AD-A051 827 CALSPAN CORP BUFFALO N Y F/G 19/1
QUALITY CONTROL OF CONTINUOUSLY PRODUCED GUN PROPELLANT.(U)
AUG 77 E B FISHER
UNCLASSIFIED CALSPAN-SA-5913-X-1 NL

1 OF 2
AD
A051827

AD A051827

12

REPORT NO. SA-5913-X-1

QUALITY CONTROL OF CONTINUOUSLY PRODUCED GUN PROPELLANT

Edward B Fisher
Calspan Corporation
P.O. Box 235
Buffalo, New York 14221

AD No.
DDC FILE COPY

August 1977

D D C
MAR 27 1978
F

Report for Period 29 March 1976 – 1 July 1977

Prepared for:

Department of the Army
Picatinny Arsenal
Dour, N.J. 07801
Attn: SARPA-QA-X

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
CALSPAN SA-5913-X-1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) Quality Control of Continuously Produced Gun Propellant.

5. TYPE OF REPORT & PERIOD COVERED
(9) Technical Summary rept. 3/29/76 - 7/1/77 29 Mar 76-1 Jul 77

6. PERFORMING ORG. REPORT NUMBER
SA-5913-X-1

7. AUTHOR(s)
(10) ~~E.~~ Edward B. Fisher

8. CONTRACT OR GRANT NUMBER(s)
DAAA21-76-C-0182 new

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Calspan Corporation
P.O. Box 235
Buffalo, New York 14221

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

11. CONTROLLING OFFICE NAME AND ADDRESS
Department of the Army
Picatinny Arsenal
Dover, NJ 07801 ATTN: SARPA-QA-X

12. REPORT DATE
(11) Aug[redacted]77

13. NUMBER OF PAGES
(12) 193p.

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

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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

DDC
RECEIVED
MAR 27 1978
RECEIVED
F

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

175mm Gun
105mm Howitzer
Mathematical Model
Propellant Ignition and Combustion
Two-Phase Flow
Interior Ballistics

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

The purpose of the subject program was to improve and optimize the Calspan artillery models. In particular, the models were examined and reformulated to include the Lennard-Jones G-12 potential as the equation of state and to use parameters generated by the BLAKE code as input. Other improvements were made and anomalies observed in previous calculations were eliminated. Comparisons between computer simulations and experimental results from an M2A2 105mm howitzer involving eight special lots were fair → next page

T,C

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407727

20.

but some deviations exist. It is evident that code inputs and representation of burn rate require further improvement.

ACCESSION for
NTIS White Section ☒
DDC Buff Section □
UNANNOUNCED □
JUSTIFICATION Per ltr
on file
BY
DISTRIBUTION/AVAILABILITY CODES
/or SPECIAL
A | 23 E.K. | -

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1

INTRODUCTION

Briefly, the existing procedure for the acceptance of propellants begins with closed bomb testing during manufacture. Batches of propellant are classified according to relative burn rates and pressure levels exhibited when tested against a standard propellant lot. The batches to be blended to make up a lot of propellant are selected according to the relative burning characteristics measured in the closed bomb. This blending process has resulted in a stable product with slight variations between lots. The finished lot is sampled and tested in the closed bomb. An approximate charge weight is determined based upon the closed bomb test results. The propellant is then loaded into charges at this weight and fired using standard metal parts and igniters. During this propellant acceptance test, muzzle velocities and peak chamber pressures are measured. Propellant acceptance is based upon achievement of satisfactory muzzle velocity and pressure levels with a propellant charge volume below a specified maximum value. Charge assessment (the determination of the charge weight necessary to provide a predefined muzzle velocity) is also determined from the acceptance test data. If a propellant lot does not exhibit satisfactory performance characteristics, it is rejected.

The first modernized propellant manufacturing facility wherein propellant is produced on a continuous production line is currently being constructed. A candidate item for production is M1 propellant for the M67 charge of the M103-105mm howitzer. The Army project entitled "Acceptance of Propellant Produced via the Continuous Process" has a goal of developing the acceptance test plan for the CASBL. Obtaining the knowledge of which propellant parameters affect the interior ballistics cycle and the ranking of parameters by sensitivity is a crucial plateau which must be reached.

An an aid in determining the sensitivity of the interior ballistics cycle to propellant characteristics and for the development of improved understanding of propellant interior ballistics functions, Calspan has developed a mathematical simulation of the 175mm gun system. This model has been used to study the effect of the propellant ignition process on the entire interior ballistics cycle. Furthermore, the model has been shown to have the ability to predict non-normal, even hazardous, combustion shock wave generation in the bed of propellant.

The model was also modified to incorporate those features unique to the 105mm howitzer. Together these models provide the ability to represent most U.S. Army artillery configurations by simply changing input parameters.

Extensive use of the models and recent developments found in the literature have revealed areas in the models that require improvement. This program is devoted to upgrading both the 175mm and 105mm howitzer codes, although primary emphasis was given to the 105mm howitzer code. The primary items addressed during this program were:

1. Reformulation of the governing equations.
2. Change of equation of state and use of BLAKE code--generated inputs.
3. Improvement in treatment of the dual-granulation propellant movement and combustion in the barrel.
4. Investigation of discontinuity and other deviations in computed pressure-time curve from the experimental curve.
5. Inclusion of chamber heat loss.
6. Improvement in treatment of propellant motion and bed compaction.

Section 2

MODEL STATUS

2.1 OVERVIEW

The Calspan artillery codes were reviewed and modified with regard to the areas listed in the Introduction. This section will describe the current status of the model with regard to its formulation, treatment of the various empirical functions, and computational procedures. This report is written with respect to work accomplished on the 105mm howitzer model, originally described in Ref. 1. However, the status of the model as presented in this report also applies to that of the 175mm gun--155mm howitzer code, as described in References 2 and 3. The basic structure of the 105mm howitzer code is given in the next section to provide continuity with previous works.

2.2 REVIEW OF MODEL STRUCTURE

The mathematical models, which consists of two major routines, chamber and barrel, with domains illustrated in Figure 1, is described in Reference 1. The following section, taken from the Reference 1, is given here to provide necessary background for the discussion which follows.

2.2.1 General 105mm Howitzer Configuration

The general configuration of the 105mm howitzer is shown in Figure 2. The complete round consists of a steel cartridge case, primer, propellant charge, and shell. The primer is made of brass or steel and is mounted to the base of the cartridge case. The propellant charge is contained in a string of up to seven small rectangular bags that fit loosely in the case around the primer. The shell fits loosely in the cartridge case and provides the major portion of the cross-sectional area for the pressure to act against. The rotating band performs a sealing function as well as the means for rotational acceleration. Any leaks past the band tend to reduce system efficiency, but since this band undergoes an interference fit as it enters the barrel, the seal is assumed to be tight, allowing negligible loss of gas. When loaded, the shell is not rammed and must travel a short distance

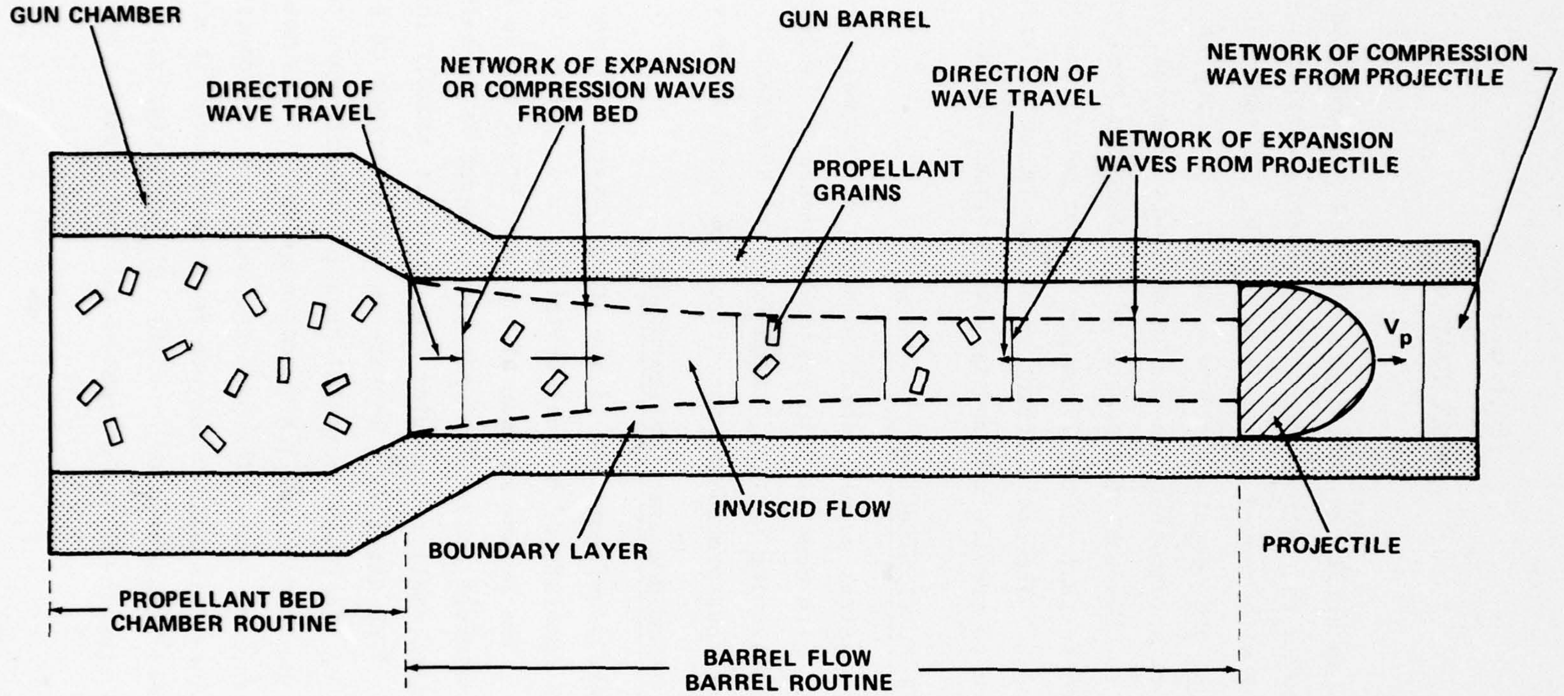

GUN CHAMBER
DIRECTION OF WAVE TRAVEL
NETWORK OF EXPANSION OR COMPRESSION WAVES FROM BED
GUN BARREL
PROPELLANT GRAINS
DIRECTION OF WAVE TRAVEL
NETWORK OF EXPANSION WAVES FROM PROJECTILE
NETWORK OF COMPRESSION WAVES FROM PROJECTILE
V_p
INVISCID FLOW
BOUNDARY LAYER
PROJECTILE
PROPELLANT BED CHAMBER ROUTINE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

Figure 1 SCHEMATIC DIAGRAM OF THE 105 mm HOWITZER SYSTEM ILLUSTRATING THE DOMAINS OF THE TWO COMPUTER ROUTINES

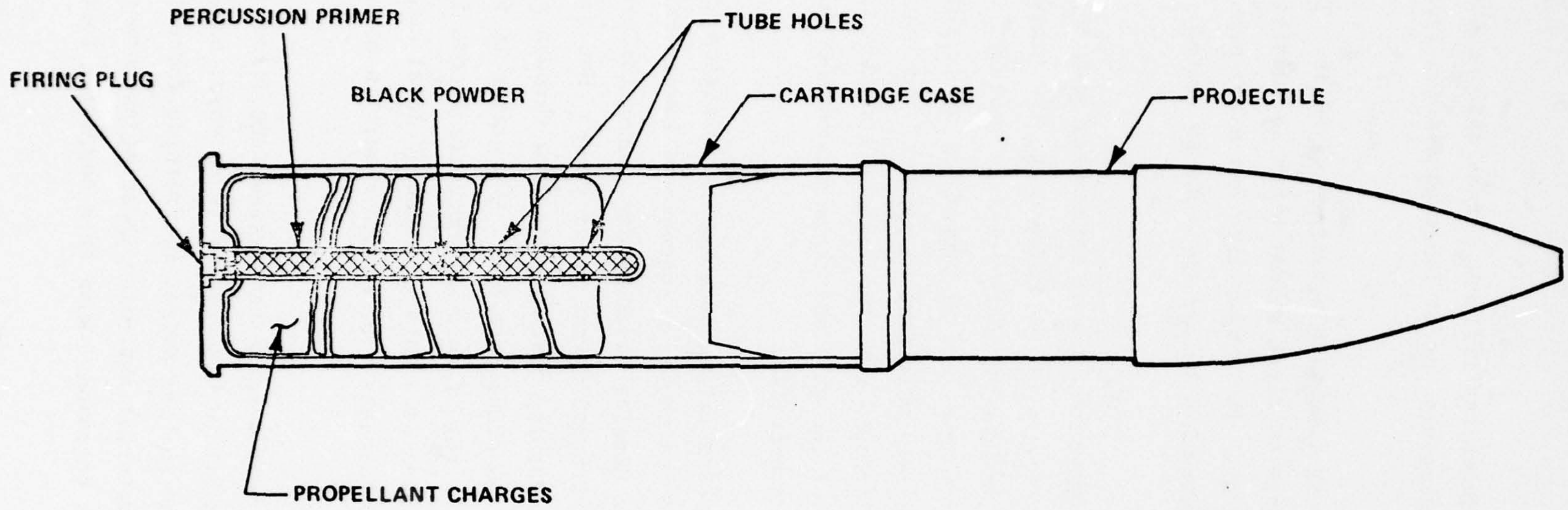


Figure 2 SCHEMATIC DIAGRAM OF THE 105 mm CARTRIDGE PRIOR TO FIRING

before the rotating band engages the rifling. It is assumed that the blow-by that occurs here is also negligible, or at least consistent from round to round.

The primer is a long tube with a pattern of holes. The tube is initially filled with a charge of black powder which is initiated by firing a percussion-sensitive element. The primer tube has a wax paper liner which allows high pressures to be reached before the tube is vented. This provides a more positive ignition.

The propellant charge consists of seven bags sewn together in a string. Before firing, the projectile is removed and the charge is adjusted by removing bags until the desired velocity level is reached. The first two bags contain 0.0135 in. web single-perf M1 propellant while the remaining five bags contain 0.0245 in. web multiperf M1 powder. The bags are contoured to fit the case and can be dropped into the case in a random fashion. The charge rests on the bottom of the case and there is considerable free volume between the charge and the projectile.

The actual gun system firing sequence is initiated when the percussion element is fired and causes a sequence of events resulting in black powder ignition. The burning black powder causes the pressure to rise and eventually exceed the strength of the paper liner. Hot gas and burning particles generated by the burning black powder flow through primer tube holes and into the end of the propellant bed. The grains in the main propellant charge 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 the pressure builds up in the system, the force created by pressure acting on the projectile base causes it to move, engage the rifling, and eventually overcome the initial barrel restraining force. This restraining force is a result of the material extrusion/shearing phenomena that occur while the rotating band is engraved. When this engraving force has been exceeded by the pressure, 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. This, along with friction and engraving forces, constitutes 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 the retarding forces. The sequence of events of interest in this model terminates when the projectile has passed from the barrel.

2.2.2 Chamber Routine

The Chamber Routine calculates all phenomena concerned with ignition, gas generation, and flow inside the chamber of the 105mm howitzer. The 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 primer tube and one for the main charge as shown in Figure 3. 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 in a manner that simulates flow through primer tube holes, thereby achieving a semblance of radial mass and energy transport.

Use of a one-dimensional grid system places some constraints on positioning of the propellant charge. Variations in propellant or propellant bed density can be expressed only as functions of axial location. The seven zone propellant charge with two different grain configurations is loaded in a random configuration, as mentioned previously. One choice for positioning the charge in the code is to distribute each zone over a length of the case with zones overlapping. Another is to assume a structured charge that is sequentially loaded according to zone number, beginning with zone 1 at the breech end of the case. The latter charge configuration was chosen for the 105mm howitzer model.

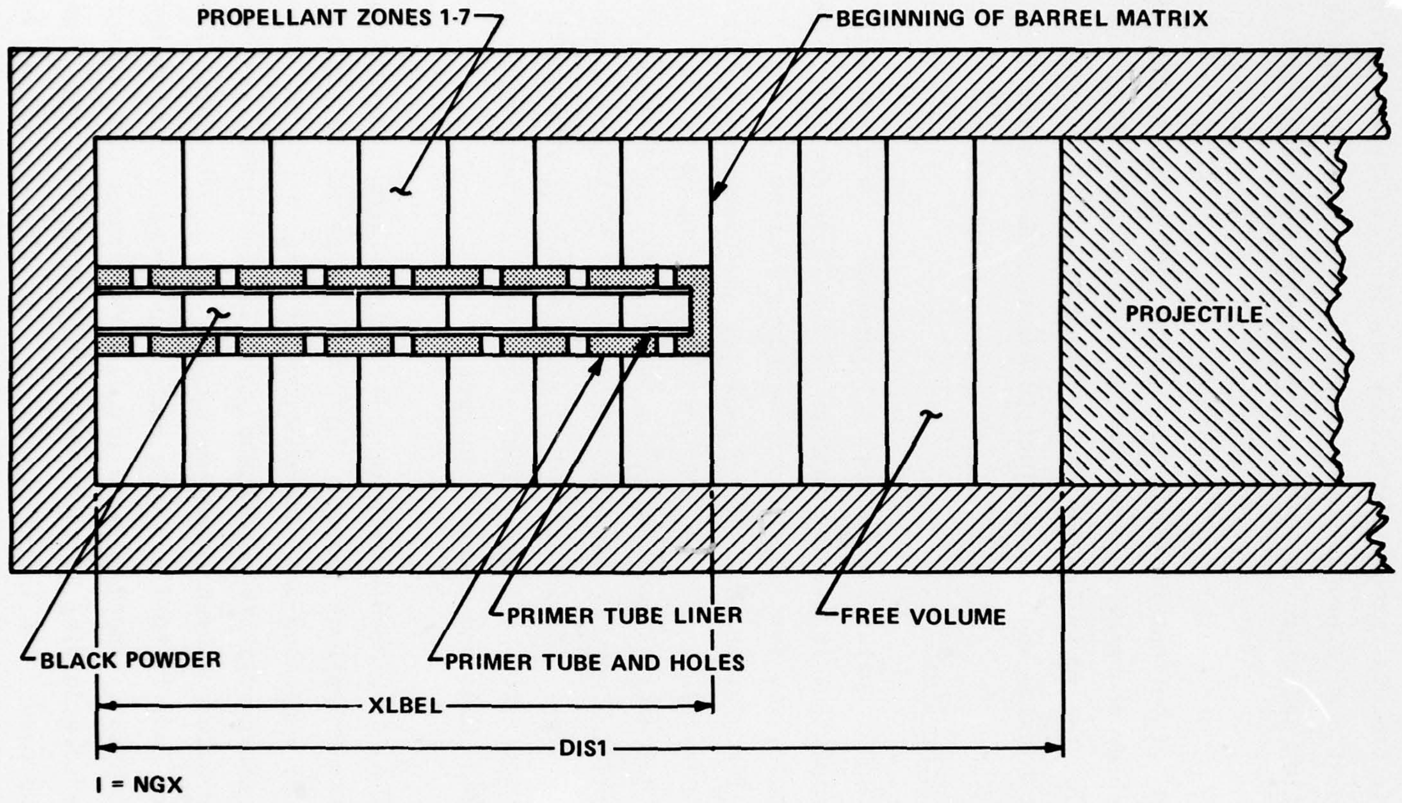


Figure 3 MULTIPLE ONE-DIMENSIONAL CHAMBER GRID NETWORK FOR THE 105 mm HOWITZER

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 primer tube holes. In addition, equations expressing conservation of mass and momentum are included in order to express movement of the propellant bed. These equations are solved in such a way that no mixing of the two grain configurations is allowed.

The output of the primer percussion element is not specifically represented in the current model. Its effects are represented by assuming the black powder in the first primer grid is ignited initially. The gas generated by this powder flows through the tube and ignites the remainder of the primer charge.

The treatment of flow through primer tube holes has been simplified but still retains the essential features. The model considers an arbitrary number of rows of exhaust ports, each row consisting of the holes (two holes per row for the 105mm howitzer) at a given axial station. Each row is treated as a continuous flow area rather than as discrete holes, since the latter would require the full three-dimensional treatment. Gas flows sonicly or subsonicly through the holes, according to the existing pressure ratio across the hole after a pressure sufficient to cause liner failure has been reached at the hole location.

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 started to move. The projectile base is assumed to be reflective so that waves are transmitted from the breech to the base.

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, is an input that can be varied independently from the others.

2.2.3 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 4.

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 consumed by 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

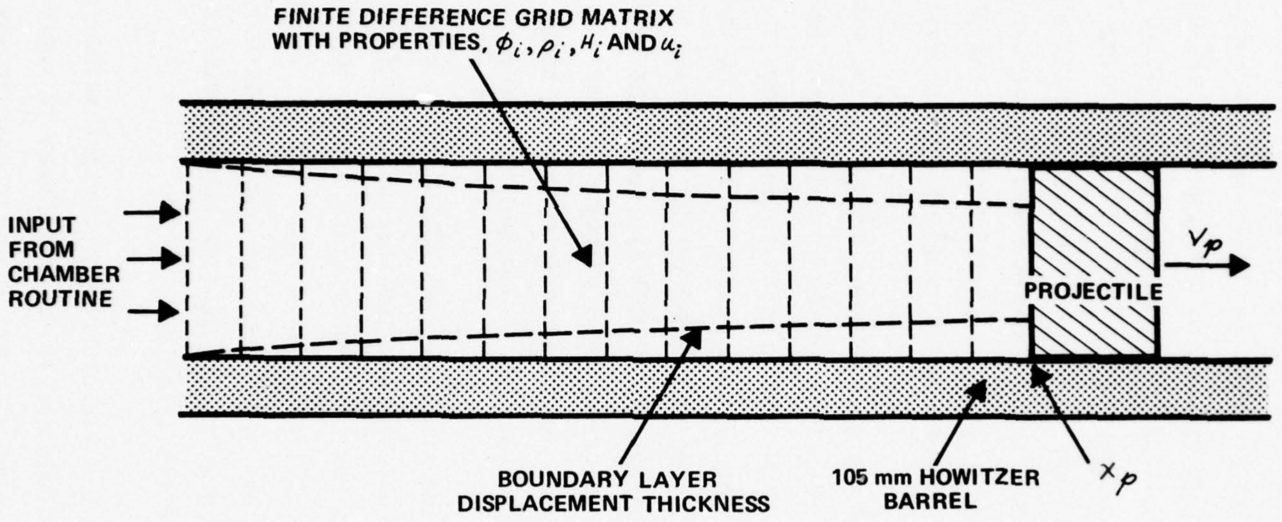


Figure 4 GRID MATRIX FOR THE BARREL OF THE 105 mm HOWITZER

rotating band as it begins motion through the barrel. Rotational acceleration involves the axial moment of inertia and the twist of the rifling. It actually becomes a component of the axial acceleration that requires some of the pressure force. In this sense, it acts as a retarding mechanism. The frictional force is assumed to occur 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 normal force. Another resistance force that has been included but is probably not too significant for the 105mm howitzer is the pressure head that is accumulated 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 is an input but the equations describing the twist of the rifling are built into the logic.

2.3 GOVERNING EQUATIONS

The governing equations for the Calspan interior ballistics codes have been reviewed in light of the JANNAF Combustion Workshop held in conjunction with the 12th JANNAF Combustion Meeting in August 1975. The derivations of Culick[4] and Gough[5] have been reviewed in order to obtain different perspective on the two-phase flow problem. This effort has resulted in some changes to the Calspan governing equations.

