                       AXIT3     16
      BUGGER = (GAM - 1.0)/TWOGJ                                          AXIT3     17
C                                                                         AXIT3     18
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST    AXIT3     19
C     POROSITY OF THE PROPELLANT.                                         AXIT3     20
      IP1 = I+1                                                           AXIT3     21
      IM1 = I - 1                                                         AXIT3     22
      F5 = PHIBG(I,J)                                                     AXIT3     23
      A5 = AREAGP(I)                                                      AXIT3     24
      G2 =     RHOBG(IM1,J)                                               AXIT3     25
      G4 =     RHOBG(IP1,J)                                               AXIT3     26
      G5 =     RHOBG(I,J)                                                 AXIT3     27
      E2 = G2*UBG(IM1,J)                                                  AXIT3     28
      E4 = G4*UBG(IP1,J)                                                  AXIT3     29
      E5 = G5*UBG(I,J)                                                    AXIT3     30
      H2 = PHIBG(IM1,J)*AREAGP(IM1)*E2                                    AXIT3     31
      H4 = PHIBG(IP1,J)*AREAGP(IP1)*E4                                    AXIT3     32
      C2 = G2*HBG(IM1,J)                                                  AXIT3     33
      C4 = G4*HBG(IP1,J)                                                  AXIT3     34
      C5 = G5*HBG(I,J)                                                    AXIT3     35
      B2 = BUGGER*UBG(IM1,J)*E2                                           AXIT3     36
      B4 = BUGGER*UBG(IP1,J)*E4                                           AXIT3     37
      B5 = BUGGER*UBG(I,J)*E5                                             AXIT3     38
C                                                                         AXIT3     39
      VOL = AREAGP(I)*DX                                                  AXIT3     40
C                                                                         AXIT3     41
C                                                                         AXIT3     42
      PHITOT = PHIBTD(I,J) + PHIPTD(I,J) - 1.0                            AXIT3     43
      RHOTOT = (F5*(G2 + G5 + G5 + G4 )*0.25                              AXIT3     44
     1           - T2DX*(H4 - H2)/ A5                                     AXIT3     45
     2           - DELT*DOTMBG(I)/ VOL)/ PHITOT                           AXIT3     46
      PHIRHO = PHITOT*RHOTOT                                              AXIT3     47
C                                                                         AXIT3     48
                                                                          AXIT3     49
      UTDT = (F5*(E2 + F5 + E5 + E4)*0.25                                AXIT3
     1         - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J)                     AXIT3
     2         + GRAV*A5*PCH(IP1,J) - GRAV*A5*PCH(IP1,J))/A5 )/PHIRHO    AXIT3
      IF( ABS(UTDT) .LT. 0.1 ) UTDT = 0.0                                AXIT3
C                                                                        AXIT3
      HBGN = HBG(I,2)                                                    AXIT3
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)                         AXIT3
      HBGTD(I,J) = GAM*( F5*(C2+B2+C4+B4+C5+B5+C5+B5)/(  4,0*GAM)        AXIT3
     3         -T2DX*( H4*HBG(IP1,J) - H2*HBG(IM1,J) )/A5                AXIT3
     4         - DELT*DOTMBG(I)*HBGN/ VOL                                AXIT3
     5         - BUGGER*UTDT*UTDT*PHIRHO/GAM )/PHIRHO                    AXIT3
C                                                                        AXIT3
      RHOBTD(I,J) = RHOTDT                                               AXIT3
      UBGTD(I,J) = UTDT                                                  AXIT3
      RETURN                                                             AXIT3
      END                                                                AXIT3
      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/FRMFLO/DOTMPM, QFRM                                        BSURA2         7
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,           BSURA2         8
     $  QTDT,VBGAVE,VTDT                                                BSURA2         9
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMI          2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMI          3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMI          4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMI          5
      COMMON/CLOCK/TIME,DELT                                            BSURA2        11
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QLCONS,RHOF,PHIO,TF,CA,RHOO,    MOD0925      519
     $  H0,P0,U0,GTRHOP,HM,DM,TIGNBP,QBCONS,TOTM                        MOD0925      520
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),       MOD0925      521
     1 THICK(60),PENET,TYFCL                                            MOD0925      522
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS           2
     $  DX,DR,NX,GJ,TWODT,HBP                                           EQNS           3
      COMMON/GASCON/R0,RR0,CV0,CVH                                      BSURA2        13
      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   DOTMIG(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),CCTMP(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
      LOGICAL PENET                                                     MOD0925      523
C                                                                       MOD0925      524
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,           BSURA2        17
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      BSURA2        18
      BUGGER = (GAM - 1.0)/TWOGJ                                        BSURA2        19
      IP1 = I+1                                                         BSURA2        20
C                                                                       BSURA2        21
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST  BSURA2        22
C     POROSITY OF THE PROPELLANT.                                       BSURA2        23
      F4 = PHIBG(IP1,J)                                                 BSURA2        24
      F5 = PHIBG(I,J)                                                   BSURA2        25
      G4 = F4*RHOBG(IP1,J)                                              BSURA2        26
      G5 = F5*RHOBG(I,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(I,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
      RHOAVE = (G4 + G5)/DENOM                                          BSURA2        36
      UBGAVE = 0.0                                                      BSURA2        37
      UPBAVE = (UPB(IP1,J) + UPB(I,J))*0.5                              BSURA2        38
C                                                                       BSURA2        39
      PHITDT = PHIBTD(I,J) + PHIPTD(I,J) - 1.0                          BSURA2        40
      RHOTDT=(F5*RHOAVE-DTDX*E4 + DELT * (DOTMIG(I) + DOTMPM)/          BSURA2        41
     1         DVAXIS*2.0 + DOTMB(I,J) + DOTPB(I,J))/PHITDT             BSURA2        42
      PHIRHO = PHITDT*RHOTDT                                            BSURA2        43
C                                                                       BSURA2        44
```

```
      UTDT = 0.0                                                        BSURA2      45
      VTDT = 0.0                                                        BSURA2      46
C                                                                       BSURA2      47
      HIGN = HBG(I,1)                                                   BSURA2      48
C                                                                       BSURA2      49
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)                        BSURA2      50
      HBGTD(I,J) = GAM*( F5*(C4+B4+C5+B5)/(DENOM*GAM)                   BSURA2      51
     1            - DTDX*C4*UBG(IP1,J)                                  BSURA2      52
     2         + DELT * ((DOTMIG(I) * HIGN + DOTMPP * HBP) * 2.0/       BSURA2      53
     3         DVAXIS - OBAG(I,J)) + DOTPB(I,J) * HMB                   BSURA2      54
     4         + DOTMP(I,J) * HBP)/PHIRHC                               BSURA2      55
C                                                                       BSURA2      56
      UTDT=2.0* DELT * DOTHPM * UPRM/RHCTOT/DVAXIS                      BSURA2      57
      RHOBTD(I,J) = RHOTOT                                              BSURA2      58
      UBGTD(I,J) = UTDT                                                 BSURA2      59
      VBGTD(I,J) = VTDT                                                 BSURA2      60
      RETURN                                                            BSURA2      61
      END                                                               BSURA2      62
      SUBROUTINE BSURT2
      COMMON/AVGDT/RHOTOT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,            BSURT2        2
     * UTOT,VBGAVE,VTOT                                                  BSURT2        3
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),  BSURT2        4
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,    CHAMI         2
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,       CHAMI         3
     +  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN                CHAMI         4
      COMMON/CLOCK/TIME,DELT                                             CHAMI         5
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,    BSURT2        6
     +  DX,DR,NX,GO,TWODT,HBP                                            EQNS          2
      COMMON/GASCON/RO,RRO,CVO,CVH                                       EQNS          3
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),         BSURT2        8
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),                     BAG           2
     .   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTPB(60,5), UPBDT(60,5),  BAG           3
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),           BAG           4
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),       BAG           5
     5  PHIPTD(60,5),TZR(60),TBP(60,5)                                   BAG           6
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN                         BAG           7
C                                                                        CHAMLOG       2
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,            BSURT2       11
     $        RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                     BSURT2       12
      BUGGER = (GAM - 1.0)/TWOGJ                                         BSURT2       13
      IP1 = I+1                                                          BSURT2       14
C                                                                        BSURT2       15
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST   BSURT2       16
C     POROSITY OF THE PROPELLANT.                                        BSURT2       17
      F4 = PHIBG(IP1,J)*AREAR(IP1)                                       BSURT2       18
      F5 = PHIBG(I,J)*AREAR(I)                                           BSURT2       19
      G4 = F4*RHOBG(IP1,J)                                               BSURT2       20
      G5 = F5*RHOBG(I,J)                                                 BSURT2       21
      E4 = G4*UBG(IP1,J)                                                 BSURT2       22
      E5 = 0.0                                                           BSURT2       23
      C4 = G4*HBG(IP1,J)                                                 BSURT2       24
      C5 = G5*HBG(I,J)                                                   BSURT2       25
      B4 = G4*BUGGER*UBG(IP1,J)**2                                       BSURT2       26
      B5 = 0.0                                                           BSURT2       27
C                                                                        BSURT2       28
      DENOM = F4 + F5                                                    BSURT2       29
      PHIAVE = DENOM/(AREAR(IP1) + AREAR(I))                             BSURT2       30
      RHOAVE = (G4 + G5)/DENOM                                           BSURT2       31
      UBGAVE = 0.0                                                       BSURT2       32
      UPBAVE = (UPB(IP1,J) + UPB(I,J))*0.5                               BSURT2       33
C                                                                        BSURT2       34
      PHITOT = (PHIBTD(I,J) + PHIPTD(I,J) - 1.0)*AREAR(I)                BSURT2       35
C                                                                        BSURT2       36
      RHOTOT = ( F5*RHOAVE - DTDX*E4 - DTDX*(DOTMIG(I) - DOTMBG(I) )     BSURT2       37
     $        + (DOTMB(I,J) + DOTMP(I,J))*AREAR(I) )/PHITOT              BSURT2       38
C                                                                        BSURT2       39
      PHIRHO = PHITOT*RHOTOT                                             BSURT2       40
C                                                                        BSURT2       41
      UTOT = 0.0                                                         BSURT2       42
      VTOT = 0.0                                                         BSURT2       43
C                                                                        BSURT2       44
      HIGN = HBG(I,1)                                                    BSURT2       45
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)                         BSURT2       46
      HBGN = HBG(I,2)                                                    BSURT2       47
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)                         BSURT2       48
                                                                         BSURT2       49
```