In addition, the equation of state has been changed to the Lennard-Jones 6-12 potential. This change necessitated revision of the energy equation. This equation, written in the form

$$p\left(\frac{1}{\rho} - \eta\right) = RT$$

where the co-volume, η , is given by

$$\eta = Ao + A1\rho + A2\rho^2 + A3\rho^3$$

with cubic fit co-volume coefficients as generated by the BLAKE code, is thought to be the most accurate equation of state for a wide range of pressures, including the extremely high pressures encountered in gun applications. This change necessitated revision of the form of the energy equation, since the previous form incorporated the state equation.

The governing equations express conservation of mass, momentum, and energy in a two-phase compressible flow system. This two-phase system is assumed to be a continuum that represents interactive flow through a mobile bed of propellant. This formulation assumes that the large propellant grains can be treated in the same manner as a molecule of air. This is not realistic and the inequality is reflected at various points in the derivation as will be noted in the subsequent discussion.

Continuity Equations

The continuity equations, which represent conservation of solid and gaseous mass, are unchanged. These equations for a one-dimensional system are:

Gas Phase:

$$\frac{\partial(\phi\rho)}{\partial t} + \frac{1}{\bar{A}}\frac{\partial}{\partial x}(A\phi\rho u) = \dot{m}_{comb} + \dot{m}_s$$

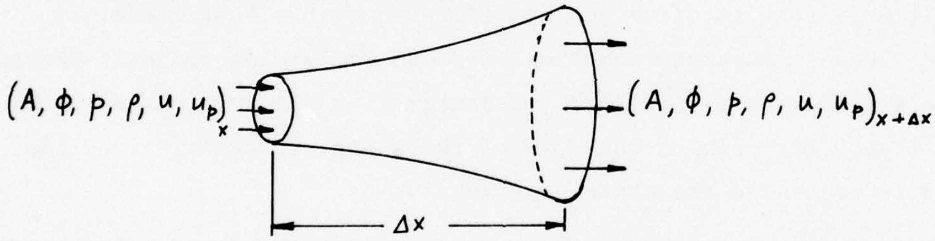
Solid Phase:

$$\frac{\partial}{\partial t}[(1-\phi)\rho_p] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A(1-\phi)\rho_p u_p] = -\dot{m}_{comb} + \dot{m}_{sp}$$

where $\dot{m}_{comb}$ is the rate of propellant mass burned and $\dot{m}_s$ and $\dot{m}_{sp}$ represent respective quantities of gas and solid added through the boundaries of the parallel grid networks. The propellant density, ρ_p, in the solid phase equation is treated as a constant in the Calspan codes.

Momentum Equations

The equations that express conservation of momentum in a one-dimensional two-phase flow system were rederived along the lines of Culick[1]. The elemental volume for this derivation is:


$(A, \phi, p, \rho, u, u_p)_x$
$(A, \phi, p, \rho, u, u_p)_{x+\Delta x}$
Δx

The total momentum contained within this volume at any instant is

$$\bar{A}\,\Delta x\,[\,\bar{\phi}\bar{\rho}\bar{u} + (1-\bar{\phi})\rho_p\bar{u}_p\,]$$

where $\bar{A}$ is the average cross-sectional area of the grid and $\bar{\phi}$, $\bar{\rho}$, $\bar{u}$ and $\bar{u}_p$ are the average flow parameters in the elemental volume and the propellant density, ρ_p, is assumed to be constant.

The net momentum change within the elemental volume due to flow through the end boundaries is

$$[A\phi\rho u^2 + A(1-\phi)\rho_p u_p^2]_x - [A\phi\rho u^2 + A(1-\phi)\rho_p u_p^2]_{x+\Delta x}$$

The parameters in this relationship are those that exist exactly at the end faces of the elemental volume and are not average quantities.

The pressure forces acting on the elemental volume are those forces acting on the end face plus the axial component of pressure acting on the side walls of the volume. The pressure force is written as

$$(pA)_x + \int_0^{\Delta x} p\left(\frac{dA}{dy}\right)dy - (pA)_{x+\Delta x}$$

If the pressure, p, and the area of the element, A, are assumed to vary linearly over Δx, then

$$p = P_{y=0} + Ky$$

and $$\frac{dA}{dy} = c$$

After integration, and substitution of $C = \frac{dA}{dx}$ *and* $K = \frac{dp}{dx}$ over the length of the elemental volume, the net pressure force on the volume is

$$-A_x \frac{dp}{dx}\Delta x - \frac{dp}{dx}\left(A_{x+\Delta x} - A_x\right)\frac{\Delta x}{2}$$

or $$-\bar{A}\frac{dp}{dx}\Delta x$$

This term is separated into components for each phase when the global momentum equation is separated.

The stress force supported by the compacted bed of propellant is

$$[\sigma A(1-\phi)]_x - [\sigma A(1-\phi)]_{x+\Delta x}$$

This is the only solid propellant stress force included in the model which means that a free slip condition exists at the wall.

Gas and solids added to the element through the boundary add to the total momentum in the element if they have a velocity component common to that of the one-dimensional element. Also, momentum is lost if moving gas or solid is allowed to flow from the element. These source or sink terms are written as:

$$\dot{m}_s u_s + \dot{m}_{sp} u_{sp}$$

where u_s and u_{sp} are appropriate velocities.

Combining these terms, dividing by x and $\overline{A}$, the global momentum equation that expresses the rate of momentum change in an elemental volume is:

$$\frac{\partial}{\partial t}[\phi\rho u + (1-\phi)\rho_p u_p] = -\frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u^2 + A(1-\phi)\rho_p u_p^2] - \frac{\partial P}{\partial x}$$
$$-\frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\sigma(1-\phi)] + \dot{m}_s u_s + \dot{m}_{sp} u_{sp}$$

Letting $\frac{\partial P}{\partial x} = \phi\frac{\partial p}{\partial x} + (1-\phi)\frac{\partial p}{\partial x}$ and separating the terms of the equation with regard to solid and gas phase components:

Gas Phase:

$$\frac{\partial}{\partial t}(\phi\rho u) + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u^2] + \phi\frac{\partial p}{\partial x} - \dot{m}_s u_s = F$$

Solid Phase:

$$\frac{\partial}{\partial t}[(1-\phi)\rho_p u_p] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A(1-\phi)\rho_p u_p^2] + (1-\phi)\frac{\partial p}{\partial x} + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\sigma(1-\phi)]$$
$$- \dot{m}_{sp} u_{sp} = -F$$

The term F represents those forces internal to the elemental volume that result from interaction between the two phases, which is simply an exchange of momentum between the phases. One such interaction is drag caused by resistance to flow of one phase relative to the other.

$$D_x = f(f, d, \phi, \rho, u-u_p)$$

The other is a result of the velocity of burning propellant grains. At the instant a volume of solid propellant burns, the gas generated has momentum equal to $\Delta m \cdot u_p$, which is properly added to that of the gas in the elemental volume. At the same time, the solid propellant has lost this amount of momentum. Therefore, the rate of momentum exchange is $\dot{m}_{comb}u_p$ and

$$F = -D_x + \dot{m}_{comb} u_p$$

where

$$\dot{m}_{comb} = g(\dot{x}_g, \text{grain geo.}, \phi)$$

The complete momentum equations in conservative form are then:

Gas Phase:

$$\frac{\partial}{\partial t}(\phi \rho u) + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u^2] + \phi\frac{\partial p}{\partial x} = -D_x + \dot{m}_{comb}u_p + \dot{m}_s u_s$$

Solid Phase:

$$\frac{\partial}{\partial t}[(1-\phi)\rho_p u_p] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A(1-\phi)\rho_p u_p^2] + (1-\phi)\frac{\partial p}{\partial x} = D_x$$
$$- \dot{m}_{comb}u_p - \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\sigma(1-\phi)] - \dot{m}_{sp}u_{sp}$$

Energy Equation

The energy equation is derived in terms of the total internal energy, thermal plus kinetic,

$$E = e + u^2/2gJ$$

$$E_p = e_p + u_p^2/2gJ$$

The terms of the global energy equation are as follows:

Total internal energy in the elemental volume:

$$\bar{A}\Delta x[\bar{\phi}\bar{\rho}\bar{E} + (1-\bar{\phi})\bar{\rho}_p\bar{E}_p]$$

Energy flux:

$$[A\phi\rho E u + A(1-\phi)\rho_p E_p u_p]_x - [A\phi\rho E u + A(1-\phi)\rho_p E_p u_p]_{x+\Delta x}$$

Flow work:

$$[A\phi p u + A(1-\phi)p u_p]_x - [A\phi p u + A(1-\phi)p u_p]_{x+\Delta x}$$

Chemical energy due to combustion:

$$\dot{m}_{comb}E_{chem}\bar{A}\Delta x$$

Heat transfer from element (to wall):

$$Q_w \Delta x$$

Work done in compacting solid phase:

$$W_c \bar{A}\Delta x$$

Source and sink terms:

$$\bar{A}\,\Delta x\,[\dot{m}_s H_s + \dot{m}_{sp}(E_{sp} + \frac{p}{\rho_p})]$$

After assembling these terms and dividing by $\bar{A}\,\Delta x$ the global energy equation in conservative form is:

$$\frac{\partial}{\partial t}[\phi\rho E + (1-\phi)\rho_p E_p] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u E + A(1-\phi)\rho_p u_p E_p] + \frac{1}{\bar{A}J}\frac{\partial}{\partial x}[A\phi p u + A(1-\phi p u_p]$$
$$- \dot{m}_{comb} E_{chem} + \frac{Q_w}{\bar{A}} + W_c - \dot{m}_s H_s - \dot{m}_{sp}(E_{sp} + \frac{p}{\rho_p}) = 0$$

This equation is then separated into gas and solid phase energy equations:

Gas Phase:

$$\frac{\partial}{\partial t}[\phi\rho E] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u E] + \frac{1}{\bar{A}J}\frac{\dot{\partial}}{\delta x}[A\phi p u + A(1-\phi)p u_p] - \dot{m}_{comb} E_{chem}$$
$$- \frac{Q_w}{\bar{A}} - \dot{m}_s H_s - \dot{m}_{sp}\frac{p}{\rho_p} = Q$$

Solid Phase:

$$\frac{\partial}{\partial t}[(1-\phi)\rho_p E_p] + \frac{1}{\bar{A}}\frac{\partial}{\delta x}[A(1-\phi)\rho_p E_p u_p] - W_c - \dot{m}_{sp} E_{sp} = -Q$$

where Q is the term representing interaction between the gas and solid phases within the elemental volume. These interaction terms include:

Heat transfer between gas and solid phases

$$Q_p$$

Transfer of kinetic energy from solid to gas phase during combustion

$$\dot{m}_{comb} u_p^2/2gJ$$

Work done by gas drag on moving propellant

$$-\frac{D_x u_p}{J}$$

Therefore, the complete gas phase energy equation is

$$\frac{\partial}{\partial t}[\phi\rho E] + \frac{1}{\bar{A}}\frac{\partial}{\partial x}[A\phi\rho u E] + \frac{1}{\bar{A}J}\frac{\partial}{\partial x}[A\phi p u + A(1-\phi)p u_p] = -\frac{Q_p}{\bar{A}}$$
$$+ \dot{m}_{comb}\left(E_{chem} + \frac{u_p^2}{2gJ}\right) - \frac{Q_w}{\bar{A}} + \dot{m}_s H_s - \frac{D_x u_p}{J}$$

The solid phase energy equation really consists of two separable parts, thermal and kinetic. The kinetic portion contains the solid-phase momentum equation and all solid-phase flow parameters are adequately specified by that equation together with the solid-phase continuity equation. What remains is simply an expression for heat transfer to the propellant together with provision for transport of these heating grains,

$$\frac{\partial}{\partial t}[(1-\phi)e_p] + \frac{1}{A}\frac{\partial}{\partial x}[A(1-\phi)e_p u_p] = \frac{Q_p}{A\rho_p}$$

A subtle feature of the derivation of these equations is that a continuum is the underlying assumption but that the inequality between the elements of the gas and solid phases, i.e., gas molecules vs. propellant grains, is also addressed. Essentially. all pressure and work terms are attributed to the gas phase, whereas in two-phase flow of equal elements, the contribution of these terms would be divided between the phases.

The energy source term, E_{chem}, is particularly important in its interpretation. It represents the total chemical energy liberated during the combustion process plus the heat contained by the solid material at the ignition temperature. Experimentally, the heat of explosion is a reasonable approximation for this parameter.

The BLAKE code is thought to be the most accurate existing mathematical representation of the chemical combustion process and it is desired to use this code to calculate inputs for the interior ballistics code. The output labeled DELTA Q was found to be the difference between the heats of formation of the propellant and combustion products, and represents the chemical heat addition. It carries a negative sign which should be reversed. The sensible heat that should be added to this chemical heat is not well defined but a reasonable approximation is probably $C_V T_{IGN}$, where C_V is the specific heat as given in the BLAKE code output and TIGN is the ignition temperature used in the interior ballistics code. The values of E_{chem} (the sum of the chemical and sensible heats) as determined from a BLAKE code printout for lot A of CASBL M1 propellant are given in Table I. A technique used at NOSIH and BRL[6,7] is to compute

TABLE I

VALUES OF CHEMICAL ENERGY FOR LOT A

Loading Density gm/cc	E_{chem} - cal/gm DELTA Q + $C_v T_{ign}$	$\frac{F}{\gamma - 1}$
0.05	766	831
0.10	766	835
0.15	768	838
0.20	772	845
0.25	778	856
0.30	786	863
0.35	796	875
0.40	808	886

$$E_{chem} = \frac{F}{\gamma - 1}$$

where F is the impetus and γ is the BLAKE code output term called I.B. GAMMA. These values are also tabulated in Table I for comparison.

The source and sink terms, $\dot{m}_s$ and $\dot{m}_{sp}$ are particularly unique to the Calspan code. These represent flow interchange between parallel grid matrices, and are used in the representation of center core ignition and also the gap between bagged propellant charges and the chamber wall. The velocity associated with these terms is given a non-zero value only if it has an X-axis component. Radial components are assumed to have no contribution. The flow work resulting from these source and sink terms is included through use of enthalpy as the energy parameter.

2.4 AUXILIARY RELATIONSHIPS AND TECHNIQUES

2.4.1 Flow Resistance

The resistance to flow through a porous bed, D_x, is represented by

$$\frac{\Delta p}{\Delta x} = \frac{2f(1-\phi)\rho u^2}{\phi_s \phi \; g d}$$

as derived from the expression found in Perry's Chemical Handbook[8]. This expression applies to particle Reynolds numbers in excess of 10^4. The friction factor, f, is close to 0.7 for extremely smooth surfaces such as glass as shown on a graph presented in the reference. However, a value of 1.0 may be more realistic for propellant. A shape factor, ϕ_s, is defined as the quotient of the area of a sphere equivalent to the volume of the particle divided by the actual surface area of the particle. The average particle diameter, d, is similarly defined as the diameter of a sphere of the same volume as the particle. The product, $\phi_s d_2$, reduces to $6V_p/S_p$, the same definition for effective diameter used by Gough[2].

At present, the Calspan code does not distinguish between fluidized and non-fluidized beds. The drag correlation is most important when the bed is in a packed or near-packed condition and the differences between the two bed states are probably swamped by such items as grain deformation, effects of grain porosity, and the influence of combustion, which effectively eliminates skin friction and alters the effective geometric size of the grain. It is recognized that large errors can be generated as porosity approaches unity and that care must be exercised in the regime, particularly in barrel flow where velocities are high.

2.4.2 Heat Transfer

Propellant heating prior to ignition and heat loss to the chamber walls occurs by the three modes; convection, conduction and radiation. Of these, convection provides the major contribution. The relationship used to express convective heating to propellant grains, as presented in Ref. 9, is

$$Nu = 0.3 \, Re^{0.62}$$

The relationship between Nusselt number and Reynolds number was determined empirically from pebble heaters. The conditions of these tests are well defined in terms of flow rate, gas temperature and steady state conditions.

However, it does not seem that this empirical relationship is adequate for interior ballistics codes. This is partly a result of use of a coarse one-dimensional grid network to calculate the flow conditions. This type of network is only capable of representing gross flow patterns and does not adequately represent local eddys and flow patterns that are important to ignition and flame spread. For example, the gas velocity at the breech is computed by the code to be zero and, therefore, the Nusselt number based on Reynolds number is zero. In addition, heat conduction and radiation becomes more significant as pressure increases.

In order to overcome this deficiency, a pressure-dependent correlation was formulated from chamber-heating data measured at Calspan in a 5.56mm fixture[10]. The correlation is

$$h = 0.972 \times 10^{-2} \, p^{0.556}$$

where h, the heat transfer coefficient, is defined as $\dot{q}/\Delta T$ Btu/ft^2-sec-°R. The data were measured near the breech of the chamber where the bulk, one-dimensional velocity is expected to be quite low and apply to pressures up to 50,000 psi. This heat flux quantity is believed to be additive to the Reynolds number-dependent heat flux on the basis that the chamber heat transfer data was observed to increase as a function of distance from the breech. This increase is believed to be the Reynolds number effect. At this time, the magnitudes as they apply to artillery are not known accurately and this represents an area for future research.

The relationship for propellant grain heating is

$$Nu_p = 0.3\, Re^{0.62} + 0.972 \times 10^{-2}\, p^{0.556}\, \frac{d}{k}$$

where d = 6V/S for the propellant grains,

k is the thermal conductivity of the gas,

and p is the pressure in psi.

The heat transfer relationship for chamber wall heating is

$$Nu_w = 0.23\, Re_x^{0.8} + 0.972 \times 10^{-2}\, p^{0.556}\, \frac{d_H}{k}$$

where d_H is now the hydraulic diameter of the propellant-filled cross-section and the turbulent flow heat flux to a pipe wall is represented by $Re_x^{0.8}$.

2.4.3 Propellant Combustion

Propellant combustion in a gun is assumed to occur in a manner similar to that in a closed bomb. Closed bomb-derived burn rates include some of the ignition transient and burning nonuniformities that are present in a gun. These transients occur at different rates and these burn rates may not be entirely representative of the gun case. However, the closed bomb is the primary source of burn rate information for granular propellant as this assumption is more or less imposed.

Basically, the same procedure is used to calculate combustion in the interior ballistics code as in the Calspan closed bomb burn rate code. All exposed surfaces of a propellant grain, including perforations, are assumed to be ignited simultaneously and burn at the same rate. In the finite difference code, this concept is expanded to include all propellant grains within a grid. The grains are assumed to maintain their physical integrity, except for the phenomenon of splintering. Bed compaction, which must, in reality, cause grain deformation and perhaps cracking, is presently allowed to occur in the model without altering the grain geometry.

The possibility of burn rates within the perforations being different from those of the surface is acknowledged but not included in the present model. Recent experiments[11] have shown this assumption to be reasonably accurate and point out the possibility for counteracting effects, such as a flame zone or at least a major portion of the combustion external to the perforation, which would decrease the local heating and, therefore, the surface recession rate inside the perforation. The data in Reference 1 seem to indicate a reduced combustion rate inside perforations which supports this premise. At any rate, the closed bomb-derived burn rate for the actual propellant used in a gun is assumed to include these effects.

The burn rate expression

$$\dot{x} = (A T_o + B) p^n + C T_o$$

has been found to represent the combined effects of pressure and initial temperature on burn rate. Of course, A and C = 0 cause the expression to revert to the familiar Bp^n. Calculation of burn rate is performed separately from the solution of the conservation equations. The calculation procedure involves determining the actual volume change of a grain during the time interval, which is the exact function

$$\Delta V = f(\dot{x}, D, d, L, \Delta t)$$

This is combined with propellant density and porosity to create a mass generation term for the governing equations,

$$\dot{m}_{comb} = (1 - \phi) \rho_p \frac{\Delta V}{V}$$

where $\frac{\Delta V}{V}$ is the fractional change in propellant grain volume, D, d, and L are propellant major diameter, perforation diameter and length, and $\dot{m}_{comb}$, the mass generation term in the conservation equations, is the gas generated per unit of gun chamber volume. For the 105mm howitzer, this technique is applied separately to both single and multiperf propellant grains of the dual granulation charge in both the chamber and barrel.

Eventually, the multiperf grains reach a condition, known as splintering, where burning surfaces coalesce. At this point the calculation becomes less precise for at least two reasons:

1. the geometry is changed drastically and calculation of surface recession is inherently less precise; and
2. the splinters can no longer be considered semi-infinite in depth and actual burn rate is increased due to more rapid temperature rise in a thin section, and generally larger surface heat transfer area in relation to volume.

At the time of splintering, the length is known and the cross-sectional area and total perimeter of the splinters can be calculated from exact geometric relationships. The assumption of equal recession of all surfaces, which may not be accurate as will be seen later, is used for this calculation. The differential volume change is

$$dV = LP\,dx = LP\dot{x}\,dt$$

and the length change is

$$dL = 2\,dx = 2\dot{x}\,dt$$

where P is the total perimeter and dx is recession normal to the surface. The change in cross-sectional area is then,

$$dA_c = dV/L$$

The problem is to functionally relate the cross-sectional area A_c to the perimeter. For a circle or square,

$$A_c = P^2$$

but for a rectangle with one dimension much larger than the other

$$A_c \sim P$$

Actual closed bomb pressure-time data from special M1 propellant lot A for the 105mm howitzer were used in Calspan's burn rate code to assess these relationships. The results are shown in Figure 5. It is noted here that the mass contained in the splinters amounts to about 10% of the total mass of the grain. However, the calculated burn rate for the last 30% of the propellant is noticeably depressed from the Bp^n curve established previously. It is postulated that this depression is a result of calculating a burning surface area larger than actually exists. If this is the case, then splintering and burnout of some grains, perhaps a result of slow or nonuniform ignition, begins quite early in the combustion cycle, uniform recession of all grains is a rather poor assumption, and detailed treatment of splintering involving use of this assumption is probably not warranted. Therefore, while the use of a linear relationship between cross-sectional area and perimeter of the splinters drives the calculated burn rate curve toward the Bp^n curve, the depression of the burn rate curve prior to ideal splintering is far more significant. If revised test or data reduction procedures should explain this depression, then assessment of splintering would be the next logical step. It is hoped that the JANNAF Burn Rate Workshop will shed new light on this problem. At present the relationship, $P^2 \sim A_c$, is contained in the model representation of splinter form function.

Prior to this program, the 105mm howitzer code lumped both propellant granulations into a single mixture defined by length, total cross-sectional area and total perimeter when the propellant flowed into the barrel. Now, the dual granulation feature is retained throughout the ballistic cycle. The propellant grains retain their length and diameter as they enter the barrel and combustion calculations in the barrel are now identical to those performed in the chamber.

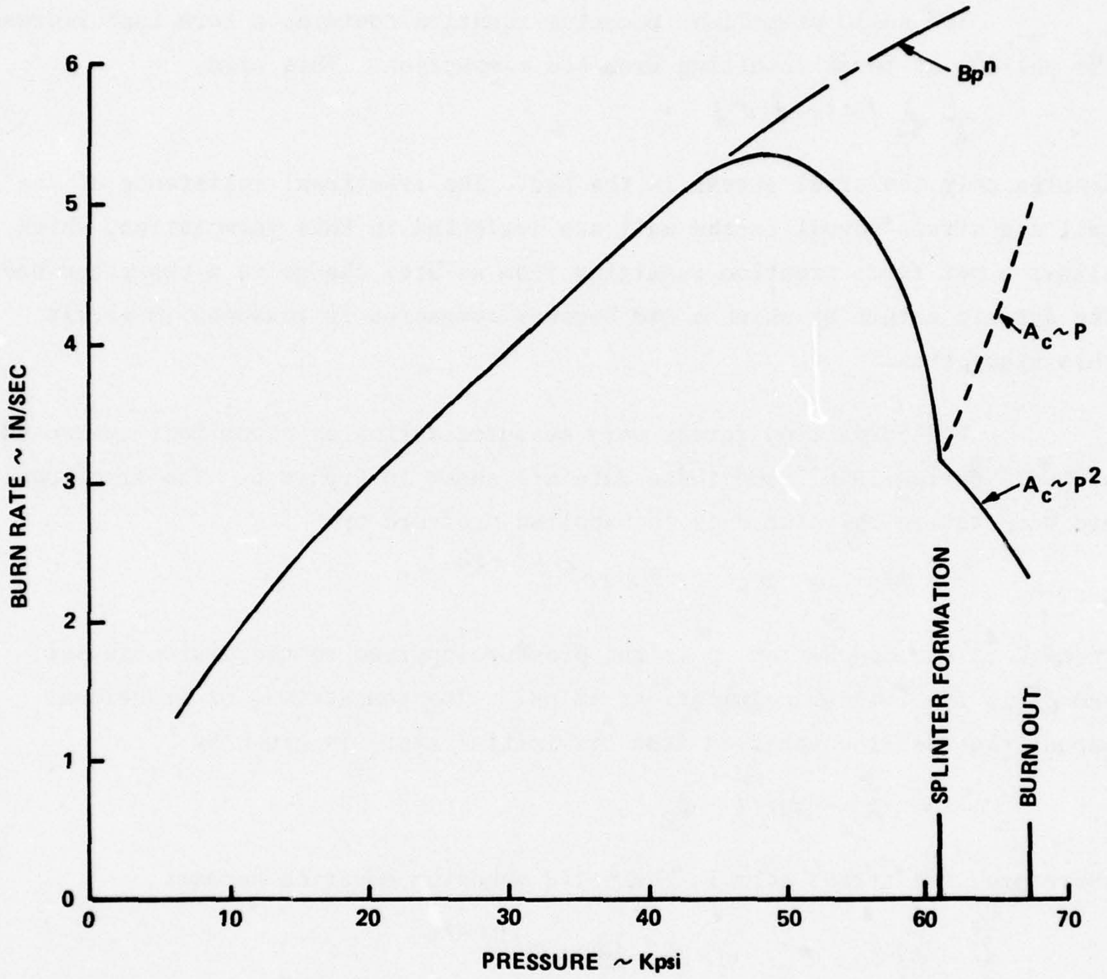


Figure 5 BURN RATE VS PRESSURE FOR SPECIAL M1 PROPELLANT LOT A SHOWING EFFECTS OF CROSS-SECTIONAL AREA-PERIMETER RELATIONSHIP AFTER SPLINTERING

2.4.4 Compaction

The solid propellant momentum equation contains a term that represents the buildup of force resulting from bed compaction. This term,

$$\frac{1}{A} \frac{\partial}{\partial x} [A(1-\phi)\sigma]$$

denotes only the axial stress in the bed. The frictional resistance of the wall and stress normal to the wall are neglected in this formulation, which allows a net force creation resulting from an area change in a compacted bed. The dynamic nature by which a bed becomes compacted is reasoned to permit this assumption.

Bed compaction forces were measured during an experiment conducted at NOSIH during 1976[12] and these data are shown in Figure 6. The fractional bed compaction is related to the applied pressure by

$$(1-\phi)\sigma = p = 1.7 \times 10^4 C^{1.224}$$

where C is the compaction, p is the pressure applied to the piston in psi, and σ is the intergranular stress in psi. The compaction, or fractional amount the bed is compressed from its initial state is given by

$$C = (\phi_o - \phi)/(1 - \phi_o)$$

Therefore, the stress term in the solid momentum equation becomes

$$(1-\phi)\sigma = 1.7 \times 10^4 \left(\frac{\phi_o - \phi}{1 - \phi_o}\right)^{1.224}$$

The Calspan code assumes this to be an elastic stress. This is obviously a deficiency in the code since plastic deformation must occur at high compaction. However, the intergranular stress is most important during initial compaction when pressures and drag forces are lowest. During this initial period, the assumption of elastic deformation is adequate.

Compaction is allowed to proceed until lower porosity limit is reached. This limit is more or less arbitrary. It has been stated[13] that this limit should be the lowest porosity that could be achieved without

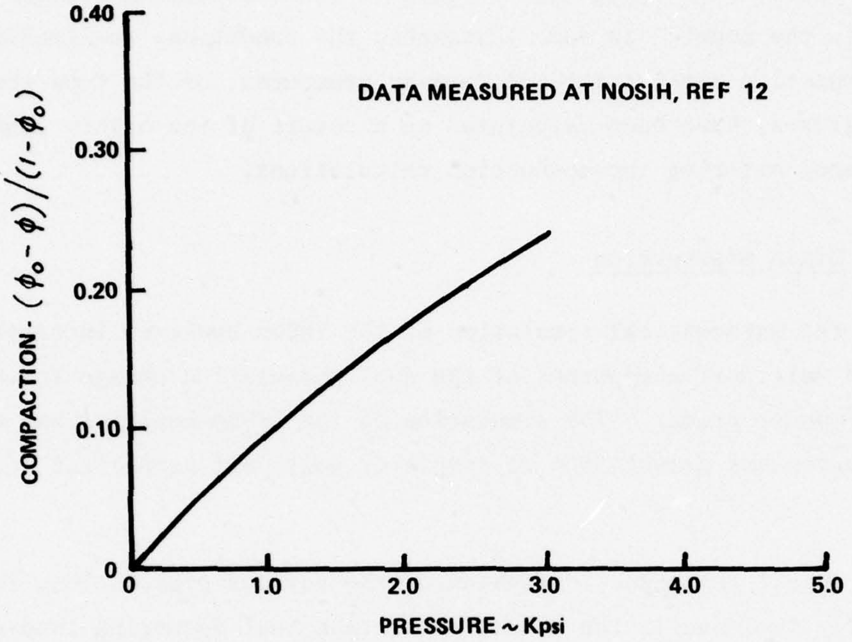


Figure 6 PROPELLANT BED COMPACTION AS A FUNCTION OF PRESSURE APPLIED TO A PISTON

deforming the grains because the model does not contain provision for deformation and breakup phenomena and effects on combustion rate that must occur at lower porosities. We believe the lower porosity limit should allow deformation to occur, even though the physics of the deformation process are not included in the model. In some instances, the conditions required to achieve severe compaction are present and runaway pressures, of the type that cause breech failures, have been calculated as a result of the highly compacted state without altering the combustion calculations.

2.4.5 Grain Segregation

The mathematical simulation of the 105mm howitzer incorporates the single and multiperf components of the dual granulation charge in addition to the black powder primer. The simulation of the 155mm howitzer has the ability to incorporate one granulation of single or multiperf propellant and black powder.

In all instances, diffusion of the various granulations is not specifically included in the model, other than that occurring inadvertently as a result of solution of the governing equations. Diffusion coefficients are quite small because of the size and mass of the granules and this phenomenon is expected to have a negligible effect on the interior ballistics calculations. The original purpose of the models was to provide an analytical technique for charge assessment. Therefore, great pains were taken to account for the masses of the charge constituents when the models were constructed. The single and multiperf charge components of the 105mm howitzer maintain strict segregation for this purpose, that is, to help eliminate inaccuracies in accounting of unburned mass. Segregation is achieved by not allowing propellant of one type to pass from an elemental volume until all of the second type is gone.

2.4.6 Mass Accounting

The finite difference technique used to solve the governing equations of the gas phase causes viscous dissipation-type terms to be introduced as a means of maintaining computational stability. This causes errors to be introduced in gas phase quantities, primarily the mass.

For this reason, the solid phase equations are integrated in a step-by-step manner that maintains accounting accuracy. That is, when a quantity of propellant is burned during a time interval, that quantity is subtracted from the amount existing in the grid at the beginning of the interval. When propellant moves, the amount moving from one grid to the next is physically added to one grid and subtracted from the other in a separate operation. Therefore, it is believed that the solid propellant is accurately accounted for during the ballistic cycle.

The total mass of gas in the system is accounted for and adjusted every calculation time interval. The gas and unburned propellant in each grid is totaled separately. The sum plus the error, E, is set equal to the initial propellant charge and gas

$$m_p = \sum_{i,j} (1-\phi)\rho_p \Delta V_{ij}$$

$$m_g = \sum_{i,j} \phi \rho \Delta V_{i,j}$$

$$m_T = m_p + m_g + \epsilon = m_{p_{init}} + m_{g_{init}}$$

where m_g is the total mass of gas in the system

m_p is the total mass of unburned propellant in the system

ρ_p is propellant density

ρ is gas density, and

V_{ij} is the volume of the V_{ij}^{th} grid.

The error, $\mathcal{E}$, is then distributed over the entire matrix by

$$\rho_{i,j\,corr} = \rho_{i,j}\left(\frac{m_g + \epsilon}{m_g}\right)$$

Thus, a truly constant mass is maintained throughout the ballistic cycle although the mass distribution may be slightly in error.

2.4.7 Treatment of the Solid Phase

The solution of the solid propellant mass and momentum conservation equations is performed in sequential operations by three subroutines in the chamber and two in the barrel. A single subroutine in the chamber and barrel perform the combustion calculation. Another chamber subroutine performs the propellant acceleration calculations of the momentum equation, leading to velocity change, and the third chamber subroutine evaluates the convective terms of both the mass and momentum equations, yielding the final updated propellant properties in each grid at the end of the time interval.

In order to simplify the calculation procedure for the barrel, the assumption was made in the original model formulation that propellant in the barrel only traveled toward the muzzle. The assumption was adequate after the projectile had traveled some distance down the barrel. However, in situations characterized by traveling waves in the chamber during early projectile motion, this assumption was clearly erroneous. Therefore, propellant motion calculations in the barrel were revised and are now the same as those in the chamber.

Briefly, the terms of the propellant conservation equations are evaluated as follows.

a. Combustion:

The combustion or solid mass loss term in the governing equations was given previously as

$$\dot{m}_{comb} = (1-\phi)\,\rho_p \frac{\Delta V}{V}$$

where $\frac{\Delta V}{V} = \frac{S\dot{x}\,\Delta t}{V}$ = the fractional volume change of propellant during a time interval and S is the burning surface area. This is evaluated in Subroutine REGRES of the chamber and DIMIN of the barrel.

b. Grain Acceleration:

The solid phase momentum equation in conservative form, when combined with the solid phase continuity equation yields

$$(1-\phi)\rho_p \frac{\partial u_p}{\partial t} + (1-\phi)\rho_p u_p \frac{\partial u_p}{\partial x} + (1-\phi)\frac{\partial p}{\partial x} = D_x - \frac{1}{A}\frac{\partial}{\partial x}[A\sigma(1-\phi)]$$

The propellant velocity is updated in two stages. This procedure recognizes that a relationship must exist between the speed of sound in the solid grains and the values of Δt and Δx in order to obtain the proper integrated results. Therefore, the convective term is considered separately in the conservative form. While this is not a rigorous mathematical technique, it eliminates some of the smearing that results from direct solution to the above equation in the time frame of the gas phase equations and helps to maintain an accurate accounting of the solid mass. The velocity change for propellant in a grid at the beginning of the time interval is

$$\Delta u_p = \Delta t \left\{ D_x - (1-\phi)\frac{\Delta p}{\Delta x} - \frac{1}{A}\frac{\Delta[A\sigma(1-\phi)]}{\Delta x} \right\}$$

where

$$\frac{\Delta[\;]}{\Delta x} = \frac{[\;]_{x+\Delta x} - [\;]_{x-\Delta x}}{2\Delta x}$$

These calculations are performed in chamber subroutine PRPVEL and barrel subroutine PRØPMØ.

c. Convective Terms:

The convective terms in the solid phase mass and momentum equations are evaluated in chamber subroutine PRØPEL and barrel subroutine PRØPMØ. Here the strict accounting procedure is also followed. The final solid mass in an elemental volume after combustion is simply

$$m_{p_i}^{t+\Delta t} = m_{p_i}^{t} - \dot{m}_{comb_i} - m_{p_{OUT}} + m_{p_{in}}$$

or

$$(1-\phi_i^{t+\Delta t})A_i = (1-\phi_i^{t})A_i - (1-\phi_i^{t})|u_{p_i}|A_i\frac{\Delta t}{\Delta x} + (1-\phi_k)|u_{p_k}|A_k\frac{\Delta t}{\Delta x}$$

where ϕ_i is the porosity in grid i after combustion is considered and K represents adjoining grids with a velocity vector directed toward the i^{th} grid.

Similarly, the convective *momentum term* is included in the final propellant velocity

$$(m_p u_p)_i^{t+\Delta t} = [\, m_p (u_p + \Delta u_p)]_i^t - \dot{m}_{comb\,i}\, u_{pi} - m_p |u_{p\,OUT}| + m_p |u_{p\,in}|$$

In this manner, propellant motion is calculated for each of the grid networks in the gun system, including the black powder center core.

Section 3
MODEL CALCULATION DISCREPANCIES

3.1 OVERVIEW

Extensive use of the 175mm gun code revealed two calculation discrepancies that occur consistently. One was a step pressure discontinuity on the rise portion of the curve. The second discrepancy pertained to the width of the curve, namely the area under the portion of the pressure-time curve where the pressure was greater than half the peak pressure. The problems are illustrated by the computed and experimental pressure curves shown in Figure 7. The discontinuity is characterized by the large spike on the left hand side of the curve and the difference in curve widths is readily apparent. This section discusses the causes of the discrepancies and the means of eliminating them.

3.2 PRESSURE DISCONTINUITY

The pressure discontinuity is observed to occur at the exact time the first grid is added to the barrel matrix. The mechanism that causes the spike is the logic that keeps continuous account of the amount of propellant and gas in the system. An inaccuracy in the technique used to allow initial projectile movement was suddenly corrected when the first barrel grid was added and, therefore, the spike was generated.

Projectile motion and grid addition is illustrated in Figure 8. The first diagram shows the projectile at its rammed position. The projectile base is assumed to be located at the end of the chamber which coincides with the beginning of the barrel. This point is important and will be discussed later. The last chamber grid, designated NGX, is initially half a grid and represents a boundary condition. This condition is currently treated by the mirror image technique which assumes that upstream and downstream conditions are identical in magnitude but opposite in direction. The second and third diagrams show how this last chamber grid stretches from a width Δx to $\Delta x'$

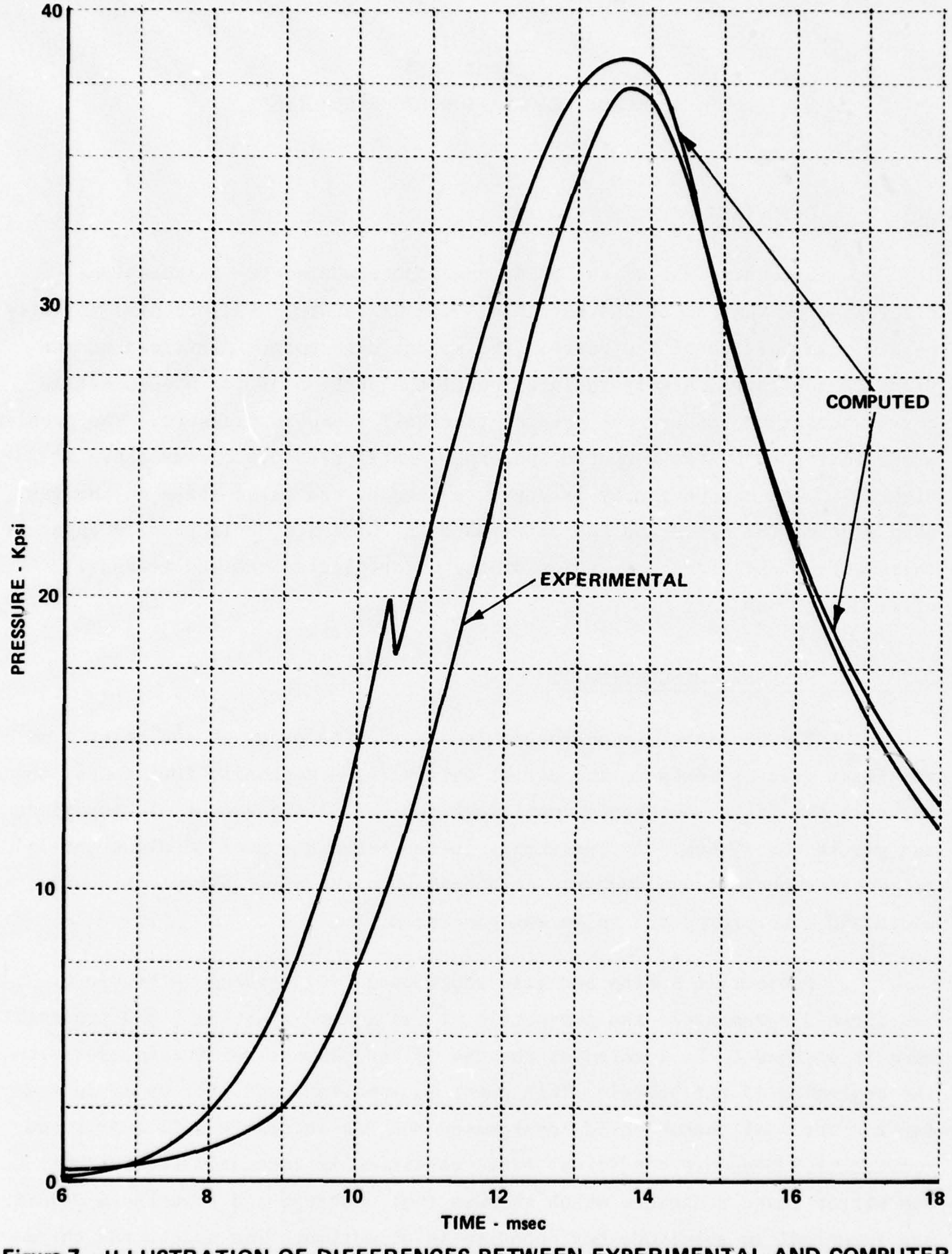


Figure 7 ILLUSTRATION OF DIFFERENCES BETWEEN EXPERIMENTAL AND COMPUTER GENERATED PRESSURE HISTORIES FOR THE M126 155mm HOWITZER

and shifts as the projectile begins to move. The grid NGX is still half size and the mirror image technique is still used with the moving boundary taken into account. Finally, the fourth diagram shows the grid pattern when the projectile has moved the width of one grid, ΔX. At this time, the first barrel grid is added and designated 2. The barrel grid 1 corresponds exactly with chamber grid NGX. At this time, the chamber grid NGX becomes a full grid of width ΔX and barrel grid 2 is half a grid. This sequence is repeated as additional grids are added to the barrel matrix.

Several instances were discovered where the current model did not represent this sequence of events exactly, this especially pertained to the gas and solid accounting procedure when the chamber grid NGX was treated as a whole grid throughout. This caused an error to occur in the volume calculation and is directly responsible for the pressure discontinuity. In addition the treatment of gas and solid propellant accumulation in the grid was in error because it remained fixed and, in effect, the addition of the first barrel grid caused a step change in conditions.

These errors were eliminated through the following steps:

a. The gas and solid propellant mass accounting equations in Subroutine UPDATE were modified so that the volume of grid NGX is now computed by the product A (ΔX' - ΔX/2). ΔX' (see Figure 8) is initially equal to Δx and eventually grow to 2ΔX. This change correctly represents the initial half width condition and eventual 1 1/2 grid size at the time the first barrel grid was added. ΔX' (DXPRIM) is computed in subroutine MOTION.

b. The effects of the change in volume of grid NGX on the quantities that specify the conditions in it; namely, porosity and density, are taken into account in Subroutine MOTION.