```
C                                                                BSURT2     50
      HBGTD(I,J) = GAM*( F5*(C4+B4+C5+B5)/(DENOM*GAM)            BSURT2     51
     1          - DTDX*(C4*UBG(IP1,J) + DCTMIG(I)*HIGN - DOTMBG(I)*HBGN)   BSURT2     52
     2               + AREAR(I)*(DOTMB(I,J)*HPB + DCTMP(I,J)*HBP  BSURT2     53
     3               - DELT*GBAG(I,J)) )/PHIRHO                   BSURT2     54
      RHOBTD(I,J) = RHOTDT                                       BSURT2     55
      UBGTD(I,J) = UTDT                                          BSURT2     56
      VBGTD(I,J) = VTDT                                          BSURT2     57
      RETURN                                                     BSURT2     58
      END                                                        BSURT2     59
      SUBROUTINE BSURT3                                                  BSURT3      2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,            BSURT3      3
     $  UTDT,VBGAVE,VTDT                                                 BSURT3      4
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IEDGE,IENDB,IPATH(60,5),AREAG(5),  CHAMI       2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,    CHAMI       3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,IOPGAP,AREAGP(60),DAVG,       CHAMI       4
     $  AREAH2,DIAMBT,BELEND,DELBEG,IPS1,IPS2,RADFS,BPIGN                CHAMI       5
      COMMON/CLOCK/TIME,DELT                                             BSURT3      6
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,    EQNS        2
     $  DX,DR,NX,GJ,TWODT,HBP                                            EQNS        3
      COMMON/GASCON/R0,RR0,CV0,CVH                                       BSURT3      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), TZC(60,5),                     BAG         3
     2   DOTMIG(60), DBAG(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),CDTMP(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
C                                                                        BSURT3     11
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,            BSURT3     12
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)                      BSURT3     13
      BUGGER = (GAM - 1.0)/TWOGJ                                         BSURT3     14
      IP1 = I+1                                                          BSURT3     15
C                                                                        BSURT3     16
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST   BSURT3     17
C     POROSITY OF THE PROPELLANT.                                        BSURT3     18
      F4 = PHIBG(IP1,J)*AREAGP(IP1)                                      BSURT3     19
      F5 = PHIBG(I,J)*AREAGP(I)                                          BSURT3     20
      G4 = F4*RHOBG(IP1,J)                                               BSURT3     21
      G5 = F5*RHOBG(I,J)                                                 BSURT3     22
      E4 = G4*UBG(IP1,J)                                                 BSURT3     23
      E5 = 0.0                                                           BSURT3     24
      C4 = G4*HBG(IP1,J)                                                 BSURT3     25
      C5 = G5*HBG(I,J)                                                   BSURT3     26
      B4 = G4*BUGGER*UBG(IP1,J)**2                                       BSURT3     27
      B5 = 0.0                                                           BSURT3     28
C                                                                        BSURT3     29
      DENOM = F4 + F5                                                    BSURT3     30
C     PHIAVE = DENOM/(AREAGP(IP1) + AREAGP(I))                           BSURT3     31
      RHOAVE = (G4 + G5)/DENOM                                           BSURT3     32
C     UBGAVE = 0.0                                                       BSURT3     33
C     UPBAVE = (UPB(IP1,J) + UPB(I,J))*0.5                               BSURT3     34
C                                                                        BSURT3     35
      PHITDT = (PHIBTD(I,J) + PHIPTD(I,J) - 1.0)*AREAGP(I)               BSURT3     36
C                                                                        BSURT3     37
      RHOTDT = ( F5*RHOAVE - DTDX*E4 - CTDX*DOTMBG(I) )/PHITDT           BSURT3     38
C                                                                        BSURT3     39
      PHIRHO = PHITDT*RHOTDT                                             BSURT3     40
C                                                                        BSURT3     41
      UTDT = 0.0                                                         BSURT3     42
C                                                                        BSURT3     43
      HBGN = HBG(I,2)                                                    BSURT3     44
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)                         BSURT3     45
C                                                                        BSURT3     46
      HBGTD(I,J) = GAM*( F5*(C4+B4+C5+B5)/(DENOM*GAM)                    BSURT3     47
     1              - DTDX*( C4*UBG(IP1,J) + DOTMBG(I)*HBGN ) )/PHIRHO   BSURT3     48
      RHOBTD(I,J) = RHOTDT                                               BSURT3     49
      DBGTD(I,J) = DTDT                                                 BSURT3      50
      RETURN                                                            BSURT3      51
      END                                                               BSURT3      52
      SUBROUTINE BNDLYR                                                 BNDLYR    2
C                                                                       BNDLYR    3
      COMMON/BARRL2/ACREA,XP,VP,BORED,ACREA,BOREDE,DT2BD,DTUSQ          BNDLYR    4
      COMMON/CONS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   CONS      2
     1  DX,DR,NX,GJ,TWODT,HBP                                           CONS      3
      COMMON/GASCON/RO,RRO,CVO,CVH                                      BNDLYR    6
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,GLCNS,RHOF,PHIO,TF,CA,RHOO,     INPUTS    2
     1  HO,PG,DO,GTRHOP,HW,DM,TIGNBP,QBCONS,TOTM                        INPUTS    3
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),     BARRL     2
     1  PG(100), TG(100),  FMDOT(100), GL(100), UDRAG(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), UAMASS(100), UAMOM(100),     BARRL     5
     4  UAENER(100)                                                     BARRL     6
      REAL MACHSQ                                                       BNDLYR    9
      DATA PIOF/.785398/                                                BNDLYR   10
C     SUBROUTINE BNDLYR CALCULATES THE THICKNESSES OF THE BOUNDARY LAYERS BNDLYR 11
C     FORMED AS PROPELLANT GAS FLOWS THROUGH THE BARREL.                BNDLYR   12
C             AMASS IS THE AREA AVAILABLE FOR MASS FLOW                 BNDLYR   13
C             AMOM IS THE AREA OUTSIDE THE MOMENTUM LAYER               BNDLYR   14
C             AENER IS THE AREA OUTSIDE THE ENERGY BOUNDARY LAYER       BNDLYR   15
C     IT ALSO CALCULATES QL, THE HEAT TRANSFERRED FROM THE GRID TO OR   BNDLYR   16
C     FROM THE BARREL.                                                  BNDLYR   17
C     ARRAY QL CONTAINS 0.0'S AT THE BEGINNING OF EACH TIME INTERVAL.   BNDLYR   18
C                                                                       BNDLYR   19
C                                                                       BNDLYR   20
      NM1 = NX - 1                                                      BNDLYR   21
C                                                                       BNDLYR   22
      SPDX = 0.0                                                        BNDLYR   23
      DO 60 I=2,NM1                                                     BNDLYR   24
         U =(UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25                          BNDLYR   25
         IF(U .LT. 10.) U = 10.                                         BNDLYR   26
         USQ = U*U                                                      BNDLYR   27
C                                                                       BNDLYR   28
C     STATIC ENTHALPY                                                   BNDLYR   29
         HSTAT = (HG(I)+HG(I)+HG(I-1)+HG(I+1))*0.25- .2E-4*USQ          BNDLYR   30
         CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PG(I),HSTAT,TSTAT,RHODUM,      BNDLYR   31
     1      0.0,0.0,GAM,CP,3)                                           BNDLYR   32
         GAM1 = GAM - 1.0                                               BNDLYR   33
         GAM3 = (3.0*GAM - 1.0)/(2.0*GAM1)                              BNDLYR   34
C                                                                       BNDLYR   35
C     VISCOSITY                                                         BNDLYR   36
         VIS = C1*TSTAT*SQRT(TSTAT)/(TSTAT + C2)                        BNDLYR   37
C                                                                       BNDLYR   38
C     REYNOLD'S NUMBER/FOOT                                             BNDLYR   39
         RE = (RHOG(I)+RHOG(I)+RHOG(I-1)+RHOG(I+1))*0.25*U/VIS          BNDLYR   40
C                                                                       BNDLYR   41
C     MACH NUMBER SQUARED                                               BNDLYR   42
         MACHSQ = USQ/(32.16*GAM*R*TSTAT)                               BNDLYR   43
C                                                                       BNDLYR   44
C     MACH NUMBER WEIGHTING FUNCTION                                    BNDLYR   45
         PRESX = MACHSQ*MACHSQ/((1.0 + 0.5*GAM1*MACHSQ)**GAM3)          BNDLYR   46
         PDX = PRESX*DX                                                 BNDLYR   47
         SPDX = SPDX + PDX                                              BNDLYR   48
C                                                                       BNDLYR   49
C     EQUIVALENT FLAT PLATE LENGTH                                      BNDLYR   50
         EXF = SPDX/PRESX                                               BNDLYR   51
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. BORER) 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
         THETA = 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. BORER) 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
        THETA = 0.036*GFUN1*REF                                         BNDLYR  82
C                                                                       BNDLYR  83
C       ENERGY BOUNDARY LAYER THICKNESS                                 BNDLYR  84
 40      DELDST = 1.269*DELSTR/(DELSTR/THETA - 0.379)                   BNDLYR  85
C                                                                       BNDLYR  86
C     MASS FLOW AREA                                                    BNDLYR  87
         UAMASS(I) =   PIDF*(BORED - 2.0*DELSTR)**2                     BNDLYR  88
C                                                                       BNDLYR  89
C     MOMENTUM FLOW AREA                                                BNDLYR  90
         UAMOM(I) =   PIDF*(BORED - 2.0*THETA)**2                       BNDLYR  91
C                                                                       BNDLYR  92
C     ENERGY FLOW AREA                                                  BNDLYR  93
         UAENER(I) =   PIDF*(BORED - 2.0*DELDST)**2                     BNDLYR  94
C                                                                       BNDLYR  95
C     AVERAGE (REFERENCE) ENTHALPY IN BOUNDARY LAYER                    BNDLYR  96
         HSTAR = 0.5*(HG(I) + HW) - 6.12ME-6*USQ                        BNDLYR  97
         CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PG(I),HSTAR,TSTAR,RHOSTR,      BNDLYR  98
     * 0.0,0.0,GAM,CP,3)                                                BNDLYR  99
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PDUM,HG(I),                       BNDLYR 100
     *  TDUM,RHOG(I),0.0.0,GAM,CP,2)                                    BNDLYR 101
         GAM1 = GAM - 1.0                                               BNDLYR 102
         MACHSQ = USQ/(32.16*GAM*R*TSTAR)                               BNDLYR 103
         VISTR = C1*TSTAR*SQRT(TSTAR)/(TSTAR + C2)                      BNDLYR 104
         RESTRX = RHOSTR*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
         QDOT = 0.0366*VISTR/EXF * RESTRX**0.8 * (HG(I) - HW)*CFR       BNDLYR 109
```

```
C
C     HEAT FLUX
            QC(I) = 3.141593*BORED*DX*QDOT
C
C
  60  CONTINUE
      GO TO 200
C
C
C
C
 100  CONTINUE
      DO 110 I=2,NM1
            UAMASS(I) = BOREA
            UAMOM(I) = BOREA
            UAENER(I) = BOREA
 110  CONTINUE
      GO TO 200
C
C
 150  CONTINUE
      I = ITEMP
      I1 = I - 1
      UAMASS(I) = UAMASS(I1)
      UAMOM(I) = UAMOM(I1)
      UAENER(I) = UAENER(I1)
C
      RED = RE*BORED
      EPSLON = 0.0005*ALOG(RED) - 0.00556
      IF(EPSLON .LT. 0.0001) EPSLON = 0.0
C
      LAMBDA = 0.3164/RED**0.25 + EPSLON
C
      RHOU = (RHOG(I) + RHOG(I) + RHOG(I-1) + RHOG(I+1))*0.25*U
      LAMU = LAMBDA*U
C
C    BORED8 IS BORED/8.
      FRICT(I) = LAMU*RHOU*BORED8
C
C    DT2BD IS -0.5*DELT/BORED
C    DTDSQ IS DELT *BORED*BORED
      QCONV(I) = PIDF*((HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 - HW)
     1         *(1.0 - EXP(LAMU*DT2BD))*RHOU*DTDSQ
C
C
      I2 = I+1
C
      DO 160 I=2,NM1
          UAMASS(I) = UAMASS(I1)
          UAMOM(I) = UAMOM(I1)
          UAENER(I) = UAENER(I1)
C
      U=(UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25
          IF(U .LT. 10.) U = 10.
          USQ = U*U
          HSTAT = (HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 - .2E-4*USQ
          CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PG(I),HSTAT,TSTAT,RHODUM,
     $             0.0,0.0,GAM,CP,3)
          VIS = C1*TSTAT*SQRT(TSTAT)/(TSTAT + C2)
          RHOU = (RHOG(I) + RHOG(I) + RHOG(I-1) + RHOG(I+1))*0.25*U
          RED = RHOU/VIS*BORED
          EPSLON = 0.0005*ALOG(RED) - 0.00536
          IF(EPSLON .LT. 0.0001) EPSLON = 0.0
C
          LAMBDA = 0.3164/RED**0.25 + EPSLON
          LAMU = LAMBDA*U
C
          FRICT(I) = LAMU*RHOU*BORED8
          QCONV(I) = PIDF*((HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 -HW)
     $       *(1.0 - EXP(LAMU*DT2BD))*RHOL*DTDSQ
  160 CONTINUE
C
C
  200 CONTINUE
      DO 220 I=2,NM1
          UAMASS(I) = (4.0*AMASS(I) + UAMASS(I))/5.0
          UAMOM(I) = (4.0*AMOM(I) + UAMOM(I))/5.0
          UAENER(I) = (4.0*AENER(I) + UAENER(I))/5.0
  220 CONTINUE
C
C     LET AREAS AT THE BASE OF THE PROJECTILE BE THE SAME AS THOSE AT THE
C     GRID IMMEDIATELY PRECEDING.
      UAMASS(NX) = UAMASS(NM1)
      UAMOM(NX) = UAMOM(NM1)
      UAENER(NX) = UAENER(NM1)
      RETURN
C
      END
      SUBROUTINE TUBFAL                                                 TUBFAL     2
      COMMON/CLOCK/TIME,DELT                                            TUBFAL     3
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,   EQNS       2
     $ DX,DR,NX,GG,TWODT,HBP                                            EQNS       3
      COMMON/HPT/TEP(2),DEP(2),CEP(2),FTUBR,PDALST                      MOD0925  528
      COMMON/PRMFLO/DOTMPM,DPRM,QPRM,PRPXF,FDSHR,CSPRKS,PRMCOF          MOD0925  529
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,                  GRAIN      2
     1 XLTOT(60,5), DOTOT(60,5), DITOT(60,5), XLO,  DOO, DIO,           GRAIN      3
     2  XLB(100), DOB(100), DIB(100), DXLB(100), DDOB(100), DDIB(100)   GRAIN      4
      COMMON/FAILED/THICKT,PHOOP,PCOMP,ETUB,XNTUB,FAIL,MFAIL(60),       TUBFAL     7
     1 THICK(60),PENET,TYPCE                                            MOD0925  530
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),        BAG        2
     1   VBG(60,5), UFB(60,5), PCH(60,5), TZC(60,5),                    BAG        3
     2   DOTMIG(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),DOTPP(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),TZR(60),THP(60,5)                                  BAG        7
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),       HOLEA      2
     1  AH(60),FRACT(60)                                                HOLEA      3
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5), CHAMIX     2
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADFS,BPIGN               CHAMIX     5
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,GCONS,RHOF,PHIO,TF,CA,RHOO,     INPUTS     2
     $  HO,PO,UO,GTRHOP,HW,DM,TIGNBP,GBCONS,TOTM                        INPUTS     3
C  THIS SUBROUTINE CALCULATES FAILURE OF THE CENTER CORE IGNITER TUBE   TUBFAL    13
C  DUE TO COMPRESSIVE OR TENSILE (HOOP) PRESSURE LOADING.               TUBFAL    14
C  INCLUDED IS STRENGTH DEGRADATION DUE TO TUBE COMBUSTION              TUBFAL    15
C        MFAIL(I) - 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        PHOOP - TENSILE OR HOOP STRENGTH OF IGNITER TUBE               TUBFAL    29
C        THICKT - INITIAL IGNITER TUBE THICKNESS                        TUBFAL    30
      LOGICAL PENET                                                     MOD0605   31
      LOGICAL FAIL                                                      TUBFAL    31
      FAIL = .TRUE.                                                     TUBFAL    32
      DO 100 I=IBEGB,IENDB                                              TUBFAL    33
      IF(MFAIL(I).EQ.2) GO TO 100                                       TUBFAL    34
      IF(PHOOP.EQ.0.0.AND.PCOMP.EQ.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=PCH(I,1)-PCH(I,2)                                            TUBFAL    39
C  DELP IS POSITIVE FOR HOOP STRESS.                                    TUBFAL    40
C  COMPUTE CURRENT TUBE STRENGTH AFTER IT HAS BEEN DEGRADED DUE TO BURNI TUBFAL   41
      DELTH1 = 0.0                                                      TUBFAL    42
      DELTH2 = 0.0                                                      TUBFAL    43
```