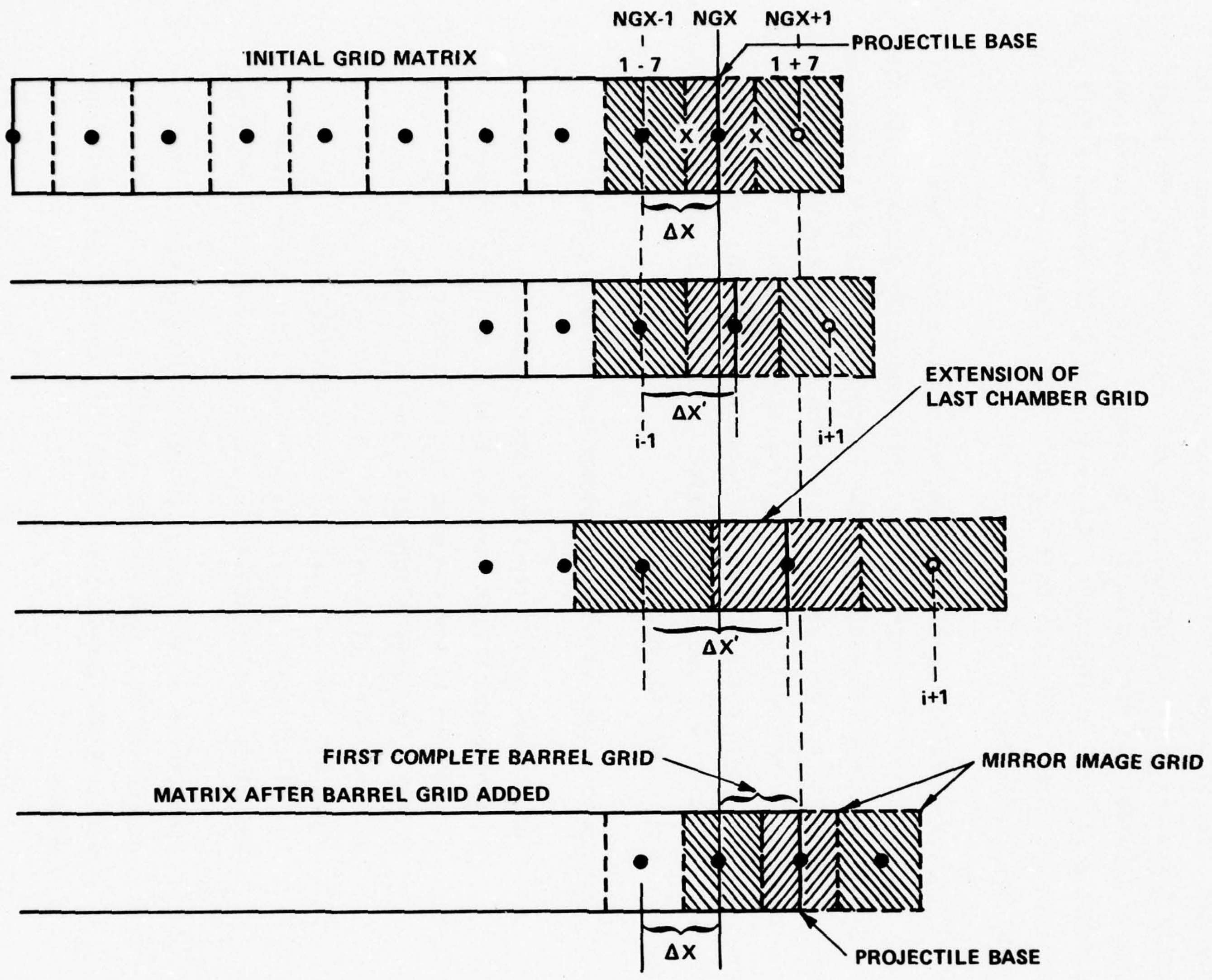


Figure 8 ILLUSTRATION OF INITIAL BARREL GRID ADDITION PROCEDURE

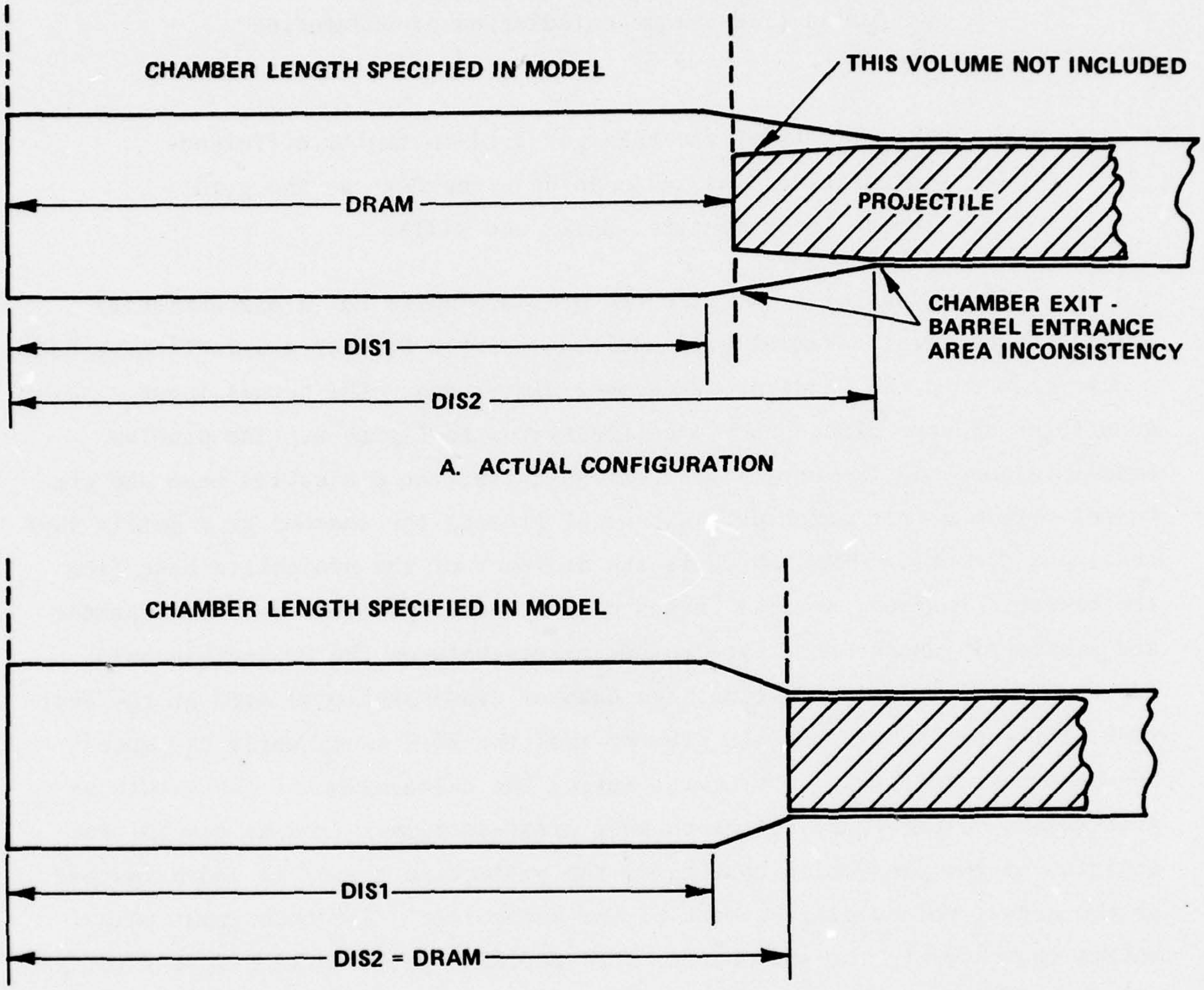
c. The calculation of porosity change due to incoming propellant was updated to include effects of the increased grid volume by use of $\Delta X' - \Delta X/2$ instead of ΔX in grid volume calculation of Subroutine PRØPEL.

d. The effects of the enlarged grid on finite difference calculations was included by using $\Delta X'$ as the grid length in Subroutines AXIT2 and AXIT3.

These changes eliminated the pressure spike but a discontinuity still remained at the barrel grid addition point. Further examination revealed a discrepancy in the handling of chamber dimensions. The actual input quantities and the discrepancy are illustrated in Figure 9. The problem arises because, in the normal gun configuration, the projectile base and the barrel origin do not coincide. The model creates the chamber grid matrix from the input dimension DRAM, which is the distance of the projectile base from the breech. However, several inches of projectile protrude into the chamber and previously unaccounted free volume exists between the projectile base and barrel origin. In addition, the chamber cross-sectional area at the position of the projectile base is greater than the bore area, while the code assumes they are equal. Therefore, during the calculation of the growth of grid NGX, a volume equal to the chamber cross-sectional area at the initial position of the projectile base times the projectile travel is added instead of the actual volume displacement of the projectile. The code input parameters that specify the gun chamber were revised, as shown in Figure 9 so that the projectile base lies at the barrel origin and the chamber volume is initially correct. These changes eliminated the pressure discontinuity as shown in Figure 10.

3.3 AREA UNDER PRESSURE CURVE

While the results shown in Figure 10 indicate that the peak pressure agrees closely with the experimental value, the area under the curve is in


CHAMBER LENGTH SPECIFIED IN MODEL
THIS VOLUME NOT INCLUDED
DRAM
PROJECTILE
CHAMBER EXIT - BARREL ENTRANCE AREA INCONSISTENCY
DIS1
DIS2
CHAMBER LENGTH SPECIFIED IN MODEL
DIS1
DIS2 = DRAM

A. ACTUAL CONFIGURATION

B. EQUIVALENT CHAMBER WITH CORRECT VOLUME AND NO AREA INCONSISTENCIES

Figure 9 ILLUSTRATION OF REQUIREMENTS FOR GUN CHAMBER SPECIFICATIONS IN MODEL

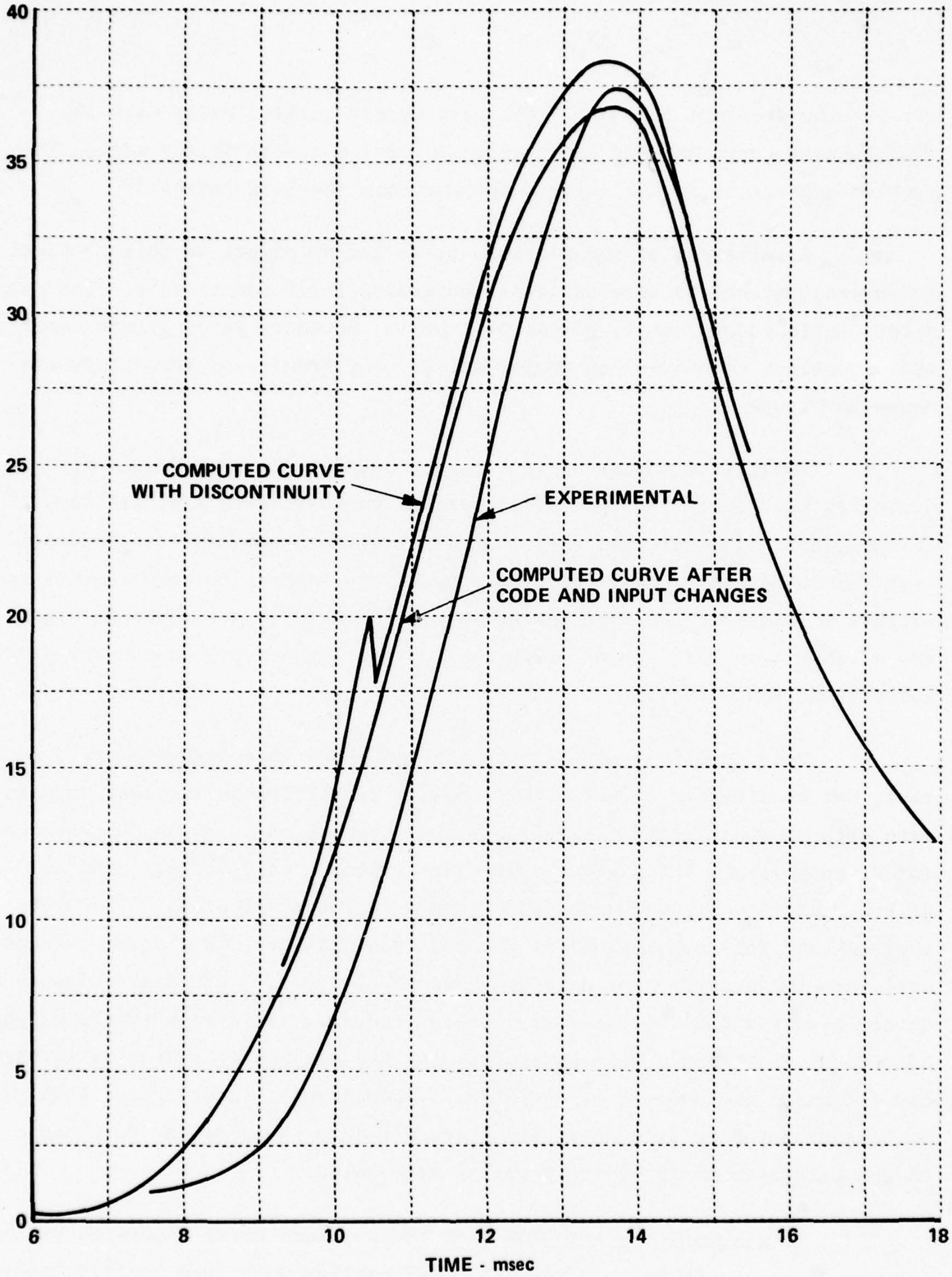


Figure 10 ELIMINATION OF PRESSURE DISCONTINUITY

error. The pressure fall off after peak agrees quite closely with the experimental curve and the pressure rise portion agrees fairly well. The problem appears to be the failure to reproduce the hump correctly.

Elimination of the pressure spike had no effect on this problem so several parameters were varied to determine their sensitivity. The parameters included projectile moment of inertia, boundary layer growth coefficients and propellant combustion characteristics. The results of this study are shown in Figure 11.

Projectile moment of inertia and boundary layer coefficients had virtually no effect. A study of propellant combustion in a closed bomb at Calspan, Ref. 14, indicated that the effective burn rate falls off drastically near burnout during the splintering process. An approximate representation of this fall off as compared to the results of Ref. 14 are shown in Figure 12. Use of this burn rate curve during the splintering process caused the pressure curve to narrow slightly.

Reference 14 also suggests that low burn rate exponents may, in fact, not be accurate. That report shows a nearly linear increase of burn rate with pressure until the fall off near splintering. Strand burner data for M1 propellant, which gives a pressure exponent of 0.91, was used in place of the Picatinny Arsenal data, which had an exponent of 0.654. The shape of the peak and fall off regions of the computed pressure curve agree extremely well with the experimental data as shown in Figure 11. The peak value is a little high and the initial rise is more gradual. It is felt that a slight adjustment of the burn rate parameters and the projectile shot start pressure may eliminate these areas of deviation. Therefore, it does appear that use of closed bomb burn rate data with a low pressure exponent may be a cause of the excessive width of the pressure-time curve.

It was noted by ARRADCOM that the pressure curve generated by the 105mm code became narrower after the modifications described in this report were incorporated. These changes have not been incorporated in the 155mm code at the writing of this report. It is conceivable that the basic formulation of the 155mm model also contributes to the excessive pressure curve width.

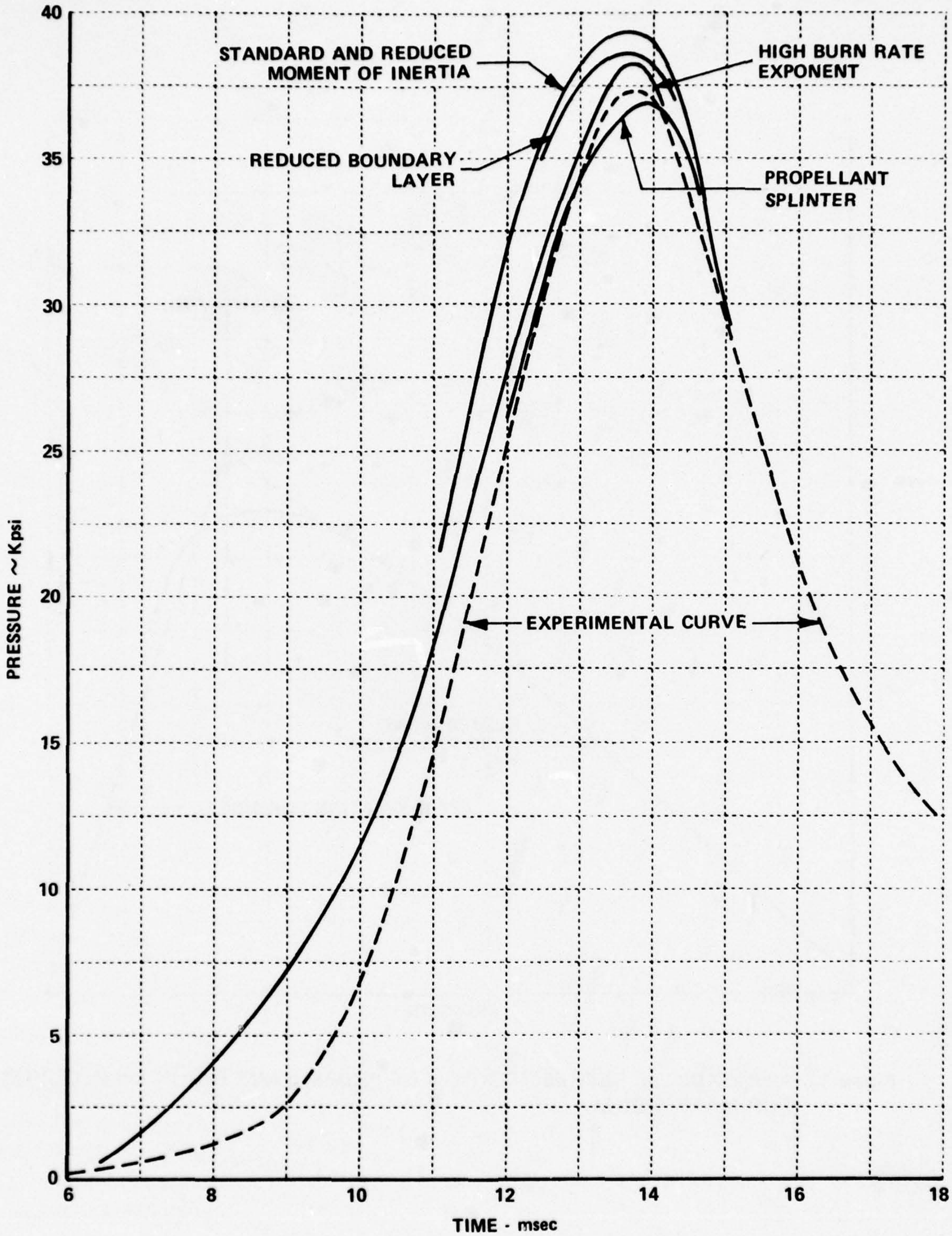


Figure 11 INFLUENCE OF SEVERAL PARAMETERS ON PRESSURE CURVE PROFILE

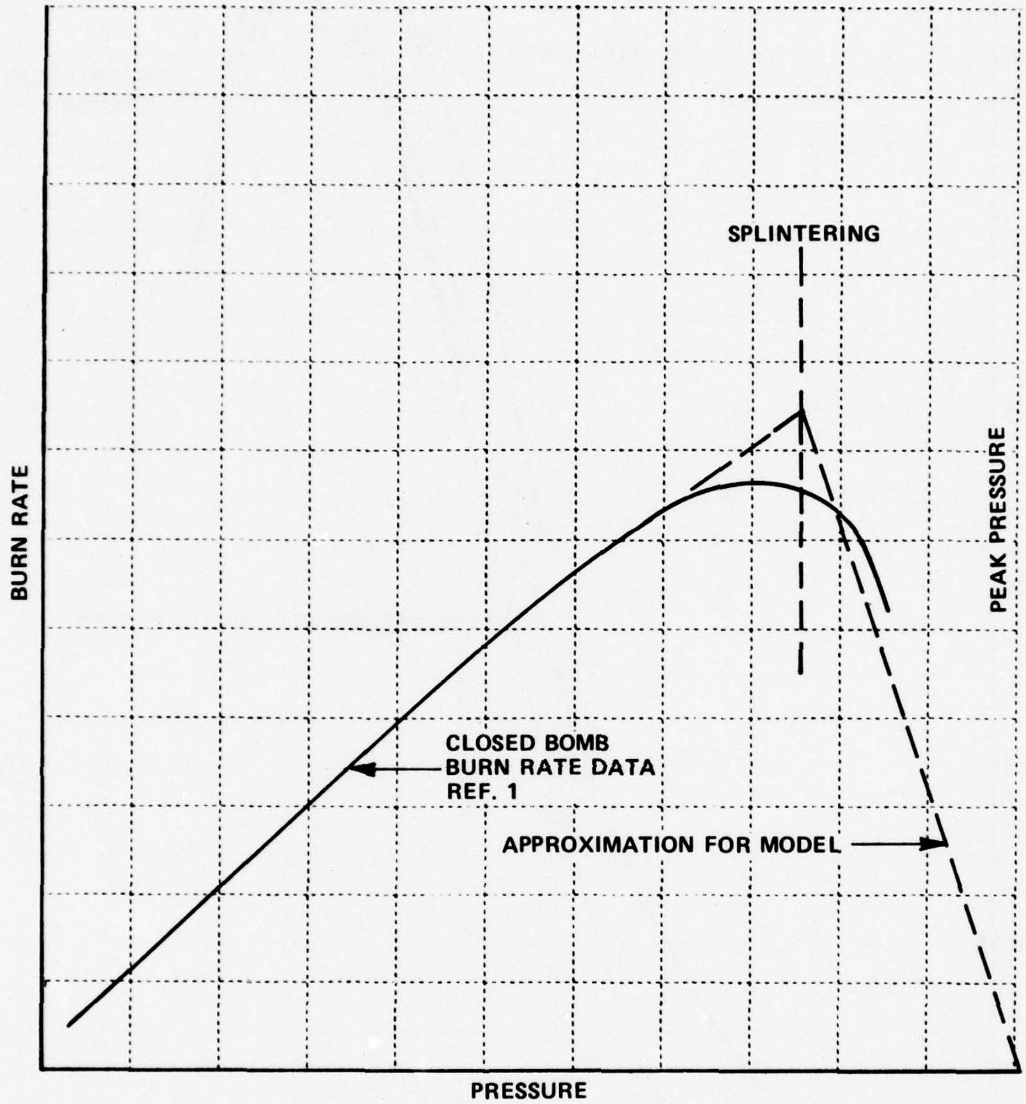


Figure 12 APPROXIMATE REPRESENTATION OF PROPELLANT GRAIN SPLINTERING FOR MATH MODEL

SECTION 4

EXPERIMENTAL/ANALYTICAL CORRELATIONS

A series of eight special lots of M1 multiperf propellant, designated PAD-PE-490-1-A through H for use in the 105mm howitzer were prepared for the purpose of determining the incremental effects of various physical, chemical, and operational factors on propellant performance. This knowledge will be used to help formulate the quality control package to be implemented for the continuous propellant line at Radford Army Ammunition Plant.

The variable factors were quantified in terms of input parameters required by the 105mm howitzer code. Burn rate parameters were determined from closed bomb tests conducted at ARRADCOM. The BLAKE code was used to generate the required energy and equilibrium gas state parameters from chemical analysis information. Other physical data such as grain dimensions and density were obtained by the propellant manufacturer and appear on the description sheet for each lot.

The appropriate input parameters are given in Table II for each special lot (A-H), the reference multiperf lot (68-051), the single perf lot (68-108) and the black powder. The burn rates for the multiperf reference lot and the single perf lot were assumed to be those given in the CPIA-M2 manual (Ref. 15). The black powder burn rate is that given for low pressure in Reference 16. Black powder energy and state parameters represent combination of values from Reference 16 and some unreported closed bomb data at Calspan. Other program inputs are given in Appendix C.

The results of these calculations are compared with experimental firing data obtained by making the special propellant into M67 charges and firing an M1 projectile from an M2A2 105mm howitzer in Table III and Figure 13. It is seen that the eight special propellant lots fall into two groups according to perforation diameter, those with large perforation diameters giving substantially higher performance than those with smaller perforation diameters. In general, the computed results for lots A-D are higher than

TABLE II

COMPUTER CODE INPUTS FOR SPECIAL PROPELLANT LOT SIMULATIONS

Lot → Input ↓	A	B	C	D	E	F	G	H	REF (68051)
AOMP	33.635	33.832	33.533	33.754	33.568	33.579	33.616	33.564	33.968
A1MP	-26.366	-25.879	-26.286	-25.931	-26.250	-26.083	-26.040	-25.995	-25.949
A2MP	20.847	19.251	21.209	19.803	20.958	20.755	20.563	20.687	18.676
A3MP	-20.176	-19.178	-20.659	-19.649	-20.471	-20.301	-20.215	-20.224	-18.350
BGEN	0.00509	0.00557	0.00448	0.00650	0.00336	0.00280	0.00359	0.00385	0.00214
DIO	0.0089	0.0086	0.0085	0.0085	0.0235	0.0220	0.0223	0.0223	0.0137
DIR1	0.0089	0.0086	0.0085	0.0085	0.0235	0.0220	0.0223	0.0223	0.0137
DØO	0.1402	0.139	0.1405	0.1407	0.1429	0.1420	0.1400	0.1400	0.1387
DØR1	0.1402	0.139	0.1405	0.1407	0.1429	0.1420	0.1400	0.1400	0.1387
GAMMP	1.259	1.258	1.257	1.258	1.257	1.257	1.257	1.256	1.260
HMB	1,521	1,548	1,555	1,556	1,562	1,586	1,585	1,596	1,531
PEXP	0.645	0.650	0.668	0.640	0.693	0.715	0.694	0.691	0.710
RHØP	97.5	98.0	98.2	97.7	97.3	97.7	97.9	98.1	97.5
RHØPIR	97.5	98.0	98.2	97.7	97.3	97.7	97.9	98.1	97.5
WMMP	22.14	22.18	22.27	22.24	22.26	22.33	22.32	22.38	22.04
XLO	0.3178	0.3164	0.3175	0.3178	0.3180	0.3180	0.3190	0.3180	0.3202
XLR1	0.3178	0.3164	0.3175	0.3178	0.3180	0.3180	0.3190	0.3180	0.3202

TABLE II (CONT.)