```
C   IF PROPELLANT IGNITION HAS OCCURRED,
C   ASSUME THE ADJACENT TUBE SURFACE IS ALSO IGNITED
      IF (TBP(I,1) .GE. TIGNBP .OR. TZC(I,1) .GE. TIGN) DELTR1 = BTUB *
     *PCM(I,1)**XNTUB*DELT
      IF (TZC(I,2) .GE. TIGN .OR. TZR(I)   .GE. TIGN) DELTR2 = BTUB *
     *PCM(I,2)**XNTUB*DELT
      THICK(I) = THICK(I) - DELTR1 - DELTR2
      RATIO = THICK(I) / THICKI
      PHOOPL = PHOOP * RATIO
      PCOMPL = PCOMP * RATIO
      IF (DELP .LT. PHOOPL .AND. DELP .GT. -PCOMPL .AND. MFAIL(I) .EQ. 0
     *) GO TO 100
   20 IF(MFAIL(I).EQ.0) AREAH(I) = 3.1416*(.75*DIAMBT*SQRT(DX))**2.
      IF(MFAIL(I).EQ.0.AND.I.EQ.1) AREAH(I)=AREAH(I)/4.0
      IF(PHOOP.EQ.0.0.AND.PCOMP.EQ.0.0) GO TO 40
      MFAIL(I) = 1
      IF (DELP .GT. -PCOMPL) GO TO 100
   40 MFAIL(I) = 2
      FI2 = 2.0 - PHIBG(I,2) - PHIBP(I,2)
      FI1 = 2.0 - PHIBG(I,1) - PHIBP(I,1)
      AREAI = AREAR(I) + AREAC(I)
      FIBGN2 = 1.0-((FI2*AREAR(I)+FI1*AREAC(I)-(1.0-PHIBP(I,2))*AREAI)/
     *AREAI)
      FIBGN1 = 1.0-((FI2*AREAR(I)+FI1*AREAC(I)-(1.0-PHIBP(I,1))*AREAI)/
     *AREAI)
      IF(FIBGN1.LE.1..AND.FIBGN2.LE.1.)GO TO 50
      FIBGN2=PHIBG(I,2)
      FIBGN1=PHIBG(I,1)
      GO TO 100
   50 CONTINUE
      RAF1=(PHIBG(I,1) + PHIBP(I,1)-1.0) * RHOBG(I,1) * AREAC(I)
      RAF2=(PHIBG(I,2) + PHIBP(I,2)-1.0) * RHOBG(I,2) * AREAR(I)
      AFN1 = (FIBGN1 + PHIBP(I,1) - 1.0) * AREAC(I)
      AFN2 = (FIBGN2 + PHIBP(I,2) - 1.0) * AREAR(I)
      RHOBGN = (RAF2 + RAF1)/(AFN1 + AFN2)
      UBGN = (RAF2 * UBG(I,2) + RAF1 * UBG(I,1))/
     1 (RHOBGN * (AFN1 + AFN2))
      HBGN = (RAF2 * HBG(I,2) + RAF1 * HBG(I,1))/
     1 (RHOBGN * (AFN1 + AFN2))
      APHI2 = AREAR(I) * (FIBGN2 - PHIBG(I,2))
      APHI1 = AREAC(I) * (1.0 - PHIBG(I,1))
      APHIN1 = AREAC(I) * (1.0 - FIBGN1)
      UPN1 = (UPB(I,2) * APHI2 + UPB(I,1) * APHI1)/APHIN1
      XLN1 = (XL(I,2) * APHI2 + XL(I,1) * APHI1)/APHIN1
      DON1 = (DO(I,2) * APHI2 + DO(I,1) * APHI1)/APHIN1
      DIN1 = (DI(I,2) * APHI2 + DI(I,1) * APHI1)/APHIN1
      PHIBG(I,1) = FIBGN1
      PHIBG(I,2) = FIBGN2
      RHOBG(I,1) = RHOBGN
      RHOBG(I,2) = RHOBGN
      UBG(I,1) = UBGN
      UBG(I,2) = UBGN
      HBG(I,1) = HBGN
      HBG(I,2) = HBGN
      UPB(I,1) = UPN1
      XL(I,1) = XLN1
      DO(I,1) = DON1
      DI(1,1) = DIN1                                                    TURFAL   101
      XLIDI(1,1)=XLN1                                                   TURFAL   102
      DOTDT(1,1)=DON1                                                   TURFAL   103
      DITDT(1,1)=DIN1                                                   TURFAL   104
      DPBDT(1,1)=DPN1                                                   TURFAL   105
      IZC(1,1)-IZC(I,2)                                                 TURFAL   106
  100 CONTINUE                                                          TURFAL   107
      RETURN                                                            TURFAL   108
      END                                                               TURFAL   109
      SUBROUTINE BPINIT                                                 BPINIT     2
      COMMON/IGNIF/PBRAD.BPCHG.BRNOUT.BGRRA.IFPBG,                      MOD0925  531
     $ BEGTC(5).XLTC(5)                                                 MOD0925  532
      COMMON/BPT/TEP(2).DEP(2).CEP(2).FTUBR.PDALST                      MOD0925  533
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBFGR,IENDR,IPATH(60,5),AREAG(5), CHAMIX     2
     1   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,   CHAMIX     3
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM4,IOPGAP,AREAGP(60),DAVG,      CHAMIX     4
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADES,BPIGN               CHAMIX     5
      COMMON/PRIMV/BPDENS,BFRAD(60,5),AGFNBP,BGFNEP,EXPBP               PRIMV      2
      COMMON/FQNS/TTDX,T2DR,T2DX,TWOTDR,DTCR,HMR,TWOGJ,DVAXIS,DVAXIT,   FQNS       2
     $  DX,DR,NX,GJ,TWODT,HBP                                           FQNS       3
      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   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(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),DOTMF(60,5),PHIBP(60,5),      BAG        6
     5  PHIPTD(60,5),TZX(60),TBP(60,5)                                  BAG        7
C  SUBROUTINE BPINIT CALCULATES INITIAL POROSITY                        BPINIT     8
C  VALUES AT EACH GRID THAT CONTAINS BLACK                              BPINIT     9
C  POWDER.  THE SUBROUTINE READS IN THE FOLLOWING PARAMETERS%           BPINIT    10
C                     TEP(2)-   THICKNESS OF ENDPADS 1 + 2 (INCHES)     BPINIT    11
C                     DEP(2)-   DIAMETER OF ENDPADS 1 + 2 (INCHES)      BPINIT    12
C                     CEP(2)-   CHARGE WEIGHT OF ENDPADS 1 + 2 (OZ.)    BPINIT    13
C                     ITUBE-    NUMBER OF TUBE CHARGES                  BPINIT    14
C                     BEGTC(5)- LOCATION OF THE TUBE CHARGES (INCHES)   BPINIT    15
C                     XLTC(5)-  LENGTH OF THE TUBE CHARGES (INCHES)     BPINIT    16
C                     CHTC(5)-  CHARGE WEIGHT OF THE TUBE CHARGES (OZ.) BPINIT    17
C                                                                       BPINIT    18
      COMMON/P/ IPRINT,MODCH,MODGR,PRT1,IDEBUG(35)                      BPINIT    19
      COMMON/CFRACT/ FRACT1,FRACT2                                      BPINIT    20
      COMMON/TUBE/IENDTC(5),ITUBE,IBEGTC(5),INBP                        MOD1115   31
C                                                                       BPINIT    21
      DIMENSION TFRACT(5)                                               BPINIT    22
      DIMENSION ENDTC(5),                                               MOD0925  534
     * CHTC(5),FRACT(2,60)                                              MOD1115   32
      LOGICAL IPAD(2),IDEBUG(35)                                        BPINIT    25
      NAMELIST/BPCHK/ FPOBEG,FPDEND,BPOBEG,BPDEND,IBEGFP,IENDFP,IBEGBP, BPINIT    26
     *               IENDBP,BEGTC,ENDTC,BELBEG,BELEND,IBEGB,IENDB,DX,   BPINIT    27
     *               IBEGTC,IENDTC,TWODR,TWODRR                         BPINIT    28
C                                                                       BPINIT    29
C  INITIALIZE PAD THICKNESS ARRAY                                       BPINIT    30
C                                                                       BPINIT    31
      TEP(1) = 0.0                                                      BPINIT    32
      TEP(2) = 0.0                                                      BPINIT    33
C                                                                       BPINIT    34
C  INITIALIZE FRACT ARRAY                                               BPINIT    35
C                                                                       BPINIT    36
      CALL CLEAR(FRACT(1,1), FRACT(2,60))                               BPINIT    37
C                                                                       BPINIT    38
C  INITIALIZE IPAD                                                      BPINIT    39
C                                                                       BPINIT    40
      IPAD(1) = .FALSE.                                                 BPINIT    41
      IPAD(2) = .FALSE.                                                 BPINIT    42
C                                                                       BPINIT    43
      NAMELIST/BPINP/TEP,DEP,CEP,BEGTC,XLTC,CHTC,ITUBE,FTUBR,PDALST     MOD0604   11
C                                                                       BPINIT    45
C  READ INPUTS AND CALCULATE CONSTANTS                                  BPINIT    46
```