COMPUTER CODE INPUTS FOR SPECIAL PROPELLANT LOT SIMULATIONS

Single Perf Propellant			Black Powder		
AOSP	=	34.192	AOBP	=	15.07
A1SP	=	-25.578	A1BP	=	0.
A2SP	=	16.500	A2BP	=	0.
A3SP	=	-16.214	A3BP	=	0.
BGEN2	=	0.00214	AGENBP	=	0.744
DI02	=	0.0198	EXPBP	=	0.24
DIR2	=	0.0198	GAMBP	=	1.08
DØ02	=	0.0467	HBP	=	1,375
DØ122	=	0.0467	WMBP	=	75.
GAMSP	=	1.261			
HMB2	=	1,531			
PEXP2	=	0.71			
RHØP2	=	97.5			
RHØP2R	=	97.5			
WMSP	=	21.97			
XL02	=	0.199			
XLR2	=	0.199			

TABLE III

COMPARISON OF COMPUTED AND EXPERIMENTAL
105MM HOWITZER PERFORMANCE FOR EIGHT LOTS
OF SPECIAL PROPELLANT

Lot	Peak Pressure Psi		Muzzle Velocity Ft/Sec	
	Comp.	Exp	Comp.	Exp.
A	30400	28800	1482	1443
B	34300	31000	1533	1491
C	32700	30000	1516	1484
D	35500	32400	1547	1510
E	45600	41700	1591	1583
F	46200	45300	1603	1605
G	49400	46400	1615	1614
H	51400	47200	1624	1618
REF	28800	34200	1454	1532

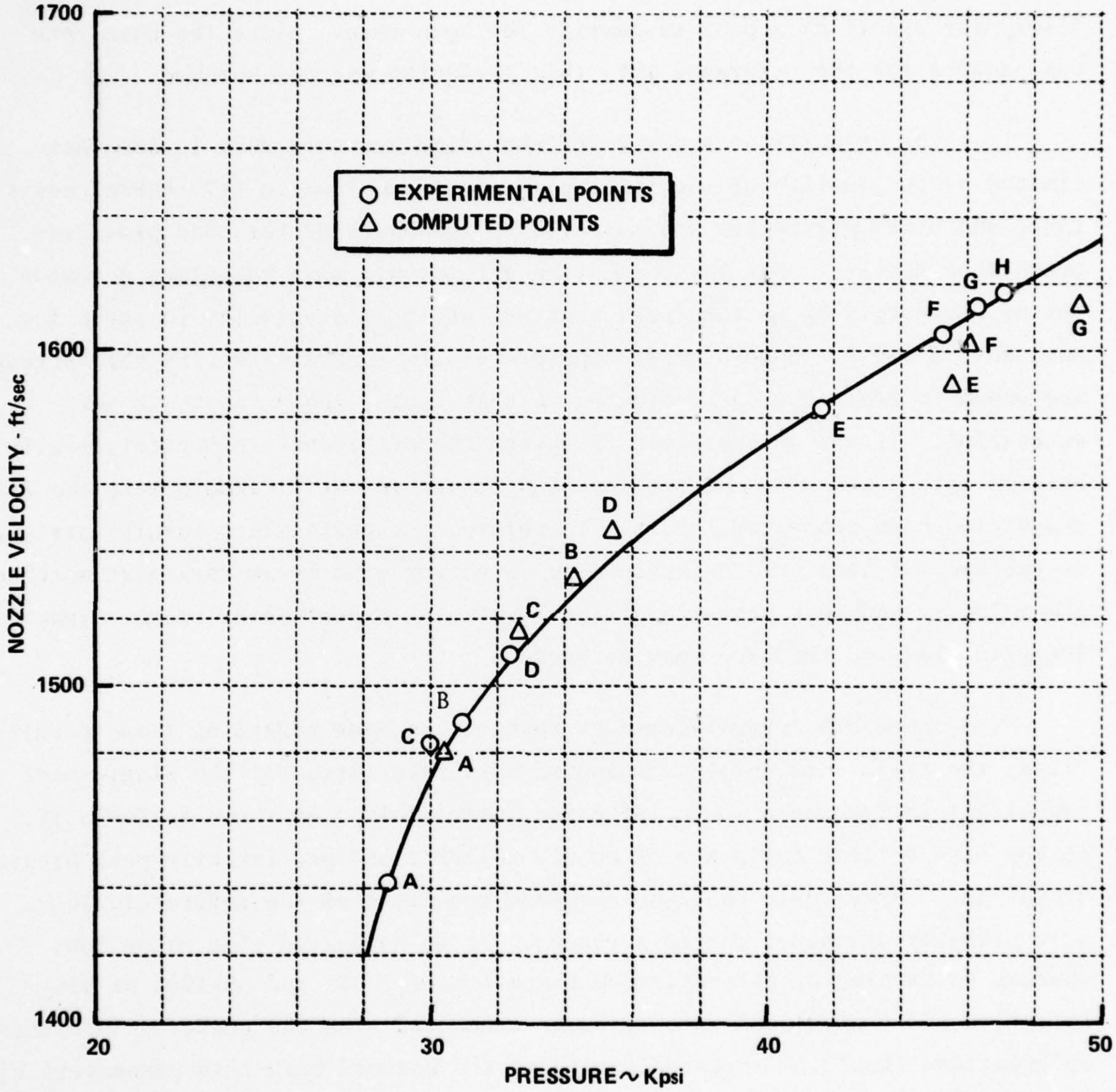


Figure 13 COMPARISON OF COMPUTED DATA WITH FIRING DATA USING EIGHT SPECIAL LOSTS OF MIMP PROPELLANT IN THE M2A2 105mm HOWITZER

the experimental results, with respect to both pressure and muzzle velocity. For lots E-H, the muzzle velocities are fairly close but the pressures are slightly higher than experimental values. The reference lot was not close, likely the result of a poor assumption for burn rate. Since the burn rate was assumed for the reference lot, this deviation was not pursued.

The calculation accuracy of the 105mm howitzer code is somewhat limited by the quality of the input parameters supplied to it. Barrel resistance and other parameters are selected on the basis of the code providing acceptable results. For these runs, an attempt was made to select a common set of parameters by getting peak pressure and muzzle velocity to agree for both lots A and F. Some of the computer results generated during this attempt are shown in Table IV. As indicated in that table, the attempt was not successful. It was not possible to spread the difference in muzzle velocity between lots A and F to the required 160 ft/sec and still incorporate the BLAKE and closed bomb generated inputs. Therefore, the performance results assigned to the special lots were determined by selecting code parameters that matched the velocity and peak pressure of lot F with the experimental value. Indeed, lot F is close to the experimental curve.

There are several comments that can be made regarding these results. First, the failure to obtain a measured burn rate curve for the single-perf propellant introduces a source of error immediately. As shown in Table IV, it can have a large influence on muzzle velocity and particularly peak pressure. It was noted previously that the computed results from the referenced lot were probably erroneous for this reason. It is suggested that propellant charges containing propellants from these lots (68-051 and 68-108) be disassembled and that closed bomb tests be conducted with the powder. Then these calculations should be repeated replacing the assumed burn rate parameters with measured ones.

Secondly, it is noted that differences in grain outside and perforation diameters of 0.001 or 0.002 inch have a large impact on computed results. A decrease of 0.002 inch in perforation diameter dropped the computed peak

TABLE IV

COMPUTED PEAK PRESSURE AND MUZZLE VELOCITY FOR A VARIETY OF COMPUTER CODE INPUTS

Lot	P_{max} psi	mv ft/sec	PZ0 psi	PDMAX psi	PINT psi	PLØ psi	CF	BGEN2 x 10^2	HMB Btu/lbm	H.F. B/F^2sec	DØ in	DI in
A	31900	1419	500	500	100	15	1.0	0.214	1509	x 2	0.1402	0.0
A	29840	1465	500	500	100	15	0.0	0.214	1509	x 2	0.1402	0.0
F	45380	1580	500	500	100	15	0.0	0.214	1555	x 2	0.142	0.022
F	48400	1520	500	500	100	15	1.0	0.214	1555	x 2	0.142	0.022
F	43870	1566	0	100	1000	15	0.0	0.214	1555	x 2	0.142	0.022
A	28400	1455	0	100	1000	15	0.0	0.214	1509	x 2	0.1402	0.0089
F	52400	1440	500	500	100	15	2.0	0.214	1555	x 2	0.142	0.022
F	41250	1675	500	500	100	15	-2.0	0.214	1555	x 2	0.142	0.022
A	26990	1539	500	500	100	15	-2.0	0.214	1509	x 2	0.1402	0.0089
A	27390	1501	500	2000	1000	1000	-2.0	0.214	1509	x 2	0.1402	0.0089
A	28420	1457	500	3000	2000	2000	-2.0	0.214	1509	x 2	0.1402	0.0089
A	28500	1405	500	3000	3000	3000	-2.0	0.214	1509	x 2	0.1402	0.0089
F	42770	1540	500	3000	3000	3000	-2.0	0.214	1555	x 2	0.142	0.022
F	43929	1545	1000	4000	3000	3000	-2.0	0.214	1555	x 2	0.142	0.022
A	29670	1440	500	1500	1000	1000	-0.5	0.214	1509	x 2	0.1402	0.0089
A	28890	1459	500	1500	1000	1000	-1.0	0.214	1509	x 2	0.1402	0.0089
A	24660	1437	0	0	0	0	0	0.170	1509	x 2	0.1402	0.0089
F	39400	1557	0	0	0	0	0	0.170	1555	x 2	0.142	0.022
A	26000	1445	0	0	0	0	0	0.190	1509	x 2	0.1402	0.0089
F	41000	1563	0	0	0	0	0	0.190	1555	x 2	0.142	0.022
F	45800	1580	0	0	0	0	0	0.214	1555	x 2	0.142	0.022

TABLE IV (CONT.)

COMPUTED PEAK PRESSURE AND MUZZLE VELOCITY FOR A VARIETY OF COMPUTER CODE INPUTS

Lot	P_{max} psi	mv ft/sec	PZ0 psi	PDMAX psi	PINT psi	PLØ psi	CF	BGEN2 x 10^2	HMB Btu/1bm	H.F. B/F^2-sec	DØ in	DI in
F	60200	1812	0	0	0	0	0	0.214	2022	x 2	0.142	0.022
A	38600	1682	0	0	0	0	0	0.214	1962	x 2	0.1402	0.0089
F	43920	1575	0	0	0	0	0	0.214	1555	x 2	0.142	0.022
A	27740	1458	0	0	0	0	0	0.214	1509	x 2	0.1402	0.0089
A	30400	1482	1000	0	0	0	0	0.214	1521	x 2	0.1402	0.0089
F	46200	1603	1000	0	0	0	0.0	0.214	1586	x 2	0.142	0.022
F	47350	1625	1000	0	0	0	0.0	0.214	1586	/5	0.142	0.022
A	31380	1505	1000	0	0	0	0.0	0.214	1521	/5	0.1402	0.0089
F	48460	1629	1000	0	0	0	0.0	0.214	1586	/5	0.140	0.022
F	44640	1614	1000	0	0	0	0.0	0.214	1586	/5	0.142	0.020

pressure by nearly 3000 psi and the muzzle velocity by 11 ft/sec for lot F. It is expected to be even greater for lot A where the perforation diameter is less than half of that of lot F. Therefore, it is suggested that extra care be given to characterize the mean and standard deviation of perforation diameter in the same manner it is done for outside diameter and length.

Finally, results being developed under the auspices of the JANNAF Burn Rate Workshop indicate that closed bomb results are not adequate for use in computer codes. Computed burn rates from a current workshop data reduction exercise indicate that loading density has a strong influence on the burn rate curve as shown in Figure 14. In essence, the effect of loading density is believed to place the propellant grains at different pressure levels for a given percentage of surface recession. Thus, effects of combustion variations at different locations on the exposed surface, i.e., in perforations or on the outside surface, become apparent. This is an extremely important phenomenon that must be understood if the model can be made to become a predictive device.

Therefore, the results of this program can be summarized by stating:

1. The model has been definitely improved through reformulation and by giving better theoretical basis to its inputs.
2. The model is still hampered in its usefulness by inadequately defined input parameters and an unusual lack of understanding of certain basic combustion phenomena.

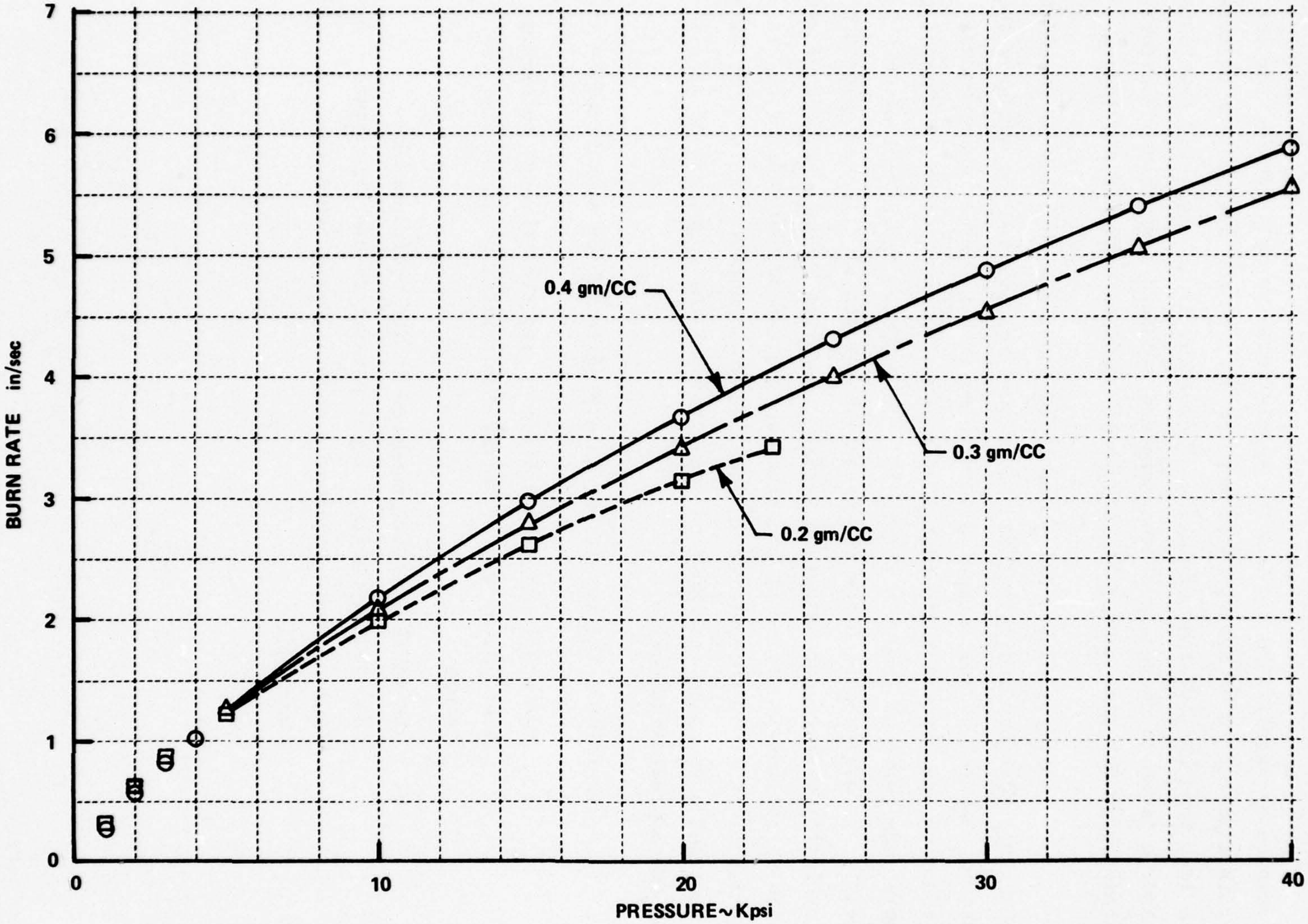


Figure 14 COMPUTED BURN RATES AT THREE CLOSED BOMB LOADING DENSITIES

Section 5

REFERENCES

1. Fisher, E.B., "Propellant Ignition and Combustion in the 105mm Howitzer," Picatinny Arsenal Contract No. DAAA21-74-C-0401, Calspan Corporation Report No. VQ-5524-D-1, January 1975.

2. Fisher, E.B., and Trippe, A.P., "Development of a Basis for Acceptance of Continuously Produced Propellant," Picatinny Arsenal Contract No. DAAA21-72-C-0577, Calspan Corporation Report No. VQ-5163-D-1, November 1973.

3. Fisher, E.B., and Trippe, A.P., "Mathematical Model of Center Core Ignition in the 175mm Gun," Picatinny Arsenal Contract No. DAAA21-72-C-0577, Calspan Corporation Report No. VQ-5163-D-2, March 1974.

4. Culick, F.E.C., "Conservation Equations for a Reacting Two-Phase Flow Based on a Continuum Theory," Notes presented at the JANNAF Combustion Workshop, August 14, 15, 1975.

5. Gough, P.S., "Computer Modeling of Interior Ballistics," Indian Head Contract Report 75-3, 15 October 1975.

6. Personal communication with A. Horst, BRL, 5/4/77.

7. Personal communication with F. Robbins, NOSIH, 5/24/77.

8. Chemical Engineers' Handbook, Fifth Edition 1973, R.H. Perry, editor, McGraw-Hill Book Co.

9. Runacres, A., and Fisher, S.A., "Thermodynamics Design of Regenerative Pebble Bed Heat Exchanger," Aeronautical Research Labs, Mechanical Engineering Note 281, September 1966.

10. Fisher, E.B., "Determination of Temperature Gradients in 5.56mm Aluminum Cases," Calspan Report No. GM-2962-Z-1, June 1972.

11. Fisher, E.B., "Investigation of Ignition Effects on Closed Bomb Test Results," Calspan Report No. VN-5891-X-1, Prepared for Naval Ordnance Station, Indian Head, Maryland, March 1977.

12. Personal communications with A. Horst, NOSIH, July 1976.

13. Personal communications with C. Nelson, BRL, August 1976.

14. Fisher, E.B., and Trippe, A.P., "The Hi-Lo Bomb--A New Device to Evaluate Propellant Combustion Properties," Calspan Internal Research Report No. 85-477, March 1975.

REFERENCES (CONT.)

15. "CPIA/M2 Solid Propellant Manual," Compiled by Solid Propellant Information Agency, Applied Physics Laboratory, John Hopkins University, Silver Spring, Maryland. CONFIDENTIAL.

16. Urbanski, T., "Chemistry and Technology of Explosives," Vol. III, Pergamon Press, 1967.

APPENDIX A

NOMENCLATURE

A	Cross-sectional area of elemental volume
C_v	Constant volume specific heat of combustion products
D_x	Drag force over the length of an elemental volume
e	Gas static internal energy
E	Gas total internal energy
E_{chem}	Total static heat contained in gas generated during combustion
e_p	Static or thermal energy contained in solid propellant
E_p	Sum of thermal and kinetic energy of solid propellant
$\dot{m}_{comb}$	Rate of gas generation during combustion
$\dot{m}_s$	Gas entering elemental volume through side walls
$\dot{m}_{sp}$	Propellant entering elemental volume through side walls
Nu	Nusselt number
p	Pressure
Q_p	Heat transfer to propellant surface
Q_w	Heat transfer to chamber wall
R	Gas constant
Re	Reynolds number
Sp	Propellant grain surface area
T	Temperature
t	Time
u	Gas velocity
u_p	Propellant velocity
u_s	Velocity in x-direction of gas entering volume through side wall
u_{sp}	Velocity in x-direction of propellant entering volume through side wall

V_p	Propellant grain volume
x	Coordinate along axis of elemental volume
γ	Ratio of specific heats
Δx	Length of elemental volume
ϕ	Porosity
ρ	Gas density
ρ_p	Propellant material density
σ	Compacted, propellant bed stress

BEST AVAILABLE COPY

APPENDIX B

FORTRAN IV - MACHINE LISTING
105MM HOWITZER CODE

```
      PROGRAM M437(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
C     MAIN
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),
     1THICK(60)
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BOREDB,DT2BD,DIDSG,XLBAR
      COMMON/CALLP/BPLEFT
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPCAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELFLG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/DETNX/NXSAVE
      COMMON/CONS/DTCX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/MOCON/CON3,CON4,CON5,AREAPB,Z0,WCB,XUB,FDMAX,PINER,
     $    CF,RADPB,PMASS,XINT,PINT,XLO,PLO,CON6
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/PROMO/FIRE
      COMMON/SPLINT/WHOLEC,WHOLEB
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      COMMON/PAC/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1    VBG(60,5), UPB(60,5), FCH(60,5), TZC(60,5),
     2    DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3    PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4    VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBC2(60,5)
      LOGICAL PRI1,IDEBUG
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL FIRE
      LOGICAL FAIL
      LOGICAL BPLEFT
      LOGICAL WHOLEC,WHOLEB
C
C
      READ(5,1000) IDEBUG
      WRITE(6,2003)
      WRITE(6,2004)
      WRITE(6,2002) IDEBUG
      IPRINT = 0
      NAMELIST/MODS/MODCH,MODGR
      READ(5,MODS)
      IF(IDEBUG(1)) WRITE(6,MODS)
```

BEST AVAILABLE COPY

```
C
C     WHEN BLACK POWDER AT ALL GRIDS IS IGNITED, BPIGN WILL BE SET TRUE
C     IN BPFIR.
       BPIGN = .FALSE.
C
C     WHEN PROPELLANT AT ALL GRIDS IS IGNITED, IGNIT WILL BE SET TRUE
C     IN PRPFIR.
       IGNIT = .FALSE.
       FAIL = .FALSE.
C
C*      FAIL WILL BE SET FALSE IN TUBFAL UNTIL THE TUBE HAS
C*      FAILED COMPLETELY(MFAIL(I) = 2) ALONG ITS ENTIRE LENGTH.
C
C
C     ONED IS FALSE UNTIL TWO CONDITIONS ARE SATISFIED AND THE CHAMBER
C     IS MADE 1-DIMENSIONAL
C               1.  PROPELLANT IS IGNITED AT ALL GRIDS, I.E. IGNIT IS TRUE
C               2.  A POROSITY CONDITION IS SATISFIED
       ONED = .FALSE.
C
C     WHOLEC WILL BE SET FALSE IN REGRES WHEN PROPELLANT GRAINS SPLINTER
C     IN AT LEAST ONE CHAMBER GRID.  WHOLEB ACTS SIMILARLY IN THE BARREL.
       WHOLEC = .TRUE.
       WHOLEB = .TRUE.
C
C     BPLEFT WILL BE SET FALSE IN PRIMER WHEN ALL THE BLACK POWDER HAS
C     BEEN BURNED.
       BPLEFT = .TRUE.
       FIRE=.FALSE.
C
C     FIRE WILL BE SET TRUE IN PRPFIR WHEN PROPELLANT IN AT LEAST ONE
C     GRID IGNITES.
       CALL CHSET
       CALL BARSET
       MXSAV1 = MXSAVE + 1
C
C
C
C**************************************************************
C**************************************************************
C**************************************************************
C
 10    CONTINUE
C *** DELETED LOGIC CALLING TUBFAL.
      PRI1 = .FALSE.
      IF(MOD(IPRINT,MODCH) .EQ. 0) PRI1 = .TRUE.
C
C**************************************************************
C     SETTING DUMMY GRIDS
C**************************************************************
```

BEST AVAILABLE COPY

```
C     SET DUMMY CHAMBER GRIDS I = NGX+1 THE SAME AS BARREL GRID 2
C     AND DUMMY BARREL GRID 1 AS THE VOLUME-WEIGHTED AVERAGE OF CHAMBER
C     GRIDS I = NGX.
C     VBG IS 0.0 AT CHAMBER GRID NGX+1
C     AREAS AT BARREL GRID 1 WERE SET TO BOREA IN CHSET
C
       NP1 = NGX+1
       IF(NX .GE. 2) GO TO 140
       NM1 = NGX - 1
       DO 130 J=1,NGR
           RHOBG(NP1,J) = RHOBG(NM1,J)
           UBG(NP1,J) = VP
           HBG(NP1,J) = HBG(NM1,J)
           PCH(NP1,J) = PCH(NM1,J)
           PHIBG(NP1,J) = PHIBG(NM1,J)
           PHIBG2(NP1,J) = PHIBG2(NM1,J)
           UPB(NP1,J) = UPB(NM1,J)
  130  CONTINUE
       GO TO 155
C
C
  140  CONTINUE
C ***ONLY NEED VALUES AT GRID (NP1,2) SINCE GRID (NGX,1) WILL NOT
C     INTERACT WITH THE BARREL.  LET PHIBG(NP1,2) GET THE TOTAL
C     POROSITY SINCE THE TOTAL POROSITY SHOULD BE IN PHIBG FOR THE PATH
C     SUBROUTINES.
       J = 2
           RHOBG(NP1,J) = RHOG(2)
           UBG(NP1,J) = UG(2)
           HBG(NP1,J) = HG(2)
           PCH(NP1,J) = PG(2)
           PHIBG(NP1,J) = PHI(2)
        PHIBG2(NP1,J)=PHI2(2)
           PHIBP(NP1,J) = 1.0
           UPB(NP1,J) = UP(2)
       AREAR(NP1)=AMASS(2)
C
C
  155  CONTINUE
C
C ***(DUMMY) BARREL GRID 1 IS ACTUALLY CHAMBER GRID (NGX,2)
       RHOG(1) = RHOBG(NGX,2)
       UG(1) = UBG(NGX,2)
       HG(1) = HBG(NGX,2)
       PG(1) = PCH(NGX,2)
       PHI(1)=1.0-(1.0-PHIBG(NGX,2))*AREAR(NGX)/BOREA
       PHI2(1)=1.0-(1.0-PHIBG2(NGX,2))*AREAR(NGX)/BOREA
       UP(1) = UPB(NGX,2)
C
  180  CONTINUE
C
C
```

BEST AVAILABLE COPY

```
C***********************************************************************
C     CHAMBER SUBROUTINES
C***********************************************************************
C
      IF(.NOT. BPIGN) CALL BPFIR
C     LOGICAL VARIABLE BPLEFT WILL BE SET TO FALSE IN PRIMER WHEN ALL
C     BLACK POWDER IS BURNED.
      IF(BPLEFT) CALL PRIMER
      IF(.NOT. ONED) CALL MFLOW
C
C     PRPFIR IS CALLED UNTIL ALL PROPELLANT IS IGNITED.
      CALL PRPFIR
      CALL REGRES
      CALL PRPVEL
      CALL PROPEL
C
C
C     PUT THE TOTAL POROSITY INTO ARRAY PHIBG FOR USE IN THE PATH
C     SUBROUTINES.
      DO 190 J=1,NGR
      DO 190 I=1,NGX
      PHIBG(I,J) = PHIBG(I,J) + PHIBG2(I,J) + PHIBP(I,J) - 2.0
 190  CONTINUE
C     PATH SUBROUTINES
      XB = -DX
C
      DO 300 IX = 1,NGX
      DO 300 IR=1,2
          IPA = IPATH(IX,IR)
      GO TO (201,202,203,204,205) IPA
 201  CALL AXIS
          GO TO 300
  202 CALL AXIT2
          GO TO 300
  203 CALL BSURA2
          GO TO 300
  204 CALL FSURT2
          GO TO 300
  205 CALL BSURT2
          GO TO 300
 300  CONTINUE
C
      UUG(1)=UBGTD(NGX,2)
      UHG(1)=HBGTD(NGX,2)
      UUP(1)=UPBDT(NGX,2)
       URHOG(1)=RHOBTD(NGX,2)
C
C     FIX ARRAY PHIBG SO THAT IT ONLY REPRESENTS POROSITY OF THE
C     PROPELLANT AND NOT THE TOTAL POROSITY.
      DO 310 J=1,NGR
      DO 310 I=1,NGX
```