```
C                                                                        BPINIT     47
      READ (5.BPINP)                                                     BPINIT     48
      PDALST=144.*PDALST                                                 MOD0606    13
      IF (IDEBUG(2)) WRITE(6.BPINP)                                      BPINIT     49
C                                                                        BPINIT     50
      DO 20 I=1,2                                                        BPINIT     51
      TEP(I) = TEP(I) / 12.0                                             BPINIT     52
      DEP(I) = DEP(I) / 12.0                                             BPINIT     53
      CEP(I) = CEP(I) / 16.0                                             BPINIT     54
 20   CONTINUE                                                           BPINIT     55
C                                                                        BPINIT     56
      IF (ITUBE .EQ. 0) GO TO 801                                        BPINIT     57
      DO 21 I=1.ITUBE                                                    BPINIT     58
      BEGIC(I) = BEGIC(I) / 12.0                                         BPINIT     59
      XLTC(I) = XLTC(I) / 12.0                                           BPINIT     60
      CHTC(I) = CHTC(I) / 16.0                                           BPINIT     61
   21 CONTINUE                                                           BPINIT     62
C                                                                        BPINIT     63
C  IPAD(1) IS SET .TRUE. IF TEP(1) IS GREATER THAN 0.0.                  BPINIT     64
C  IPAD(2) IS SET .TRUE. IF TEP(2) IS GREATER THAN 0.0.                  BPINIT     65
C                                                                        BPINIT     66
C  DETERMINE IF%  THE FRONT PAD, THE BACK PAD                            BPINIT     67
C  OR BOTH BLACK POWDER PADS ARE PRESENT                                 BPINIT     68
C                                                                        BPINIT     69
  801 IF( TEP(1) .GT. 0.0) IPAD(1) =.TRUE.                               BPINIT     70
      IF (TEP(2) .GT. 0.0) IPAD(2) = .TRUE.                              BPINIT     71
      BPCHG=CEP(1)+CEP(2)+CHTC(1)+CHTC(2)+CHTC(3)+CHTC(4)+CHTC(5)        MOD1002    16
C                                                                        BPINIT     72
C  CALCULATE RADIUS FOR RADIAL GRID NUMBER 2.                            BPINIT     73
C                                                                        BPINIT     74
      TWODR = (AREAC(IBEGB) + AREAR(IBEGB))/3.1415927                    BPINIT     75
      TWODR = SQRT(TWODR)                                                BPINIT     76
C                                                                        BPINIT     77
      TWODRB = (AREAC(IENDB) + AREAR(IENDB))/3.1415927                   BPINIT     78
      TWODRB = SQRT(TWODRB)                                              BPINIT     79
      IF (IPAD(1)) GO TO 500                                             BPINIT     80
      IF (IPAD(2)) GO TO 510                                             BPINIT     81
      GO TO 516                                                          BPINIT     82
C                                                                        BPINIT     83
C  CALCULATE POSITION OF FRONT PAD. THE GRIDS IT LIES IN                 BPINIT     84
C  AND THE FRACTIONS OF THE GRIDS IT LIES IN                             BPINIT     85
C                                                                        BPINIT     86
C  IF THE FRONT PAD EXTENDS BEYOND RADIAL                                BPINIT     87
C  GRID NUMBER 2 - REDISTRIBUTE IT IN RADIAL                             BPINIT     88
C  GRIDS 1 AND 2 BY INCREASING TEP(1).                                   BPINIT     89
C                                                                        BPINIT     90
  500 REP1 = DEP(1) / 2.0                                                BPINIT     91
      IF (REP1 .LE. TWODR) GO TO 505                                     BPINIT     92
      TEP(1) = REP1*REP1*TEP(1)/(TWODR*TWODR)                            BPINIT     93
      REP1 = TWODR                                                       BPINIT     94
 505  FPDBEG = BELBEG - TEP(1)                                           BPINIT     95
C                                                                        BPINIT     96
C  DETERMINE WHICH GRIDS THE PAD IS IN                                   BPINIT     97
C                                                                        BPINIT     98
      IF (FPDBEG .GE. .5*DX) GO TO 506                                   BPINIT     99
      IBEGFP = 1                                                         BPINIT    100
      GO TO 509                                                          BPINIT    101
  508  IBEGFP = FPDBEG / DX + .5                                            APINIT     102
       IBEGFP = IBEGFP + 1                                                  APINIT     103
  509  IENDFP = IBEGB                                                       APINIT     104
C  FRACT IS AN ARRAY WHICH WILL CONTAIN THE FRACTIONS OF GRIDS THE PAD O APINIT     105
       K = IBEGFP                                                           APINIT     106
       IF (K .EQ. 1) GO TO 600                                              APINIT     107
       IF(K.LT. IENDFP) GO TO 601                                           APINIT     108
       FRACT(1,IENDFP) = TEP(1)/DX                                          APINIT     109
       GO TO 603                                                            APINIT     110
  601  FRACT(1,K) = ((FLOAT(IBEGFP) - .5)*DX - FPDBEG)/DX                   APINIT     111
       GO TO 604                                                            APINIT     112
   600 IF(K .LT. IENDFP) GO TO 602                                          APINIT     113
       FRACT(1,IENDFP) = TEP(1)/(.5*DX)                                     APINIT     114
       GO TO 603                                                            APINIT     115
   602 FRACT(1,IBEGFP) = ((FLOAT(IBEGFP) - .5)*DX - FPDBEG)/(.5*DX)         APINIT     116
   604 K = K + 1                                                            APINIT     117
       DO 512 I=K,IENDFP                                                    APINIT     118
          IF(I .EQ. IENDFP) GO TO 605                                       APINIT     119
       FRACT(1,I) = 1.0                                                     APINIT     120
   512 CONTINUE                                                             APINIT     121
   605 FRACT(1,IENDFP) = 1.0 - FRACT1                                       APINIT     122
   603 IF( .NOT. IPAD(2)) GO TO 515                                         APINIT     123
C                                                                           APINIT     124
C  DO CALCULATIONS FOR BACK PAD                                             APINIT     125
C                                                                           APINIT     126
C  IF THE BACK PAD EXTENDS BEYOND RADIAL GRID                               APINIT     127
C  NUMBER 2 - REDISTRIBUTE IT IN RADIAL GRIDS                               APINIT     128
C  1 + 2 BY INCREASING TEP(2).                                              APINIT     129
C                                                                           APINIT     130
  510  REP2 = DEP(2) / 2.0                                                  APINIT     131
       IF(REP2 .LE. TWODRB) GO TO 511                                       APINIT     132
       TEP(2) = REP2*REP2*TEP(2)/(TWODRB*1.CORR)                            APINIT     133
       REP2 = TWODRB                                                        APINIT     134
C                                                                           APINIT     135
C                                                                           APINIT     136
   511 IBEGBP = IENDB                                                       APINIT     137
       BPDBEG = BELEND                                                      APINIT     138
       BPDEND = BELEND + TEP(2)                                             APINIT     139
       IENDBP = BPDEND/DX+.5                                                APINIT     140
       IENDBP = IENDBP + 1                                                  APINIT     141
       FRACT(1,IBEGBP) = 1.0 - FRACT2                                       APINIT     142
       IF(IBEGBP .EQ. IENDBP) GO TO 515                                     APINIT     143
       KB = IBEGBP + 1                                                      APINIT     144
       DO 514 I = KB,IENDBP                                                 APINIT     145
       IF (I .EQ. IENDBP) GO TO 521                                         APINIT     146
       FRACT(1,I) = 1.0                                                     APINIT     147
  514  CONTINUE                                                             APINIT     148
   521 FRACT(1,IENDBP) = (BPDEND - (FLOAT(IENDBP) - 1.5)*DX)/DX             APINIT     149
   515 IF(.NOT. IPAD(1)) GO TO 608                                          APINIT     150
C                                                                           APINIT     151
C  CALCULATE PROPER VALUES FOR FRACT(2,I)                                   APINIT     152
C                                                                           APINIT     153
       TEMPVL = REP1*REP1*3.141593                                          APINIT     154
       IF(IBEGFP .GT. 1 ) GO TO 606                                         APINIT     155
       VOL1 = (TEMPVL - AREAC(1))*DX*.5*FRAC((1,1)                          APINIT     156
       FRACT(2,1) = VOL1/(AREAR(1)*DX*.5)                                   APINIT     157
       GO TO 607                                                            APINIT     158
  606 VOL1 = (TEMPVL - AREAC(IBEGFP))*DX*FRACT(1,IBEGFP)
      FRACT(2,IBEGFP) = VOL1/(AREAR(IBEGFP)*DX)
  607 K = IBEGFP + 1
      IF(K .GT. IENDFP) GO TO 608
      DO 609 I=K,IENDFP
      VOL1 = (TEMPVL - AREAC(I))*DX*FRACT(1,I)
      FRACT(2,I) = VOL1/(AREAR(I)*DX)
      IF(I .EQ. IENDFP) GO TO 608
  609 CONTINUE
  608 IF( .NOT. IPAD(2)) GO TO 516
      TEMPV = REP2*REP2*3.141593
      VOL2 = (TEMPV - AREAC(IBEGBP))*DX*FRACT(1,IBEGBP)
      FRACT(2,IBEGBP) = VOL2/(AREAR(IBEGBP)*DX)
      KB = IBEGBP + 1
      IF(KB .GT. IENDBP) GO TO 516
      DO 700 I = KB,IENDBP
         VOL2 = (TEMPV - AREAC(I))*DX*FRACT(1,I)
         FRACT(2,I) = VOL2/(AREAR(I)*DX)
  700 CONTINUE
C
C  COMPUTE LOCATION OF TUBE CHARGES AND THE PROPER FRACTIONS
C  FOR THE GRIDS THEY OCCUPY
C
 516  DTC = DIAMBT
C
C  CHECK FOR EXTENSION OF TUBE CHARGES
C  BEYOND THE LIMITS OF THE WELL TUBE
C
      IF(ITUBE .EQ. 0 ) GO TO 800
      IF (BEGTC(1) .LT. BELBEG) GO TO 120
      ENDTC(ITUBE) = BEGTC(ITUBE) + XLTC(ITUBE)
      IF (ENDTC(ITUBE) .GT. BELEND) GO TO 121
C
C  DETERMINE GRIDS IN WHICH EACH TUBE CHARGE BEGINS AND ENDS.
C
      DO 100 I=1,ITUBE
      ENDTC(I) = BEGTC(I)+XLTC(I)
      IF(BEGTC(I) .GE. .5*DX) GO TO 805
      IBEGTC(I) = 1
      GO TO 806
  805 IBEGTC(I) = BEGTC(I)/DX + .5
      IBEGTC(I) = IBEGTC(I) +1
  806 IENDTC(I) = ENDTC(I)/DX +.5
      IENDTC(I) = IENDTC(I) +1
C
C  CALCULATE FRACTIONS OF BEGINNING AND END GRIDS
C  WHICH EACH TUBE CHARGE OCCCUPIES
C
      KE = IENDTC(I)
      IF(I .EQ. 1) GO TO 804
      KK = IENDTC(I-1)
      IF(IENDTC(I-1) .EQ. IBEGTC(I)) TFRACT(I) = FRACT(1,KK)
  804 FRACT(1,KE) = (ENDTC(I) - (FLOAT(KE) - 1.5)*DX)/DX
      K = IBEGTC(I)
      IF(K .EQ. 1) FRACT(1,K) = ((FLOAT(K) - .5)   *DX - BEGTC(I))/.5*DX
      FRACT(1,K) = ((FLOAT(K) - .5)*DX - BEGTC(I))/DX
 100  CONTINUE
```