BEST AVAILABLE COPY

```
      PHIBG(I,J) = PHIBG(I,J) + 2.0 - PHIBP(I,J) - PHIBG2(I,J)
 310  CONTINUE
C
C*****************************************************************************
C    BARREL SUBROUTINES
C*****************************************************************************
C
      IF(NX .LT. 2) GO TO 350
      CALL DIMIN
      IF(NX .GT. NXSAV1) CALL BNDLYR
      CALL RHOUH
      CALL PROPMU
 350  CONTINUE
      CALL MOTION
C
C
C*****************************************************************************
C    UPDATE AND PRINT
C*****************************************************************************
C
      IF(NX .LT. 21) GO TO 380
      CALL NEWDX
      PRI1 = .TRUE.
C
 380  CONTINUE
      CALL UPDATE
C
      IF((XP - XOB) .LT. XLBAR) GO TO 390
      STOP
C
 390  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 AREAS
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BOREDB,DT2BD,DTDSG,XLBAR
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
      DATA PIDF/.785398/
C
C    SUBROUTINE AREAS CALCULATES CROSS-SECTIONAL AREAS AND VOLUME
C    INCREMENTS ASSOCIATED WITH THE CHAMBER.
C
C
C   **********************************************************************
C   *    NOTE THAT FOR 105MM PROGRAM CHAM2 IS ALWAYS TRUE
C   *                                                                    *
C   **********************************************************************
C        AREAAX - AREA OF BELL IGNITER TUBE
C        AREAGP(I) - AREA OF GAP AT TOP OF CHAMBER AT AXIAL GRID I
C        AREAC(I) - AREA OF ENTIRE CHAMBER AT AXIAL GRID I WHEN CHAMBER
C             BECOMES ONE DIMENSIONAL.  SO THAT CALCULATIONS IN AXIS GO
C             SMOOTHLY, AREAC(I) IS INITIALLY SET TO 1.0 IF CHAM1 IS
C             TRUE AND AREAAX OTHERWISE.
C        AREAR(I) - AREA OF ENTIRE CHAMBER MINUS THE BELL IGNITER TUBE
C             AT AXIAL GRID I IF CHAM1 OR CHAM2 IS TRUE.  IF CHAM3 IS
C             TRUE, AREAGP(I) IS ALSO SUBTRACTED.
C        AREAG(J) - GENERALIZED AREA OF RADIAL GRID J, USED FOR
C             SETTING DUMMY BARREL GRID 1 PROPERTIES.  IF CHAM2 IS
C             TRUE AREAG(1) IS AREAAX AND AREAG(2) IS AREAR(NGX).
C             IF CHAM3 IS TRUE AREAG(1) AND AREAG(2) ARE THE SAME AS
C             WHEN CHAM2 IS TRUE AND AREAG(3) IS AREAGP(NGX).  IF CHAM1
C             IS TRUE, AREAG(J) IS SET USING DR.
C        AREACH - AREAG(1) + . . . + AREAG(NGR)
C    IF CHAM1 IS TRUE, AREAG IS USED FOR MAKING THE CHAMBER ONE
C    DIMENSIONAL AT EVERY I.  IF CHAM2 IS TRUE, AREAAX AND AREAR(I) ARE
C    USED AT GRID I.  IF CHAM3 IS TRUE, AREAGP(I) IS ALSO USED.
C    FOR DOTMIG AND DOTMBG TERMS VOLUME INCREMENTS ARE NEEDED.
C        DVAXIS - VOLUME INCREMENT OF GRIDS FOR J = 1.
C        DVAXIT - VOLUME INCREMENT OF GRIDS FOR J = 2.  DVAXIT WOULD
C             NEED TO BE AN ARRAY FOR CHAM2 OR CHAM3 TRUE BUT IT IS NOT
C             NEEDED THEN.
C
C
      AREAAX = PIDF*DIAMBT*DIAMBT
C
```

```
C     CALCULATE ARRAY AREAC AND FROM IT GET AREAR
      XB = 0.0
C
C
C *** THE CHAMBER DIAMETER DECREASES LINEARLY OVER DIS1 FROM DIAM1 TO
C *** DIAM2.
C
      SLOPE = (DIAM1 - DIAM2)/DIS1
      DO 270 I=1,NGX
         DIAM = DIAM1 - SLOPE*XB
         AREAC(I) = PIO4*DIAM*DIAM
         XB = XB + DX
         IF( XB .GT. DIS1) GO TO 280
  270 CONTINUE
C
  280 NGP1 = NGX + 1
C ***SET AREAR(NGP1) TO BOREA FOR USE IN AXIT2 AND PROPEL
      AREAR(NGP1) = BOREA
C
      DO 360 I = 1,NGX
          AREAR(I) = AREAC(I) - AREAAX
 360  CONTINUE
C
C *** AREAC(NGP1) SHOULD NOT BE NEEDED ANYWHERE.  SET IT TO A LARGE
C     NEGATIVE NUMBER TO CATCH ANY PLACES WHERE IT IS NEEDED.
      AREAC(NGP1) = -10.E+15
C
      DO 410 I = 1,NGX
          AREAC(I) = AREAAX
 410  CONTINUE
C
C *** AREAG(1), AREAG(2), AND AREACH SHOULD NOT BE NEEDED ANYWHERE.
      AREAG(1) = -10.E+15
      AREAG(2) = -10.E+15
      AREACH = -10.E+15
      DVAXIS = AREAAX*DX
C
C
 500  CONTINUE
C
      IF(.NOT. IDEBUG(6)) 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)
      RETURN
C
 2002 FORMAT(///,' ARRAY AREAR',//)
 2003 FORMAT(9X,10F11.7,/)
 2004 FORMAT(///,' ARRAY AREAC',//)
 2005 FORMAT(///,' ARRAY AREAG',//,20X,5F11.7)
      END
      SUBROUTINE AXIS
C     IX IS I HERE, IR IS J
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,
     $  UTDT,VBGAVE,VTDT
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RRO,CV0,CVH
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      DATA GRAV/32.16/
C
      RB = 0.0
      XB=XB+DX
C
      CALL GSPROP(R0,RRO,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
      BUGGER = (GAM - 1.0)/TWOGJ
      IP1 = I+1
      IM1 = I -1
C
C
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST
C     POROSITY OF THE PROPELLANT.
      F5 = PHIBG(I,J)
      A5 = AREAC(I)
      G2 = RHOBG(IM1,J)
      G4 = RHOBG(IP1,J)
      G5 = RHOBG(I,J)
      E2 = G2*UBG(IM1,J)
      E4 = G4*UBG(IP1,J)
      E5 = G5*UBG(I,J)
      H2 = PHIBG(IM1,J) * AREAC(IM1) * E2
      H4 = PHIBG(IP1,J) * AREAC(IP1) * E4
      P2=PCH(IM1,J)
      P4=PCH(IP1,J)
      P5=PCH(I,J)
      A2=AREAC(IM1)
      A4=AREAC(IP1)
      EI2=HBG(IM1,J)-PCH(IM1,J)/G2/778.0
      EI4=HBG(IP1,J)-PCH(IP1,J)/G4/778.0
      EI5=HBG(I,J)-P5/G5/778.0
      C2=G2*EI2
       C4=G4*EI4
      C5=G5*EI5
C
      PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25
      RHOAVE = (G2 + G5 + G5 + G4)*0.25
      UBGAVE = (UBG(IM1,J) + UBG(I,J) + UBG(I,J) + UBG(IP1,J))*0.25
      VBGAVE = 0.0
      UPBAVE = (UPB(IM1,J) + UPB(I,J) + UPB(I,J) + UPB(IP1,J))*0.25
C
      PHITDT = PHIBTD(I,J) + PHI2TD(I,J) + PHIPTD(I,J) - 2.0
      RHOTDT = ( F5*RHOAVE - T2DX*(H4 - H2)/A5 + DELT*DOTMIG(I)/DVAXIS
     $         + DOTMB(I,J) + DOTMP(I,J) )/PHITDT
      PHIRHO = PHITDT*RHOTDT
      TERM=0.0
      IF(DOTMIG(I).LT.0.0)TERM=1.0
C
      UTDT = ( F5*(E2 + E5 + E5 + E4)*0.25
     1        - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J)
     2  + GRAV * AREAC(I) * PHIBG(I,J) * (PCH(IP1,J)
     3   - PCH(IM1,J)))/A5
     *  +DELT*DOTMIG(I)*UBG(I,J)/DVAXIS*TERM
     4        + DOTMB(I,J)*UPB(I,J)  )/PHIRHO
      IF (  ABS(UTDT).LT. 0.1  )UTDT = 0.0
C
      VTDT = 0.0
C
      IF(PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)
C
      HIGN = HBG(I,1)
      IF ( DOTMIG(I) .GT. 0.00001 ) HIGN = HBG(I,2)
C
      ETDT=(F5*(C2+C5+C5+C4)/4.0
     $-T2DX*(H4*EI4+A4*P4/778.0*(PHIBG(IP1,J)*UBG(IP1,J)
     $+(1.0-PHIBG(IP1,J))*UPB(IP1,J))-H2*EI2-A2*P2/778.0*
     $(PHIBG(IM1,J)*UBG(IM1,J)+(1.0-PHIBG(IM1,J))*UPB(IM1,J) ) )/A5
     $+DELT*(DOTMIG(I)/DVAXIS*HIGN-QBAG(I,J))
     $-XDRAG(I,J)*UPB(I,J)*DELT/778.0
     $+DOTMB(I,J)*(HMB+UPB(I,J)**2/TWOGJ)
     $+DOTMP(I,J)*(HBP+UPB(I,J)**2/TWOGJ))/PHIRHO
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PN,       ETDT,TDUM,
     $RHOTDT,UTDT,0.0,GAM,CP,4)
      HBGTD(I,J)=ETDT+PN       /RHOTDT/778.0
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE AXIT2
C     SUBROUTINE AXIT2 IS CALLED FOR GRIDS IN THE 2ND RADIAL ROW WHEN
C     THE CHAMBER HAS 2 SEPARATE ONE DIMENSIONAL ROWS.
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,
     $  UTDT,VBGAVE,VTDT
      COMMON/GRIDNX/DXPRIM
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $    AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RRO,CV0,CVH
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1    VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2    DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3    PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4    VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      DATA GRAV/32.16/
C
C
C     SAVE DX, CHANGE TO DXPRIM WHEN I=NGX, NX=1.
      DXTEMP=DX
      IF(I.EQ.NGX.AND.NX.EQ.1)DX=DXPRIM
      T2DX=DELT/(2.0*DX)
C
      CALL GSPROP(R0,RRO,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
      BUGGER = (GAM - 1.0)/TWOGJ
      IP1 = I+1
      IM1 = I - 1
C
C
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST
C     POROSITY OF THE PROPELLANT.
      F5 = PHIBG(I,J)
      A5 = AREAR(I)
      G2 = RHOBG(IM1,J)
      G4 = RHOBG(IP1,J)
      G5 = RHOBG(I,J)
      E2 = G2*UBG(IM1,J)
      E4 = G4*UBG(IP1,J)
      E5 = G5*UBG(I,J)
      H2 = PHIBG(IM1,J)*AREAR(IM1)*E2
      H4 = PHIBG(IP1,J)*AREAR(IP1)*E4
      P2=PCH(IM1,J)
      P4=PCH(IP1,J)
      P5=PCH(I,J)
       A2=AREAR(IM1)
      A4=AREAR(IP1)
       A5=AREAR(I)
      EI2=HBG(IM1,J)-P2/G2/778.0
      EI4=HBG(IP1,J)-P4/G4/778.0
      EI5=HBG(I,J)-P5/G5/778.0
       C2=G2*EI2
      C4=G4*EI4
       C5=G5*EI5
      VOL = AREAR(I)*DX
C
      PHIAVE = (PHIBG(IM1,J) + F5 + F5 + PHIBG(IP1,J))*0.25
      RHOAVE = (G2 + G5 + G5 + G4)*0.25
      UBGAVE = (UBG(IM1,J) + UBG(I,J) + UBG(I,J) + UBG(IP1,J))*0.25
      VBGAVE = 0.0
      UPBAVE = (UPB(IM1,J) + UPB(I,J) + UPB(I,J) + UPB(IP1,J))*0.25
C
      PHITDT = PHIBTD(I,J) + PHI2TD(I,J) + PHIPTD(I,J) - 2.0
      RHOTDT = ( F5*RHOAVE - T2DX*(H4 - H2)/A5
     1         - DELT*(DOTMIG(1) - DOTMBG(I))/VOL
     2         + DOTMB(I,J) + DOTMP(I,J) )/PHITDT
      PHIRHO = PHITDT*RHOTDT
      TERM1=0.0
      IF(DOTMIG(1).GT.0.0)TERM1=1.0
      TERM2=0.0
      IF(DOTMBG(I).LT.0.0)TERM2=1.0
C
      UTDT = ( F5*(E2 + E5 + E5 + E4)*0.25
     1         - T2DX*(H4*UBG(IP1,J) - H2*UBG(IM1,J)
     2  + GRAV * AREAR(1) * PHIBG(I,J) *
     3  (PCH(1P1,J) - PCH(IM1,J)))/A5
     *  -DELT*(DOTMIG(I)*UBG(I,J)*TERM1-(DOTMBG(I)*UBG(I,J)*TERM2))/VOL
     4         + DOTMB(I,J)*UPB(I,J) )/PHIRHO
      IF (  ABS(UTDT).LT. 0.1  )UTDT = 0.0
C
      VTDT = 0.0
C
      IF(PHIAVE.LT.0.999)CALL DRAG(XDRAG(I,J),.FALSE.,I,J)
C
      HIGN = HBG(I,1)
      IF ( DOTMIG(1) .GT. 0.00001 ) HIGN = HBG(I,2)
      HBGN = HBG(I,2)
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)
C
      ETDT=(F5*(C2+C5+C5+C4)/4.0
     $-T2DX*(H4*EI4+A4*P4/778.0*(PHIBG(IP1,J)*UBG(IP1,J)
     $+(1.0-PHIBG(IP1,J))*UPB(IP1,J))-H2*EI2-A2*P2/778.0*
     $(PHIBG(IM1,J)*UBG(IM1,J)+(1.0-PHIBG(IM1,J))*UPB(IM1,J)))/A5
     $+DELT*((DOTMBG(I)*HBGN-DOTMIG(I)*HIGN)/VOL
     $-QBAG(I,J)-XDRAG(I,J)*UPB(I,J)/778.0)
     $+DOTMB(I,J)*(HMB+UPB(I,J)**2/TWOGJ)
     $+DOTMP(I,J)*(HBP+UPB(I,J)**2/TWOGJ))/PHIRHO
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PN,       ETDT,TDUM,
     $RHOTDT,UTDT,0.0,GAM,CP,4)
      HBGTD(I,J)=ETDT+PN      /RHOTDT/778.0
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
C     RESTORE DX AND T2DX
      DX=DXTEMP
      T2DX=DELT/(2.0*DX)
      RETURN
      END
      SUBROUTINE BARSET
      COMMON/DETNX/ NXSAVE
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ,XLBAR
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/GRIDNX/DXPRIM
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/MOCON/CON3,CON4,CON5,AREAPB,Z0,WOB,XOB,FDMAX,PINER,
     $  CF,RADPB,PMASS,XINT,PINT,XLO,PLO,CON6
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL PRI1,IDEBUG
      DATA GRAV/32.16/
C
      NAMELIST/BARINP/RADPB,PMASS,PINER,CF,XLBAR,
     $  PZ0,WOB,PDMAX,XINT,PINT,XLO,PLO
      READ(5,BARINP)
      IF(IDEBUG(7)) WRITE(6,BARINP)
      NX = NXSAVE
C
      RADPB = RADPB/12.
      XLBAR = XLBAR/12.
      WOB = WOB/12.
      DEPTH = DEPTH/12.
      XOB = FLOAT(NX - 1)*DX
      XINT = XINT/12.
      XLO = XLO/12.
      PZ0 = PZ0*144.
      PDMAX = PDMAX*144.
      PINT = PINT*144.
      PLO = PLO*144.
C
C
```

```
C***********************************************************************
C    CONSTANTS FOR BARREL SUBROUTINES
C***********************************************************************
C
C    CONSTANTS FOR SUBROUTINE MOTION
      AREAPB = 3.141593*RADPB*RADPB
      CON5 = 0.5*DELT*DELT
      CON3 = (PDMAX - PZ0)*AREAPB/WOB
      CON4 = (PDMAX - PINT)*AREAPB/(XINT - WOB)
      FDMAX = PDMAX*AREAPB
      Z0 = PZ0*AREAPB
      CON6 = ( (PINT - PLO)/(XLO - XINT) )*AREAPB
C
C    INITIALLY THE PROJECTILE IS NOT MOVING
      XP = XOB
      VP = 0.0
      DXPRIM = DX
C
C
C    CONSTANTS FOR SUBROUTINE BNDLYR
      BORED8 = BORED/8.0
      DTDSQ = DELT*BORED*BORED
      DT2BD = -0.5*DELT/BORED
C
C
C    CONSTANT FOR SUBROUTINE PROPMO
      GTRHOP = GRAV*DELT/RHOP
C
C    FINISH TOTALLING THE GAS MASS.  NOTE THAT PHI IS 1.0 NOW.
      IF(NX .EQ. 1) GO TO 30
      IF(NX .EQ. 2) GO TO 25
      TOTM = TOTM + (FLOAT(NX - 2) + 0.5)*RHO0*BOREA*DX
      GO TO 30
 25   TOTM = TOTM + 0.5*RHO0*BOREA*DX
 30   CONTINUE
C
      NAMELIST/BARCHK/RADPB,WOB,BORED,XOB,
     $  CON3,CON4,CON5,AREAPB,XP,VP,BOREA,GTRHOP,BORER,BORED8,DTDSQ,
     $  DT2BD,TOTM,NX,CON6,FDMAX,Z0,      XINT,XLO,PZ0,PDMAX,PINT,PLO
      IF(IDEBUG(8)) WRITE(6,BARCHK)
C
C
C***********************************************************************
C    INITIALIZE BARREL ARRAYS
C***********************************************************************
      CALL CLEAR(PHI(1),UAENER(100))
      CALL CLEAR(XLB(1), UDIB2(100))
C
      IF(NX .EQ. 1) GO TO 60
      DO 50 I=2,NX
      RHOG(I) = RHO0
      HG(I) = H0
      PG(I) = P0
      UG(I) = U0
      TG(I) = T0
      AMASS(I) = BOREA
      AMOM(I) = BOREA
      AENER(I) = BOREA
      UAMASS(I) = BOREA
      UAMOM(I) = BOREA
      UAENER(I) = BOREA
C
C     PMDOT AND UF HAVE ALREADY BEEN CLEARED
 50   CONTINUE
C
C     SET AREAS AT BARREL GRID 1 TO BORE AREA OF BARREL.
C     FOR THE 105MM PROGRAM, HOWEVER, SET AREAS AT BARREL GRID 1 TO
C     AREAR(NGX).
 60   CONTINUE
      AMASS(1)=BOREA
      UAMASS(1)=BOREA
      AMOM(1)=BOREA
      UAMOM(1)=BOREA
      AENER(1)=BOREA
      UAENER(1)=BOREA
C
C
C     SET ALL ENTRIES IN ARRAYS PHI AND UPHI TO 1.0 SO THAT WHEN GRIDS AR
C     THERE WILL BE NO PROPELLANT IN THEM.
C *** PHI(1) AND UPHI(1) WILL BE SET IN MAIN BEFORE ANY BARREL
C *** CALCULATIONS.
      DO 80 I = 2,100
         PHI(I) = 1.0
         UPHI(I) = 1.0
      PHI2(I)=1.0
       UPHI2(I)=1.0
   80 CONTINUE
      RETURN
      END
      SUBROUTINE BNDLYR
C
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ
      COMMON/DETNX/NXSAVE
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RRO,CV0,CVH
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,GCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      REAL MACHSQ
      LOGICAL REPEAT
      DATA REPEAT/.FALSE./
      DATA PIDF/.785398/
C     SUBROUTINE BNDLYR CALCULATES THE THICKNESSES OF THE BOUNDARY LAYERS
C     FORMED AS PROPELLANT GAS FLOWS THROUGH THE BARREL.
C              AMASS IS THE AREA AVAILABLE FOR MASS FLOW
C              AMOM IS THE AREA OUTSIDE THE MOMENTUM LAYER
C              AENER IS THE AREA OUTSIDE THE ENERGY BOUNDARY LAYER
C     IT ALSO CALCULATES QL, THE HEAT TRANSFERRED FROM THE GRID TO OR
C     FROM THE BARREL.
C     ARRAY QL CONTAINS 0.0'S AT THE BEGINNING OF EACH TIME INTERVAL.
C
C
      IF(REPEAT) GO TO 10
      REPEAT = .TRUE.
      NXSAV1 = NXSAVE + 1
C
 10   CONTINUE
      NM1 = NX - 1
      ITEMP=NM1
C
      SPDX = 0.0
      DO 60 I=NXSAV1,NM1
          U =(UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25
          IF(U .LT. 10.) U = 10.
          USQ = U*U
C
C     STATIC ENTHALPY
          HSTAT = (HG(I)+HG(I)+HG(I-1)+HG(I+1))*0.25- .2E-4*USQ
          CALL GSPROP(R0,RRO,R,CV0,CVH,CV,PG(I),HSTAT,TSTAT,RHODUM,
     $       0.0,0.0,GAM,CP,3)
          GAM1 = GAM - 1.0
          GAM3 = (3.0*GAM - 1.0)/(2.0*GAM1)
C
C     VISCOSITY
          VIS = C1*TSTAT*SQRT(TSTAT)/(TSTAT + C2)
```

```
C
C    REYNOLD'S NUMBER/FOOT
          RE = (RHOG(I)+RHOG(I)+RHOG(I-1)+RHOG(I+1))*0.25*U/VIS
C
C    MACH NUMBER SQUARED
          MACHSQ = USQ/(32.16*GAM*R*TSTAT)
C
C    MACH NUMBER WEIGHTING FUNCTION
          PRESX = MACHSQ*MACHSQ/((1.0 + 0.5*GAM1*MACHSQ)**GAM3)
          PDX = PRESX*DX
          SPDX = SPDX + PDX
C
C    EQUIVALENT FLAT PLATE LENGTH
          EXF = SPDX/PRESX
C
C    REYNOLD'S NUMBER
          REX = RE*EXF
          GFUN1 = (1.0 + 0.25*GAM1*MACHSQ)**(-0.7)
          GFUN2 = (1.0 + 2.0*GAM1*MACHSQ)**0.44
          IF( REX .LT. 5.0E6  ) GO TO 30
          REF = EXF/REX**0.1667
          DELTA = 0.23*REF
C
          IF(DELTA .LT. BORER) GO TO 20
          ITEMP = I
          GO TO 150
C
C    DISPLACEMENT BOUNDARY LAYER THICKNESS
 20       CONTINUE
          DELSTR = 0.028*GFUN2*REF
C
C    MOMENTUM BOUNDARY LAYER THICKNESS
          THETA = 0.022*GFUN1*REF
          GO TO 40
C
 30       REF = EXF/REX**0.2
          DELTA = 0.37*REF
C
          IF(DELTA .LT. BORER) GO TO 35
          ITEMP = I
          GO TO 150
C
 35       CONTINUE
          DELSTR = 0.046*GFUN2*REF
         THETA = 0.036*GFUN1*REF
C
C       ENERGY BOUNDARY LAYER THICKNESS
 40       DELDST = 1.269*DELSTR/(DELSTR/THETA - 0.379)
C
C    MASS FLOW AREA
          UAMASS(I) =   PIDF*(BORED - 2.0*DELSTR)**2
```

```
C
C     MOMENTUM FLOW AREA
           UAMOM(I) =    PIDF*(BORED - 2.0*THETA)**2
C
C     ENERGY FLOW AREA
           UAENER(I) =    PIDF*(BORED - 2.0*DELDST)**2
C
C     AVERAGE (REFERENCE) ENTHALPY IN BOUNDARY LAYER
           HSTAR = 0.5*(HG(I) + HW) - 6.128E-6*USQ
           CALL GSPROP(RO,RRO,R,CV0,CVH,CV,PG(I),HSTAR,TSTAR,RHOSTR,
      * 0.0,0.0,GAM,CP,3)
       CALL GSPROP(RO,RRO,R,CV0,CVH,CV,PDUM,HG(I),
      *  TDUM,RHOG(I),0,0.0,GAM,CP,2)
           GAM1 = GAM - 1.0
           MACHSQ = USQ/(32.16*GAM*R*TSTAR)
           VISTR = C1*TSTAR*SQRT(TSTAR)/(TSTAR + C2)
           RESTRX = RHOSTR*U/VISTR*EXF
C
C     COMPRESSIBLE SKIN FRICTION COEFFICIENT
           CFR = (1.0 + GAM1*GAM1*MACHSQ)**(-0.6)
           QDOT = 0.0366*VISTR/EXF * RESTRX**0.8 * (HG(I) - HW)*CFR
C
C     HEAT FLUX
           QL(I) = 3.141593*BORED*DX*QDOT
C
C
 60    CONTINUE
       GO TO 200
C
C
C
C
 100   CONTINUE
       DO 110 I = NXSAV1,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
      U=(UG(I)+UG(I)+UG(I-1)+UG(I+1))*0.25
      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)
     $        *(1.0 - EXP(LAMU*DT2BD))*RHOU*DTDSQ
C
C
      I2 = I+1
C
      DO 160 I=NXSAV1,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(R0,RR0,R,CV0,CVH,CV,PG(I),HSTAT,TSTAT,RHODUM,
     $                0.0,0.0,GAM,CP,3)
          VIS = C1*TSTAT*SQRT(TSTAT)/(TSTAT + C2)
          RHOU = (RHOG(1) + RHOG(I) + RHOG(I-1) + RHOG(I+1))*0.25*U
          RED = RHOU/VIS*BORED
          EPSLON = 0.0005*ALOG(RED) - 0.00556
          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(1) = PIDF*((HG(I) + HG(I) + HG(I-1) + HG(I+1))*0.25 -HW)
     $        *(1.0 - EXP(LAMU*DT2BD))*RHOU*DTDSQ
 160  CONTINUE
C
C
 200  CONTINUE
      DO 220 I = NXSAV1,NM1
      IF(1.GE.ITEMP) GO TO 220
      IP1=I+1
      IF(I.EQ.NM1) IP1=I
      UAMASS(I)=(2.0*AMASS(I)+2.0*AMASS(I-1)+2.0*AMASS(IP1)
     *  +UAMASS(I-1)+UAMASS(IP1)+UAMASS(I))/9.0
      UAMOM(I)=(2.0*AMOM(I)+2.0*AMOM(I-1)+2.0*AMOM(IP1)
     *  +UAMOM(I-1)+UAMOM(IP1)+UAMOM(I))/9.0
      UAENER(I)=(2.0*AENER(I)+2.0*AENER(I-1)+2.0*AENER(IP1)
     *  + UAENER(I-1)+UAENER(I)+UAENER(IP1))/9.0
  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 BPFIR
      COMMON/ CHARGE/ CHTC, IBEGC
      COMMON/BPT/ TEP(2)
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
C
C     ARRAY QBAG IS CLEARED IN UPDATE EACH TIME INTERVAL.
C     BPIGN WILL REMAIN TRUE ONLY IF AT EACH GRID THERE IS NO BLACK
C     POWDER OR THE BLACK POWDER IS IGNITED.
      BPIGN = .TRUE.
      DO 100 J=1,NGR
      DO 100 I=1,NGX
      IF(PHIBP(I,J).GE.0.99999) GO TO 100
          IF(TBP(I,J) .GE. TIGNBP) GO TO 100
          BPIGN = .FALSE.
C
C     GAS TEMPERATURE
          TTX = TBG(I,J)
          TEMP = TTX*SQRT(TTX)
C
C     GAS VISCOSITY
          TXMU = C1*TEMP/(TTX + C2)
C
C     THERMAL CONDUCTIVITY OF GAS
          TXK = C3*TEMP/(TTX + C4)
C
C     REYNOLDS NUMBER
          RETXBP = RHOBG(I,J)*ABS(UBG(I,J))*BPRAD(I,J)/TXMU
C
C     NUSSELT NUMBER
      IF(RETXBP.LE.1.0E-10)TXNUBP=0.000615*PCH(I,J)**0.556/TXK*
     1 BPRAD(I,J)
      IF(RETXBP.GT.1.0E-10)TXNUBP=0.3*RETXBP**0.62+
     1 0.000613*PCH(I,J)**0.556/TXK*BPRAD(I,J)
        TXNUBP=TXNUBP*0.5
C
C    BPRAD(I,J) SHOULD NOT BE LESS THAN 0.001, BECAUSE OTHERWISE
C    PHIBP(1,J) WOULD HAVE BEEN SET TO 1.0 IN PRIMER.
          ADVBP = 3.0*(1.0 - PHIBP(I,J))/BPRAD(I,J)
C
C    HEAT FLUX TO BLACK POWDER
          QCONBP = TXNUBP*TXK/BPRAD(I,J)*(TTX - TBP(I,J))
      IF(I.NE.IBEGC.OR.J.EQ.NGR) GO TO 50
      STDOFF=BELBEG-TEP(1)
      SPDJ=0.0
      THKJ2=0.5*(DIAM1-DIAMBT)-TOPGAP
      IF(J.EQ.2)SPDJ=0.5*(THKJ2+DIAMBT)
      DISPR=SPDJ+STDOFF
      IF(TIME.GE.0.0004) QPMR=3260.0*EXP(-30.5*DISPR-600.0*TIME)
      IF(TIME.LT.0.0004) QPMR=1.28E5*SQRT(TIME)*EXP(-30.5*DISPR)
      IF(QPMR.GT.QCONBP) QCONBP=QPMR
   50 CONTINUE
          IF(QCONBP .LT. 0.001) GO TO 100
          QBAG(I,J) = QCONBP*ADVBP
          TEMP = QBCONS*QCONBP
          TEFFBP = ((TBP(I,J) - T0)/TEMP)**2
          IF(TEFFBP .GT. TIME) TEFFBP = TIME
          TBP(I,J) = TBP(I,J) +
     $        TEMP*(SQRT(TEFFBP + DELT) - SQRT(TEFFBP))
          IF(TBP(I,J) .LT. TIGNBP) GO TO 100
          IF(IDEBUG(9)) WRITE(6,2000) TIME,I,J
 100  CONTINUE
C
      RETURN
C
 2000 FORMAT(/,' TIME =',E14.8,' BLACK POWDER AT GRID',I4,I4,
     $  ' IS IGNITED')
      END
      SUBROUTINE BPINIT
      COMMON/ CHARGE/ CHTC, IBEGC
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $    AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),      UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
C   SUBROUTINE BPINIT CALCULATES INITIAL POROSITY
C   VALUES AT EACH GRID THAT CONTAINS BLACK
C   POWDER.  THE SUBROUTINE READS IN THE FOLLOWING PARAMETERS%
C ***                BEGTC = LOCATION OF CHARGE (INCHES)
C ***                XLTC = LENGTH OF CHARGE (INCHES)
C ***                CHTC = CHARGE WEIGHT (POUNDS)
C
      COMMON/P/ IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/CFRACT/ FRACT1,FRACT2
      LOGICAL IDEBUG
C
      DIMENSION FRACTP(60)
      NAMELIST/BPCHK/ BEGTC,ENDTC,BELBEG,BELEND,IBEGB,IENDB,DX,IBEGTC,
     *                 IENDTC
C
C
C   INITIALIZE FRACT ARRAY
C
      CALL CLEAR(FRACTP(1),FRACTP(60))
C
      NAMELIST/BPINP/BEGTC,XLTC,CHTC
C
C   READ INPUTS AND CALCULATE CONSTANTS
C
      READ (5,BPINP)
      IF (IDEBUG(2)) WRITE(6,BPINP)
C
      BEGTC = BEGTC/12.0
      XLTC = XLTC/12.0
      CHTC = CHTC/16.0
C
C *** COMPUTE THE LOCATION OF THE TUBE CHARGE AND THE FRACTIONS FOR THE
C *** GRIDS IT OCCUPIES.
C
```

```
C *** DETERMINE THE GRIDS IN WHICH THE CHARGE BEGINS AND ENDS.
C
      ENDTC = BEGTC + XLTC
      IF(BEGTC .GE. .5*DX) GO TO 10
      IBEGTC = 1
      GO TO 20
 10   IBEGTC = BEGTC/DX + .5
      IBEGTC = IBEGTC + 1
 20   IENDTC = ENDTC/DX + .5
      IENDTC = IENDTC + 1
      IBEGC = IBEGTC
C
C *** CALCULATE FRACTIONS OF EACH GRID OCCUPIED BY THE CHARGE AND STORE
C *** THEM IN FRACTP
C
      IF(IBEGTC .EQ. 1) FRACTP(IBEGTC) = ((FLOAT(IBEGTC) - .5)*DX -
     *                                    BEGTC)/(DX*.5)
      IF(IBEGTC .EQ. 1) GO TO 30
      FRACTP(IBEGTC) = ((FLOAT(IBEGTC) - .5)*DX - BEGTC)/DX
  30  FRACTP(IENDTC) = (ENDTC - (FLOAT(IENDTC) - 1.5)*DX)/DX
      KS = IBEGTC + 1
      KE = IENDTC - 1
      DO 40 I = KS,KE
         FRACTP(I) = 1.0
  40  CONTINUE
C
C *** CALCULATE POROSITIES
C
      VOLC = (DIAMBT/2.0)*(DIAMBT/2.0)*XLTC*3.141593
      DO 50 J = IBEGTC,IENDTC
         PHIBP(J,1) = 1.0 - (CHTC*FRACTP(J))/VOLC/BPDENS
  50  CONTINUE
      IF(IDEBUG(30)) WRITE(6,BPCHK)
      RETURN
      END
      SUBROUTINE BSURA2
C
C     AREA FACTORS ARE NOT NEEDED IN THIS SUBROUTINE BECAUSE THEY WOLLD
C     ALL CANCEL.
C
      COMMON/PRMFLO/DOTMPM, UPRM
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,
     $  UTDT,VBGAVE,VTDT
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/RO,RRO,CVO,CVH
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
      NAMELIST/BSURCK/ BUGGER,GAM,IP1,F5,RHOAVE,G4,G5,E4,F4,RHOTDT,PHITD
     .T,CHBGTD,C4,C5,HIGN,I,J,QQ
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
      BUGGER = (GAM - 1.0)/TWOGJ
      IP1 = I+1
C
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST
C     POROSITY OF THE PROPELLANT.
      F4 = PHIBG(IP1,J)
      F5 = PHIBG(I,J)
      P4=PCH(IP1,J)
      P5=PCH(I,J)
      G4=RHOBG(IP1,J)
      G5=RHOBG(I,J)
      E4=G4*UBG(IP1,J)
      EI4=HBG(IP1,J)-P4/G4/778.0
      EI5=HBG(I,J)-P5/G5/778.0
      C4=G4*EI4
      C5=G5*EI5
      H4=F4*E4
      DENOM=2.0
      PHIAVE=(F4+F5)/DENOM
      RHOAVE = (G4 + G5)/DENOM
      UBGAVE = 0.0
      UPBAVE = (UPB(IP1,J) + UPB(I,J))*0.5
```

```
C
      PHITDT = PHIBTD(I,J) + PHI2TD(I,J) + PHIPTD(I,J) - 2.0
      RHOTDT=(F5*RHOAVE-DTDX*H4+DELT*(DOTMIG(I)+
     $DOTMPM)/DVAXIS*2.0+DOTMB(I,J)+DOTMP(I,J))/PHITDT
      QQ = QBAG(I,J)
      IF(RHOTDT .LT. 0.0) WRITE(6,BSURCK)
      PHIRHO = PHITDT*RHOTDT
C
      UTDT = 0.0
      VTDT = 0.0
C
      HIGN = HBG(I,1)
C
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(1,2)
      ETDT=(F5*(C4+C5)/DENOM-DTDX*(H4*EI4
     $+P4*(F4*UBG(IP1,J)+(1.0-F4)*UPB(IP1,J))/778.0)
     $+DELT*((DOTMIG(I)*HIGN+DOTMPM*HBP)*2.0/DVAXIS
     $-GBAG(I,J))+DOTMB(I,J)*HMB+DOTMP(1,J)*HBP)/PHIRHO
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PN,      ETDT,TDUM,
     $RHOTDT,UTDT,0.0,GAM,CP,4)
      HBGTD(I,J)=ETDT+PN       /RHOTDT/778.0
      CHBGTD = HBGTD(I,J)
      IF (HBGTD(I,J) .LT. 0.0) WRITE(6,BSURCK)
C
      UTDT=2.0* DELT * DOTMPM * UPRM/RHOTDT/DVAXIS
      RHOBTD(1,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE BSURT2
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE.
     $  UTDT,VBGAVE,VTDT
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/RO,RRO,CVO,CVH
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPR2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2([illegible],5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PCH(I,J),HBG(I,J),TDUM,
     $      RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
      BUGGER = (GAM - 1.0)/TWOGJ
      IP1 = I+1
C
C    IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST
C    POROSITY OF THE PROPELLANT.
      P4=PCH(IP1,J)
      P5=PCH(I,J)
      A4=AREAR(IP1)
      A5=AREAR(I)
      F4=PHIBG(IP1,J)
      F5=PHIBG(I,J)
       G4=RHOBG(IP1,J)
      G5=RHOBG(I,J)
      E4=G4*UBG(IP1,J)
      EI4=HBG(IP1,J)-P4/G4/778.0
      EI5=HBG(I,J)-P5/G5/778.0
      C4=G4*EI4
      C5=G5*EI5
       H4=F4*E4*A4
      DENOM=2.0
      PHIAVE=(F4+F5)/DENOM
      RHOAVE = (G4 + G5)/DENOM
      UBGAVE = 0.0
      UPBAVE = (UPB(IP1,J) + UPB(I,J))*0.5
C
      PHITDT=PHIBTD(I,J)+PHI2TD(I,J)+PHIPTD(I,J)-2.0
C
      RHOTDT=(F5*RHOAVE-DTDX*H4/A5-DTDX*
     $(DOTMIG(I)-DOTMBG(I))/A5+DOTMB(I,J)+DOTMP(I,J))/PHITDT
```

```
C
C
      PHIRHO = PHITDT*RHOTDT
C
      UTDT = 0.0
      VTDT = 0.0
C
      HIGN = HBG(I,1)
      IF(DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)
      HBGN = HBG(I,2)
      IF(DOTMBG(I) .GT. 0.00001) HBGN = HBG(I,3)
C
      ETDT=(F5*(C4+C5)/DENOM-DTDX*(H4*E14
     $+A4*P4/778.0*(F4*UBG(IP1,J)+(1.0-F4)*UPB(IP1,J))
     $+DOTMIG(I)*HIGN-DOTMBG(I)*HBGN)/A5
     $+DOTMB(I,J)*HMB+DOTMP(I,J)*HBP
     $-DELT*QBAG(I,J))/PHIRHO
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PN,        ETDT,TDUM,
     $RHOTDT,UTDT,0.0,GAM,CP,4)
      HBGTD(I,J)=ETDT+PN        /RHOTDT/778.0
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE CHSET
C
      DIMENSION CHWT2(7)
      COMMON/ CHARGE/ CHTC, IBEGC
      COMMON/NEWPHI/PHI02,IENDC2
      COMMON/DETNX/ NXSAVE
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),
     1THICK(60)
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSG
      COMMON/BURN/ATPB,CT,PEXP
      COMMON/CAREAS/ARROW1,ARROW2,ARROW3,ARTOT
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/DRGCON/VTSG
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/FORCE/ PFORCE(60,5), PFORDT(60,5)
      COMMON/GASCON/RO,RRO,CVO,CVH
      COMMON/GASES/ROM6,RROM6,CVOM6,CVHM6,ROBP,RROBP,CVOBP,CVHBP
      COMMON/CFRACT/ FRACT1,FRACT2
      COMMON/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMP,WMBP,
     $  GAMIB,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLE
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $  AH(60),FRACT(60)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  CCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
      LOGICAL FAIL
C
      DATA GRAV,XJUL,PI,PIDF/32.16,778.,3.141593,.785398/
      NAMELIST/CHINP/
     1A0BP,A1BP,A2BP,A3BP,A0MP,A1MP,A2MP,A3MP,A0SP,A1SP,A2SP,A3SP,
     2AGEN,AGEN2,AGENBP,ALPHA,ALPHBP,B0,B1,B2,B3,B4,BETA,BGEN,
     3BGEN2,BGENBP,BCRED,BPDENS,BPRAD0,C1,C2,C3,C4,CA,CGEN,CGEN2,
     4CHAM2,CHAM3,CHWT2,DI0,DI02,DIAM1,DIAM2,DIAMBT,DIFFPR,DIR1,DIR2,
     5DIS1,DO0,DO02,DOR1,DOR2,EXPEF,GAMBP,GAMMP,GAMSP,
     6HBP,HMAX,HMB,HMB2,NGR,NGX,NHOLES,NPERF,NPERF2,NROWH,
     7P0,PEXP,PEXP2,RADHOL,RF1,RF2,RHOP,RHOP2,RHOP1R,RHOP2R,RQ1,RQ2,
     8T0,TF,TIGN,TIGNBP,TW,U0,WMBP,WMMP,WMSP,XCL,XK,XKBP,XL0,XL02,
     9XLBEL,XLR1,XLR2
C
C
C
C***********************************************************************
C    READ INPUTS AND CALCULATE CONSTANTS FOR SUBROUTINES
C***********************************************************************
      READ(5,CHINP)
      IF(IDEBUG(2)) WRITE(6,CHINP)
      DO 2222 I = 1,60
      MFAIL(I) = 0
 2222 CONTINUE
      F0=1728.0
      F1=F0*62.4
      F2=F1*62.4
      F3=F2*62.4
      A0SP=A0SP/F0
      A1SP=A1SP/F1
      A2SP=A2SP/F2
      A3SP=A3SP/F3
      A0MP=A0MP/F0
      A1MP=A1MP/F1
      A2MP=A2MP/F2
      A3MP=A3MP/F3
      A0BP=A0BP/F0
      A1BP=A1BP/F1
      A2BP=A2BP/F2
      A3BP=A3BP/F3
      A0=A0BP
      A1=A1BP
      A2=A2BP
      A3=A3BP
       CUMSP=0.0
      CUMMP=0.0
      CUMBP=0.0
      WMCLE=WMBP
       GAMIB=GAMBP
      ROM6=ROM1
      RROM6=RROM1
      CVOM6=CVOM1
      CVHM6=CVHM1
      CHAM1=.FALSE.
C
      FN = FLOAT(NPERF)
      FN2 = FLOAT(NPERF2)
      XL0 = XL0/12.
            XL02 = XL02/12.
            DI02 = DI02/12.
            DO02 = DO02/12.
      DO0 = DO0/12.
      DI0 = DI0/12.
      DIAM1 = DIAM1/12.
      DIAM2 = DIAM2/12.
      DIS1 = DIS1/12.
      DIAMBT = DIAMBT/12.
      XLBEL = XLBEL/12.
      P0 = P0*144.
      DIFFPR = DIFFPR*144.0
      AGEN = AGEN/(12.*144.**PEXP)
      BGEN = BGEN/(12.*144.**PEXP )
      CGEN = CGEN/12.
      AGEN2=AGEN2/(12.0*144.0**PEXP2)
      BGEN2=BGEN2/(12.0*144.0**PEXP2)
      CGEN2=CGEN2/12.0
      DIR1=DIR1/12.0
      DIR2=DIR2/12.0
      DOR1=DOR1/12.0
      DOR2=DOR2/12.0
      XLR1=XLR1/12.0
      XLR2=XLR2/12.0
      BPRAD0 = BPRAD0/12.
      BPDENS = BPDENS/454.*16.38*1728.
      AGENBP = AGENBP/(12.0*144.0**EXPBP)
      BGENBP = BGENBP/12.0
C
C     INITIALLY THE GAS CONSTANTS ARE THOSE OF THE BLACK POWDER.
C     AFTER EACH TIME INTERVAL, IN UPDATE, THE GAS CONSTANTS WILL BE
C     CALCULATED ON THE BASIS OF THE GASES PRESENT.
      R0 = R0BP
      RR0 = RR0BP
      CV0 = CV0BP
      CVH = CVHBP
C
```

```
C     BORED WAS TAKEN OUT OF BARINP BECAUSE IT IS NEEDED HERE FOR AREA
C     CALCULATIONS.
       BORED = BORED/12.
       BORER = 0.5*BORED
       BOREA = PI*BORER*BORER
C
C     DETERMINE DX, IBEGB, IBEGC, IENDB
C *** IBEGC WILL BE GOTTEN FROM BPINIT.
       DX = DIST/FLOAT(NGX - 1)
       BELBEG = 0.0
       IBEGB = 1
       BELEND = XLBEL
       IENDB = XLBEL/DX + 1.5
       NXSAVE = NGX - IENDB + 1
       NGX = IENDB
C
C     FRACT1 AND FRACT2 ARE THE FRACTIONS OF GRIDS IBEGB  AND IENDB
C     RESPECTIVELY THAT THE BELL TUBE OCCUPIES.
       IF(IBEGB.EQ.1)FRACT1=1.0-2.0*BELBEG/DX
       IF(IBEGB.GT.1)FRACT1=((FLOAT(IBEGB)-0.5)*DX-BELBEG)/DX
       FRACT2 = (BELEND - (FLOAT(IENDB) - 1.5)*DX)/DX
C
C     DM,GCONS, AND GBCONS ARE NEEDED IN SUBROUTINE PRPFIR.
       DM = (1.5*DO0*DO0*XL0)**0.333
       DM2 = (1.5*DO02*DO02*XL02)**.333
       GCONS = 2.0*SQRT(ALPHA/3.141593)/XK
       GBCONS = 2.0*SQRT(ALPHBP/3.141593)/XKBF
C
C     ATPB AND CT ARE NEEDED FOR CALCULATING THE BURN RATE IN REGRES
       TEMPD=DOR1*DOR1-FN*DIR1*DIR1
       TEMPR1=(0.5*TEMPD+XLR1*(DOR1+FN*DIR1))/(XLR1*TEMPD)*RHOP1R
       TEMPD=DO0*DO0-FN*DI0*DI0
       TEMP1=(0.5*TEMPD+XL0*(DO0+FN*DI0))/(XL0*TEMPD)*RHOP
       RFACT1=RQ1/RF1*TEMPR1/TEMP1
       ATPB=(AGEN*T0+BGEN)*RFACT1
       CT=CGEN*T0*RFACT1
       TEMPD=DOR2*DOR2-FN2*DIR2*DIR2
       TEMPR2=(0.5*TEMPD+XLR2*(DOR2+FN2*DIR2))/(XLR2*TEMPD)*RHOP2R
       TEMPD=DO02*DO02-FN2*DI02*DI02
       TEMP2=(0.5*TEMPD+XL02*(DO02+FN2*DI02))/(XL02*TEMPD)*RHOP2
       RFACT2=RQ2/RF2*TEMPD/TEMP2
       RFACT2=RQ2/RF2*TEMPR2/TEMP2
       ATPB2=(AGEN2*T0+BGEN2)*RFACT2
       HMB1=HMB*RF1
       HMB2=HMB2*RF2
C
C      VTSG IS A CONSTANT USED IN SUBROUTINE DRAG
       VTSG = DM*GRAV
C
C     CALCULATE HW FROM TW AND P0
       CALL GSPROP(R0,RR0,R,CV0,CVR,CV,P0,HW,TW,RHODUM,0.0,0.0,GAM,CF,1)
```