```
C
C  CALCULATE FRACTIONS FOR WHOLE GRIDS OCCUPIED
C
      DO 101 I=1,ITUBE
      KBEG=IBEGTC(I) +1
      KEND = IENDTC(I) - 1
      IF (KBEG .LT. KEND) GO TO 103
      FRACT(I,KBEG) = 1.0
      GO TO 101
  103 DO 102 J= KBEG,KEND
      FRACT(I,J) = 1.0
  102 CONTINUE
  101 CONTINUE
C
C  USING THE FRACTIONS STORED IN ARRAY FRACT,
C  CALCULATE THE POROSITIES FOR EACH GRID
C
C  FRONT PAD
  806 IF (.NOT. IPAD(1)) GO TO 140
C
      DO 110 I=IBEGFP,IENDFP
      DO 111 J=1,2
      VOLF = 3.141593*RFP1*RFP1*TFP(1)
      PHIBP(I,J) = 1.0 - (CEP(1)*FRACT(J,I))/VOLF/BPUENS
  111 CONTINUE
  110 CONTINUE
C
C  BACK PAD
  140 IF( .NOT. IPAD(2) )GO TO 150
C
      DO 112 I = IBEGBP,IENDBP
      DO 113 J = 1,2
      VOLB = 3.141593*RFP2*RFP2*TFP(2)
      PHIBP(I,J) = 1.0 - (CEP(2)*FRACT(J,I))/VOLB/BPUENS
  113 CONTINUE
  112 CONTINUE
C
C  TUBE CHARGES
C
  150 IF(ITUBE .EQ. 0 ) GO TO 802
      RBT = DIAMBT/2.0
      RBTSQ = RBT * RBT
      DO 115 J=1,ITUBE
      KBEG = IBEGTC(J)
      KEND = IENDTC(J)
      IF(IBEGFP .EQ. 1 .AND. IENDFP .EQ. 1) GO TO 810
      FRACT3 = 1.0 - FRACT1
      GO TO 811
  810 FRACT3 = TFP(1)/(.5*DX)
  811 IF(IBEGBP .EQ. IENDBP) GO TO 812
      FRACT4 = 1.0 - FRACT2
      GO TO 813
  812 FRACT4 = TFP(2)/DX
  813 M = KBEG
      N = KEND
      VOLC = RBTSQ*XLTC(J)*3.141593
      DO 116 I=KBEG,KEND
      PHIBP(1,1) = 1.0 - (CHTC(J)*FRACT(1,1))/VOLC/BPDENS
      NAMELIST/BILLT/ VOLC,TUBVOL,CHTC,IFRACT,BPDENS,FREDT
      NAMELIST/BILLB/  VOLC,VOLB,CEP,FRACT2,FRACT4,CHTC,BPDENS,FRED2
      NAMELIST/BILLF/ VOLC,VOLF,CEP,FRACT1,FRACT3,CHTC,BPDENS,FRED1,CHK
      CHK = PHIBP(1,1)
      FRED1= FRACT(1,1)
      FRED2 = FRACT(1,16)
      FREDT = FRACT(1,10)
      IF(IENDFP .EQ. KBFG) PHIBP(M,1) = 1.0 - ((VOLC*CEP(1)*(    FRACT3)
     *          +CHTC(J)*VOLF*FRACT(1,KBFG))/(VOLF*
     *                               VOLC)/BPDENS)
      IF( IDEBUG(33) .AND. IENDFP .EQ. KBFG) WRITE(6,BILLF)
      IF(IBEGBP .EQ. KEND) PHIBP(N,1) = 1.0 - ((VOLC*CEP(2)*(    FRACT4)
     *          +CHTC(J)*VOLB*FRACT(1,KEND))/(VOLB*
     *                               VOLC)/BPDENS)
      IF( IDEBUG(33) .AND. IBEGBP .EQ. KEND) WRITE(6,BILLB)
      IF(J .EQ. 1) GO TO 116
      TUBVOL = RBTSQ*XLTC(J-1)*3.141593
      L = IBEGTC(J)
      IF(IENDTC(J-1) .EQ. IBEGTC(J))PHIBP(L,1) = 1.0 - ((VOLC*IFRACT(J)*
     .CHTC(J-1) + TUBVOL*FRACT(1,L)*CHTC(J))/(TUBVOL*VOLC)/BPDENS)
      IF( IDEBUG(33) .AND. IENDTC(J-1) .EQ. IBEGTC(J)) WRITE(6,BILLT)
  116 CONTINUE
  115 CONTINUE
  802 IF(IDEBUG(30)) WRITE(6,BPCHK)
      RETURN
  120 WRITE(6,1000)
      STOP
  121 WRITE(6,1001)
      STOP
 1000 FORMAT(1H1,'TUBE CHARGE = 1 EXTENDS OUTSIDE THE HELI  TUBE')
 1001 FORMAT(1H1,'TUBE CHARGE = 5 EXTENDS OUTSIDE THE HELI  TUBE')
      END
```

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

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

In this expression, p is in psi, T_o in ^oR, 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 x 10^{-6} ft^2/sec was assumed for the checkout calculations.

ALPHBP The thermal diffusivity for black powder. This quantity is assumed to be 1.0 x 10^{-6} ft^2-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/ft^3.

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 $0.555 \times 10^{-4}/^{o}R$.

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

CV0BP Intercept term in the equation for specific heat of black powder, assumed equal to 0.177.

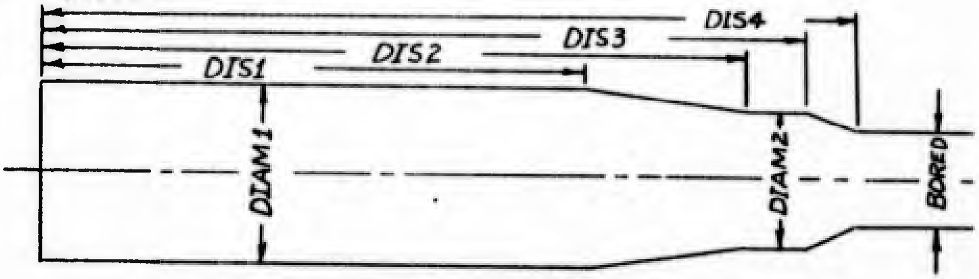
CV0M6 Intercept term in the specific heat equation, $C_v = C_{v_0} + C_{v_H} H$, equal to 0.198 Btu/lbm ^{o}R.

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

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

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

DIAM1
DIAM2
DIS1
DIS2
DIS3
DIS4 Chamber dimensions as defined by the following sketch:



DI0 Propellant grain minor or perforation diameter, equal to 0.03 in. for M6 propellant.

DØ0 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.

HMAX 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.

HMB 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 NHOLES 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.

P0 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 (Btu/ft^2-sec) for present results.

RADHØL 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 (ft^4-lbf)/(lbm^2-°R), for the base run.

RRØM6 Slope of the curve of gas state parameter versus density for the propellant, equal to 1.9 (ft^4-lbf)/(lbm^2-°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 ft-lbf/lbm-°R, for the base run.

ROM6 The ordinate intercept of the curve of gas state parameter versus density for the propellant, equal to 63 ft-lbf/lbm °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, ≈ 900°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.

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

U0 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 x 10^{-4} Btu/ft-sec-°R.

XKBP Thermal conductivity of black powder, assumed equal to 0.2 x 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, $\Delta th = Bp^n$, 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.

P DMAX 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/sec^2-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 subroutine 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 MØDCH and space intervals specified by MØDGR. 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) to (33) and 35 Not used at present.

(34) Printout of primer jet characteristics from subroutine UPDATE.

The title blocks and sample printouts 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	^{o}R

```
IDEBUG(18)                                   CHAMBER OUTPUT
 IPRINT =       0
 TIME =0.

    I     J        RHIRG            RHORG           URG             VRG             HRG             TRG             PCH
                   RMIHR            UDR             TTC             QHAR            RORAH           COTMR           UOTMR
                   THP              RL              DD              DI
    0.                        .55833E-02

    1     1    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .64300000E+00   0.              0.              0.              0.              0.              0.
               .53500000E+03   0.              0.              0.
    0.                        .55826E-02

    1     2    .73613662E+00   .61390653E-01   0.              0.              .18574250E+03   .48214570E+03   .19227716E+02
               .97689968E+00   0.              .53500000E+03   0.              0.              0.              .18909954E-02
               .74500000E+03   .85333333E-01   .36583333E-01   .36666667E-02
    0.              0.

    1     3    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .13000000E+01   0.              0.              0.              0.              0.              0.
          0.              0.              0.              0.
    0.                        .55833E-02

    4     1    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .81926846E+00   0.              0.              0.              0.              0.              0.
               .53500000E+03   0.              0.              0.
    0.              0.

    4     2    .58099656E+00   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .10000000E+01   0.              .53500000E+03   0.              0.              0.              0.
          0.                   .85333333E-01   .36583333E-01   .36666667E-02
    0.              0.

    4     3    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .10000000E+01   0.              0.              0.              0.              0.              0.
          0.                   0.              0.              0.
    0.                        .55833E-02

    8     1    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .75168568E+00   0.              0.              0.              0.              0.              0.
               .53500000E+03   0.              0.              0.
    0.              0.

    8     2    .58099656E+00   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .10000000E+01   0.              .53500000E+03   0.              0.              0.              0.
          0.                   .85333333E-01   .36583333E-01   .36666667E-02
    0.              0.

    8     3    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .10000000E+01   0.              0.              0.              0.              0.              0.
          0.              0.              0.              0.
    0.                        .55833E-02

   12     1    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .40317616E+00   0.              0.              0.              0.              0.              0.
               .53500000E+03   0.              0.              0.
    0.              0.

   12     2    .58099656E+00   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .10000000E+01   0.              .53500000E+03   0.              0.              0.              0.
          0.                   .85333333E-01   .36583333E-01   .36666667E-02
    0.              0.

   12     3    .10000000E+01   .59846591E-01   0.              0.              .14444781E+03   .53500000E+03   .14700000E+02
               .13000000E+01   0.              0.              0.              0.              0.              0.
          0.              0.              0.              0.
    0.                        .55833E-02
```

```
  18   1   .10000000E+01   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .94157986E+00  0.             0.            0.            0.        0.              0.
           .53500000E+03  0.             0.            0.
    0.          0.

  18   2   .54019420E+00   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .10000000E+01  0.             .53500000E+03 0.            0.        0.              0.
          0.              .45333333E-01  .36583333E-01 .36666667E-02
    0.          0.

  18   3   .10000000E+01   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .10000000E+01  0.             0.            0.            0.        0.              0.
          0.              0.             0.            0.
    0.          0.

  20   1   .10000000E+01   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .10000000E+01  0.             0.            0.            0.        0.              0.
          0.              0.             0.            0.
    0.          0.

  20   2   .10000000E+01   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .10000000E+01  0.             0.            0.            0.        0.              0.
          0.              0.             0.            0.
    0.          0.

  20   3   .10000000E+01   .59865591E-01  0.            0.                     .14444781E+03   .53500000E+03   .14700000E+02
           .10000000E+01  0.             0.            0.            0.        0.              0.
          0.              0.             0.            0.
IDEBUG(18)
  SDOT

  DOTMIG  =  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,  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,  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,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,

  DOTMAG  =  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,  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,  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,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,

  T20     =  0.535E+03,  0.535E+03,  0.535E+03,  0.535E+03,  0.535E+03,            0.535E+03,  0.535E+03,  0.535E+03,
             0.535E+03,  0.535E+03,  0.535E+03,  0.535E+03,  0.535E+03,            0.535E+03,  0.535E+03,  0.535E+03,
             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,  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,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,

  $END

IDEBUG(19)                                        BARREL OUTPUT
   I          PHI                 RHOG              UG               HG              TG              PG              UP
                                   QL              UDRAG            RHOAT           AMASS           AMOM            AFMER
                                  XLM              DOA              DIA

     1    .10000000E+01    .59865591E-01   0.               .14444781E+03   0.                      .14700000E+02   0.
                          0.               0.               0.                    .26435984E+00   .26435984E+00   .26435984E+00
                          0.               0.               0.
IDEBUG(20)
  PRESSURE AT GRID (1+1) IS      .14700000E+02
   THE MASS OF GAS IN THE SYSTEM IS      .0806
IDEBUG(21)
   THE MASS OF PROPELLANT AND BLACK POWDER IN THE SYSTEM IS    56.7000
  ACTUAL GAS IN THE SYSTEM IS     .1241   ADJUSTING FRACTION IS    1.5441
IDEBUG(24)
      8     .1401026E-03    .7004126E+02    .3964551E+02    .2202906E+02    .2275936E+02    .2275476E+02   0.             0.
     19     .2957716E-03    .1352796E+03    .6619166E+02    .2283116E+02    .2275446E+02    .2276216E+02   0.             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.

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

AFN1, AFN2 Parameters representing the product of 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.

AMØM(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.

AREAH2(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 I.

ATPB Propellant burning parameter, = AGEN * TO + 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.

BØRED8 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.

BPRAD0 See Appendix C.

BRNOUT 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 * TO.

CV Specific heat at constant volume.

CVH See Appendix C.

CVO 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,** $= (1.5\ DØ0^2 \times LO)^{\frac{1}{3}}$

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.

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

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

DØN1 **New propellant grain exterior diameter at J = 1 after igniter tube has failed in compression and propellant has shifted to the axis.**

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

DØTM(I,J) Dummy mass flow array used in Subroutine MFLOW.

DØTMBG(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.

DØO 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.

GTRHØP Parameter used in computing propellant movement in the barrel, GRAV*DELT/RHØP.

HBG Total enthalpy at grid I,J in the chamber matrix.

HBGN 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 MØDCH 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.
JTSHR	Computed shear force in end pad caused be 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.

NRØWH 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.

PO 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.

PMDØT(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.

PRØMAS 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.

QBCØNS Parameter used in the calculation of black powder temperature rise.

QCØNBP Heat lost from gas as result of heating black powder.

QCØNS Parameter required for propellant surface temperature rise calculations, = 2 * SQRT (ALPHA/3.1416)/XK.

QCØNV Heat transfer per ft^2 - used in PRPFIR and BNDLYR.

QDØT 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.

RADHØL See Appendix C.

RADPB See Appendix C.

RADPS Calculated pseudo hole radius - Subroutines CHSET and HØLSET

RAPJT Radius of primer jet at base pad.

RAF1, RAF2 Parameters used in subroutine TUBFAL that 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 propellant grain diameter, used in calculationsof 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 paramtrer 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.

TWØDT Calculation parameter, 2 * DELT.

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 * DELT/DR.

T2DX Calculation parameter, 0.5 * DELT/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.

UAMØM(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.

UUP(I) Updated propellant grain velocity in barrel grid I.

UXLB(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 * DØO ** 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.

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

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

WØB 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.

XJUL 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.

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

XLO See Appendix C.

XNTUB See Appendix C.

XØB 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
A	AGENBP
A	AREAH
a	ACC
A_{bp}	ADVBP
A_{ci}	ARRAC
A_{dj}	ADJUST
Aener	AENER
$Aener_{t+\Delta t}$	UAENER
$Aex_{i,n}$	AREAXP
Aex_{n}	TA
A_{fi}	AREAH
A_{f1}	AFN1
A_{f2}	AFN2
A_{gpi}	AREAGP
A_{hi}	AH
Amass	AMASS
$Amass_{t+\Delta t}$	UAMASS
Amom	AMØM
$Amom_{t+\Delta t}$	UAMØM
Ap	ADV
A_{ri}	AREAR
A_{sector}	AREAS
$A_{triangle}$	AREAT
B	BGEN
B	BGENBP
Beg	BELBEG
Bend	BELEND
BP_{chg}	BPCHG
B_{t}	BTUB
C	CGEN
C_{beg}	BEGTC
C_{cn}	CHTC
C_{ep}	CEP
c_{f}	CF
c_{f}/c_{fi}	CFR
C_{hwt}	CHWT
CL	XLTC
cp	CP
cv	CV
c_{vh}	CVH
c_{vo}	CVO
$D, D_{\emptyset}$	DØ
d, Di	DI
d_{bt}	DIAMBT
dc_{1}	DIAM1
dep	DEP
D_{p}	DEFF
$D_{t+\Delta t}$	DØTDT
$d_{t+\Delta t}$	DITDT
DIn1	DIN1
dis	DISCRM, DISCRP
DØn1	DØN1
dpr	DM
Dr	RDRAG
d_{ram}	DRAM
D_{x}	XDRAG

E	PEXP
e	EXPBP
f_1	FRACT
F_d	FDPRIM
F_{dmax}	FDMAX
f_{bi}	FRACT
Fract1	FRACT1
Fract2	FRACT2
Frict	FRICT
f_m	FRCTF
g	GRAV
gp	GAP
gt	TØPGAP
H	HBG
$H_{t+\Delta t}$	HBGTD
H^*	HSTAR
H_o	HO
Hbgn	HBGN
H_{bp}	HBP
H_g	HG
H_{gp}	HBGN
H_{ign}	HIGN
H_{max}	HMAX
H_{mb}	HMB
H_s	HSTAT
H_{stat}	HSTAT
I	PINER
I_{begb}	IBEGB
I_{begc}	IBEGC
I_{endb}	IENDB
I_{epb}	IBEGFP
I_{epd}	IENDFP
Incre	INCRE
J	XJUL
k	RØTK
k_{ij}	TXK
kp	XK
L	XL
$L_{t+\Delta t}$	XLTDT
Lim1	LIM1
Lim2	LIM2
l_{min}	ELT
M	FM
M^2	MACHSQ
m	PMASS
$\dot{m}$	DØTM
m_B	DØTMBS
$\dot{m}_M$	DØTMB
m_g	PRØMAS
$\dot{m}_{gp}$	DØTMBG
$\dot{m}_{ign}$	DØTMIG
$\dot{m}_p$	GASMAS
m_p	PMDØT
$\dot{m}_{pr}$	DØTMPM
N_{bp}	TERMBP
Ngr	NGR
N_{gt}	NGT
N_{gt3}	NGT3
N_{gx}	NGX

n_n	NRØWH
Npr	TERMP
n_t	XNTUB
Nu	TXNUS
Nu_{bp}	TXNUBP
Nu_h	TXNUSH
p	PCH
p_o	Po
p_c	PCØMPL
p_{co}	PCØMP
p_{force}	PFORCE
p_g	PG
p_{ho}	PHØØP
p_h	PHØØPL
p_L	PL
p_{prop}	PRØP
p_s	PSTAT
p_x	PRESX
$P_{px,t,M82}$	PRMXP
Q	QDØT
q	QBAG
Q_b	QBAG
q_{bp}	QCØNBP
q_{conv}	QCØNV
R	RADPB
R	RGAS
r	R
R_{af1}	RAF1
R_{af2}	RAF2
r_{bp}	BPRAD
Ro	RO
Re	RETX
Re^*	RESTRX
Re_{bp}	RETXBP
R_{eh}	RETXH
R_{ex}	RE
r_h	RADHØL
r_o	FTUBR
r_{pb}	R
r_{ps}	RADPS
$R\rho$	RRØ
S_{do}	STDØFF
S_{dj}	SPDJ
S_{dtot}	DISPR
S_o	AREA
T	TBG
T	TG
t	TIME
T_o	TO
T^*	TSTAR
T_{bp}	TBP
t_{eff}	TEFF
t_{ep}	TEP
t_{ep1}	TEP(1)
T_{hi}	THICK
T_{ho}	THICKT
T_z	TZC
u	UBG
u_{bgn}	UBGN

u_{hi}	UHØLE	Zo	ZO
$u_{t+\Delta t}$	UBGTD	Z_d	ZD
u_{ave}	UBGAVE	Z_u	ZU
u_g	UG		
u_p	UPB	α	ALPHA
$u_{pt+\Delta t}$	UPBDT	β	BETA
u_{pave}	UPBAVE	γ	GAM
u_{pn1}	UPN1	γ_{bp}	GAMBP
Upr	UPRM	δ	DELTA
U_R	UGAS	δ	DEPTH , STDOFF
		δ^*	DELSTR
v	VBG	δ^{**}	DELDST
$v_{t+\Delta t}$	VBGTD	Δm_{bp}	DØTMP
V(t)	VNEW	Δm_{pi1}	DMPI1
VØ	VØLD	Δm_{pi}	DMPI
V_{ave}	UBGAVE	Δm_{pi2}	DMPI2
V_{bp}	VBP	Δmpi+1	DMPIP1
V_{chg}	VØLCHG	Δmpi-1	DMPIM1
V_{new}	VNEW	Δr	DR
V_{old}	VØLD	Δr_1	DELTR1
V_p	VP	Δr_2	DELTR2
V_{pr}	VPRØP	Δr_{bp}	BURNL
		Δt	DELT
W_{ob}	WØB	ΔV_{axis}	DVAXIS
		ΔV_{axit}	DVAXIT
X	EXF	ΔX	DX
x	XB		
x_{cL}	XCL	λ	LAMBDA
x_{epb}	FPDBEG		
$x_{\ell b}$	XLBEL	μ	TXMU
x_{Ln1}	XLN1	μ(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
$\phi_{i\ t+\Delta t}$	PHII	ρ_b	RHØB
$\phi_{i1\ t+\Delta t}$	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		
ϕ_s	SHAPE	θ	THETA
ρ or ρ_{bg}	RHØBG	ω	ØMEGA
$\rho_{t+\Delta t}$	RHØBTD		