AD-A051 827   CALSPAN CORP BUFFALO N Y   F/G 19/1

QUALITY CONTROL OF CONTINUOUSLY PRODUCED GUN PROPELLANT.(U)

AUG 77 E B FISHER

UNCLASSIFIED   CALSPAN-SA-5913-X-1   NL

2 OF 2
AD
A051827

END
DATE
FILMED
4 78
DDC

```
C
C
C***********************************************************************
C    INITIALIZE ARRAY IPATH.
C***********************************************************************
C
      CALL PATHS
C
C
C***********************************************************************
C    DETERMINE DELT AND CALCULATE CONSTANTS FOR FINITE DIFFERENCE EQNS.
C***********************************************************************
C
C    START TIME AT 0.0 AND DETERMINE DELT, THE TIME INTERVAL LENGTH
      TIME = 0.0
      GJ = GRAV*XJUL
      TWOGJ = 2.0*GJ
      DELT = BETA*DX/SQRT(TWOGJ*HMAX)
      TWODT = 2.0*DELT
      DTDX = DELT/DX
      T2DX=0.5*DTDX
C
C
C
C***********************************************************************
C    CALCULATE CROSS-SECTIONAL AREAS ASSOCIATED WITH THE CHAMBER
C***********************************************************************
C
C
      CALL AREAS
C
C    USING THE AREAS JUST CALCULATED, CALCULATE THE INITIAL VOLUME
C    OF CHAMBER RADIAL ROWS 1, 2 AND 3.  ALSO CALCULATE THE TOTAL
C    CHAMBER VOLUME.
      ARROW1 = (FLOAT(NGX) - 0.5)*DX*AREAAX
C
      ARROW2 = AREAR(1)*DX*0.5
      DO 100 I = 2,NGX
      ARROW2 = ARROW2 + AREAR(I)*DX
 100  CONTINUE
C
      ARTOT = ARROW1 + ARROW2
      IF(CHAM2) GO TO 120
C
      ARROW3 = AREAGP(1)*DX*0.5
      DO 110 I = 2,NGX
      ARROW3 = ARROW3 + AREAGP(I)*DX
 110  CONTINUE
      ARTOT = ARTOT + ARROW3
C
 120  CONTINUE
C
C
```

```
C***********************************************************************
C    INITIALIZE CHAMBER MATRIX
C***********************************************************************
C
      CALL CLEAR(XL(1,1),DI(60,5))
      CALL CLEAR(XLTDT(1,1),DITDT(60,5))
      CALL CLEAR(PHIBG(1,1),PHIBG2(60,5))
      CALL CLEAR(BPRAD(1,1),BPRAD(60,5))
      CALL CLEAR(PFORCE(1,1),PFORD1(60,5))
      CALL CLEAR(XL2(1,1),DI2TDT(60,5))
C
C     UPB IS INITIALLY 0.0
C    INITIALLY SET ALL POROSITIES TO 1.0 AND THEN CHANGE THOSE THAT HAVE
C    BLACK POWDER OR PROPELLANT
      DO 510 J=1,NGR
      DO 510 I=1,NGX
          PHIBG(I,J) = 1.0
          PHIBG2(I,J) = 1.0
          PHI2TD(I,J) = 1.0
          PHIBTD(I,J) = 1.0
          PHIBP(I,J) = 1.0
          PHIPTD(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)=BPRADO
 1000 TBP(I,J)=T0
 1001 CONTINUE
C
      DO 511 I = 1,NGX
          TZR(I) = T0
  511 CONTINUE
C
C *** THIS STATEMENT CAUSES THE BLACK POWDER IN GRID (IBEGC,1) TO BE
C *** IGNITED.
      TBP(IBEGC,1)=TIGNBP
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.
C     DETERMINE THE POROSITY OF GRIDS CONTAINING PROPELLANT:
C         1.  CALCULATE BED DENSITIES.
C         2.  CALCULATE THE POROSITY OF A FULL GRID.
```

```
C           3.  FROM THE WEIGHTS OF THE CHARGES, DETERMINE THE CHARGE
C               VOLUMES.
C           4.  CALL SUBROUTINE DETPHI WHICH CALCULATES THE POSITIONS OF
C               THE CHARGES, THE FRACTIONS OF GRIDS FILLED WITH
C               PROPELLANT AND WHICH FILLS ARRAYS PHIBG AND PHIBG2.
C
C
C *** BED DENSITY IS DETERMINED FROM THE REGRESSION EQUATION ........B0,
C ***  B1,B2,B3,B4 HAVE BEEN DETERMINED EXPERIMENTALLY
C
      RHOB = B0 + B1*XL0 + B2*D00 + B3*DI0 + B4*FLOAT(NPERF)
      RHOB2= B0 + B1*XL02+ B2*D002+ B3*DI02+ B4*FLOAT(NPERF2)
      PHI02=1.0-RHOB2/RHOP2
      PHI0 = 1.0 - RHOB/RHOP
C
C
C     CHWT2(1) AND CHWT2(2) ARE SP PROPELLANT, CHWT2(3) THROUGH CHWT2(7)
C     ARE MP PROPELLANT.
C
      CH1 = CHWT2(1) + CHWT2(2)
      CH2 = CHWT2(3) + CHWT2(4) + CHWT2(5) + CHWT2(6) + CHWT2(7)
C
      IF(CH2.NE.0.0) GO TO 520
      ATPB=ATPB2
      CT=CT2
      RHOP=RHOP2
  520 CONTINUE
C
C
      VOLCH1 = CH1/RHOB2
      VOLCH2 = CH2/RHOB
      IENDC2 = 0
C
C
      IF(VOLCH1 .GT. 1.0E-7) CALL DETPHI(VOLCH1,VOLCH2)
C
C
C *** INITIALIZE GRAIN PROPERTIES
C
  910 DO 570 I=IBEGB,IENDC2
          XL(I,2) = XL0
          D0(I,2) = D00
          DI(I,2) = DI0
          TZC(I,2) = T0
          XLTDT(I,2) = XL0
          DOTDT(I,2) = D00
          DITDT(I,2) = DI0
          XL2(I,2) = XL02
          D02(I,2) = D002
          D12(I,2) = DI02
          TZC2(I,2) = T0
          XL2TDT(I,2) = XL02
          D02TDT(I,2) = D002
          D12TDT(I,2) = DI02
 570  CONTINUE
C
C    DETERMINE H0 AND RHO0 FROM P0 AND T0 AND GSPROP
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,P0,H0,T0,RHO0,0.0 ,0.0 ,GAM,CF,1)
C
      DO  600 J=1,NGR
      DO 600 I=1,NGX
          PCH(I,J) = P0
          HBG(I,J) = H0
          RHOBG(I,J) = RHO0
          UBG(I,J) = U0
          TBG(I,J) = T0
 600  CONTINUE
C
      IF(IDEBUG(3)) WRITE(6,2006)
      DO 630 J=1,NGR
          IF(IDEBUG(3)) WRITE(6,2007) J,(PHIBG(I,J),I=1,NGX)
 630  CONTINUE
C
      IF(IDEBUG(3)) WRITE(6,2008)
      DO 640 J = 1,NGR
          IF(IDEBUG(3)) WRITE(6,2007) J,(PHIBP(I,J),I=1,NGX)
 640  CONTINUE
C
      IF(IDEBUG(3)) WRITE(6,2009)
      DO 625 J=1,NGR
         IF(IDEBUG(3)) WRITE(6,2007) J,(PHIBG2(I,J),I=1,NGX)
  625 CONTINUE
C
C
C***************************************************************************
C    SET UP HOLES
C***************************************************************************
C
C    SET UP HOLES IN BELL TUBE AND ANY PSEUDO HOLES BETWEEN RADIAL ROWS
C    ONE AND TWO.
      CALL HOLSET
C
C    GET HOLE AREA AT EACH GRID BETWEEN RADIAL ROWS ONE AND TWO.
      CALL HOLES
C
C    GET HOLE AREA AT EACH GRID BETWEEN RADIAL ROWS TWO AND THREE IF
C    CHAM3 IS TRUE.  IT IS CALCULATED USING AN AVERAGE DIAMETER
C    OBTAINED FROM SUBROUTINE AREAS.
```

```
      IF(.NOT. CHAM3) GO TO 700
      AREAH2 = 0.05*(PI/BETA)*0.5*(DAVG - TOPGAP)*TOPGAP*
     $     SQRT(2.0/(GAM - 1.0))
C
C
 700  CONTINUE
C
C***********************************************************************
C     COMPUTE TOTAL MASS IN THE SYSTEM
C***********************************************************************
C
C
C *** SUM1 IS THE CALCULATED INITIAL MASS OF BLACK POWDER AND PROPELLANT
C *** SUMB IS THE CALCULATED INITIAL MASS OF BLACK POWDER
C *** SUMP IS THE CALCULATED INITIAL MASS OF PROPELLANT
C *** TOTM1 IS THE INPUT MASS OF PROPELLANT AND BLACK POWDER
C
C     LOGIC IS NOT WRITTEN FOR CHAM1 TRUE.
       IF(CHAM1) WRITE(6,2000)
       IF(CHAM1) GO TO 800
C
C     TO FIND THE TOTAL GAS MASS, SUM THE PRODUCT OF TOTAL POROSITY*
C     DENSITY OF GAS*VOLUME  AT EACH GRID.
C     SINCE GAS DENSITY AND GRID LENGTH ARE CONSTANT, MULTIPLY BY THEM
C     AFTER SUMMING.
C *** FIRST SUM ALONG THE AXIZ
      SUM = 0.5*(PHIBG(1,1) + PHIBP(1,1) + PHIBG2(1,1) - 2.0)
      SUM = SUM*AREAC(1)
      SUMP=((1.0-PHIBG(1,1))*RHOP+(1.0-PHIBG2(1,1))*RHOP2)*DVAXIS*0.5
      SUMB = (1.0 - PHIBP(1,1))*DVAXIS*0.5*BPDENS
      VOL = DVAXIS
      VOLBPD = VOL*BPDENS
      DO 720 I=2,NGX
C
      SUMP=SUMP+((1.0-PHIBG(I,1))*RHOP*(1.0-PHIBG2(I,1))*RHOP2)*DVAXIS
      SUMB = SUMB + (1.0 - PHIBP(I,1))*VOLBPD
      SUM = SUM + (PHIBG(I,1) + PHIBP(I,1) + PHIBG2(I,1) - 2.0)*AREAC(I)
 720  CONTINUE
      TOTM = SUM
C
      SUM = 0.5*(PHIBG(1,2) + PHIBP(1,2) + PHIBG2(1,2) - 2.0)*AREAR(1)
      VOL = AREAR(1)*DX*0.5
      SUMP=SUMP+((1.0-PHIBG(1,2))*RHOP+(1.0-PHIBG2(1,2))*RHOP2)*VOL
      SUMB = SUMB + (1.0 - PHIBP(1,2))*BPDENS*VOL
      DO 730 I=2,NGX
      VOL = AREAR(I)*DX
      SUMP=SUMP+((1.0-PHIBG(I,2))*RHOP+(1.0-PHIBG2(1,2))*RHOP2)*VOL
      SUMB = SUMB + (1.0 - PHIBP(I,2))*BPDENS*VOL
         SUM = SUM + (PHIBG(1,2) + PHIBP(I,2) + PHIBG2(I,2) - 2.0)
     )     *AREAR(I)
 730   CONTINUE
       TOTM = TOTM + SUM
C
       TOTM = TOTM*RHOO*DX
C
C     ADD IN MASS OF BLACK POWDER AND PROPELLANT.
C
       CHWT = 0.0
       DO 906 I=1,7
          CHWT = CHWT + CHWT2(I)
  906  CONTINUE
C
C *** CALCULATE PERCENT ERROR FOR INITIAL MASS OF BLACK POWDER AND PROPE
       SUM1 = SUMB + SUMP
C *** WRITE CALCULATED MASSES OF BLACK POWDER AND PROPELLANT
       IF(IDEBUG(31)) WRITE(6,2004) SUMB,SUMP
       IF(IDEBUG(31)) WRITE(6,2005) CHTC,CHWT
       TOTM1 = CHWT + CHTC
       XPERCT = ((TOTM1 - SUM1)/TOTM1)*100.
C *** WRITE TOTAL CALCULATED MASS OF PROPELLANT AND BLACK POWDER
       IF(IDEBUG(31)) WRITE(6,2001) SUM1
C *** WRITE TOTAL INPUT MASS OF BLACK POWDER AND PROPELLANT
       IF(IDEBUG(31)) WRITE(6,2002) TOTM1
C *** WRITE PERCENT ERROR
       IF(IDEBUG(31)) WRITE(6,2003) XPERCT
       TOTM = TOTM + CHTC + CHWT
       TOTMG = TOTM - TOTM1
C *** WRITE INITIAL GAS MASS
       IF(IDEBUG(31)) WRITE(6,2010) TOTMG
C
 800   CONTINUE
C
       NAMELIST/CHKIN/XLO,DOO,DIO,RADHOL,PO,BOEN,XCL,NROWH,
      $ NHOLES,AREAH,TIME,DELT,TWOGJ,DTDX,T2DX,PHIO,
      $ HO,RHOO,TO,DVAXIS,DVAXIT,NCONS,ATPB,C1,VTSG,HW,DIAM1,DIAM2,
      '  LIS1,ONEL,IGNIT,AGEN,CGEN,
      $  DIAMBT,CHAM2,BORED,BORER,BOREA,AREAAX,AREACH,
      ' BPRADO,BPDENS,AGENBP,RO,RHO,
      $  XLBEL,DX,IBEGB,IBEGC,IENDB,ARROW1,ARROW2,ARROW3,ARTOT,
      $  BELBEG,BELEND,FRACT1,FRACT2,DAVG,GAM,TOTM,DM,DM2,OBCONS,AH,
      $  FRACT, IPS1, IPS2, IENDC2, CH1, CH2, CHWT, VOLCH1, VOLCH2,
      $  PHIO2, RHOB, NGX, XLO2, DOO2, DIO2, IBEGC
       IF(IDEBUG(4)) WRITE(6,CHKIN)
C
C
C
       RETURN
C
 2000 FORMAT(//,' LOGIC FOR FINDING TOTM WHEN CHAM1 IS TRUE IS NOT WRITT
      $EN YET')
```

```
2001 FORMAT(///,* CALCULATED MASS OF BLACK POWDER AND PROPELLANT IS *,
    .F10.5)
2002 FORMAT(//,* INPUT MASS OF BLACK POWDER AND PROPELLANT IS *,F10.5)
2003 FORMAT(//,* PERCENT ERROR IN INITIAL MASS CALCULATION IS *,F10.5)
2004 FORMAT(//,* BLACK POWDER CALCULATED MASS IS *,F10.5,* PROPELLANT C
    .ALCULATED MASS IS *,F10.5)
2005 FORMAT(//,* BLACK POWDER INPUT MASS IS *,F10.5,* PROPELLANT INPUT
    .MASS IS *,F10.5)
2010 FORMAT(//,* THE INITIAL MASS OF GAS IN THE SYSTEM IS *,F10.5)
2006 FORMAT(///,' ARRAY PHIBG',/)
2007 FORMAT(/,' RADIAL ROW',I2,/,(20X,10F10.6))
2008 FORMAT(///,' ARRAY PHIBP',/)
2009 FORMAT(///,* ARRAY PHIBG2 *,/)
     END
```

```
          IDENT CLEAR
          ENTRY CLEAR
CLEAR     BSSZ 1B
          SB1 1
          SA2 A1
          SA2 X2
          SA3 A1+B1
          SB3 X3
          MX6 0
LOOP      SA6 A2
          SB2 A2
          SA2 A2+B1
          NE B2,B3,LOOP
          EQ CLEAR
          END
```

```
      SUBROUTINE DETPHI(VOLCH1,VOLCH2)
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,RPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/NEWPHI/PHI02,IENDC2
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      LOGICAL PRI1,IDEBUG
C
C
C     DETERMINE THE EXTENT OF SP PROPELLANT, CHRGL1, BY ADDING ON GRID
C     VOLUME AFTER GRID VOLUME UNTIL VOLCH1 IS EXCEEDED.  THEN DETERMINE
C     WHAT INCREMENT OF THE LAST GRID IS NEEDED.
C          PREVV CONTAINS THE VOLUME THROUGH THE PREVIOUS GRID.
C          TOTV CONTAINS THE VOLUME THROUGH THE PRESENT GRID.
C          DELX IS THE LENGTH OF THE LAST GRID NEEDED.
C          RESTX IS THE REMAINDER OF THE LAST GRID.
C          FRACT3 IS THE FRACTION OF THE LAST GRID WHICH IS FILLED WITH
C               SP PROPELLANT.
C          FRACT4 IS THE FRACTION OF THE LAST GRID WHICH IS FILLED WITH
C               MP PROPELLANT (IF MP PROPELLANT OCCURS).
      HALFDX = 0.5*DX
      TOTV = AREAR(1)*HALFDX
      IF(VOLCH1 .GT. TOTV) GO TO 20
      ISAVE = 1
      CHRGL1 = VOLCH1/AREAR(1)
      RESTX = HALFDX - CHRGL1
      FRACT3 = CHRGL1/HALFDX
      FRACT4 = RESTX/HALFDX
      GO TO 50
C
   20 CONTINUE
      DO 30 I=2,NGX
      PREVV = TOTV
      TOTV = TOTV + AREAR(I)*DX
      ISAVE = I
      IF(TOTV .GT. VOLCH1) GO TO 40
   30 CONTINUE
      WRITE(6,1000)
      STOP
```

```
C
 40   CONTINUE
      DELX = (VOLCH1 - PREVV)/AREAR(ISAVE)
      CHRGL1 = (ISAVE - 1.5)*DX + DELX
      RESTX = DX - DELX
      FRACT3 = DELX/DX
      FRACT4 = RESTX/DX
C
 50   CONTINUE
      ENDC1 = CHRGL1
      BEGC2 = ENDC1
      IF(VOLCH2 .GT. 1.0E-6) GO TO 55
      VOLCH2 = 0.0
      CHRGL2 = 0.0
      GO TO 100
C
C
C     DETERMINE THE EXTENT OF MP PROPELLANT BY FILLING THE REMAINDER
C     OF THE LAST GRID WITH MP PROPELLANT AND THEN ADDING GRID VOLUME
C     AFTER GRID VOLUME AS BEFORE.
C         FRACT5 IS THE FRACTION OF THE LAST GRID FILLED WITH MP
C            PROPELLANT.
 55   CONTINUE
      TOTV = AREAR(ISAVE)*RESTX
      IF(VOLCH2 .GT. TOTV) GO TO 60
      CHRGL2 = VOLCH2/AREAR(ISAVE)
      RESTX = RESTX - CHRGL2
      IF(ISAVE .EQ. 1) FRACT5 = CHRGL2/HALFDX
      IF(ISAVE .GT. 1) FRACT5 = CHRGL2/DX
      FRACT4 = FRACT5
      GO TO 100
C
 60   CONTINUE
      INEXT = ISAVE + 1
      IF(INEXT .LE. NGX) GO TO 70
      WRITE(6,1000)
      STOP
C
 70   CONTINUE
      DO 80 I=INEXT,NGX
      PREVV = TOTV
      TOTV = TOTV + AREAR(I)*DX
      ISAVE = I
      IF(TOTV .GT. VOLCH2) GO TO 90
C
 80   CONTINUE
      WRITE(6,1000)
      STOP
C
```

```
 90   CONTINUE
      DELX = (VOLCH2 - PREVV)/AREAR(ISAVE)
      CHRGL2 = (ISAVE - 1.5)*DX - CHRGL1 + DELX
      FRACT5 = DELX/DX
C
C     DETERMINE THE GRIDS WHERE THE CHARGES BEGIN AND END AND LOAD
C     POROSITIES INTO GRIDS CONTAINING PROPELLANT.
 100  CONTINUE
      ENDC2 = BEGC2 + CHRGL2
      IBEGC1 = 1
      IENDC1 = ENDC1/DX + 1.5
      IBEGC2 = IENDC1
      IENDC2 = ENDC2/DX + 1.5
C
      IF(IENDC1 .EQ. 1) GO TO 120
      IM1 = IENDC1 - 1
      DO 110 I = 1,IM1
      PHIBG2(I,2) = PHI02
 110  CONTINUE
C
 120  CONTINUE
      PHIBG2(IENDC1,2) = 1.0 - FRACT3*(1.0 - PHI02)
      IF(VOLCH2 .LT. 1.0E-6) GO TO 150
C
      PHIBG(IBEGC2,2) = 1.0 - FRACT4*(1.0 - PHI0)
      IF(IENDC2 .EQ. IBEGC2) GO TO 150
      IF(IENDC2 .EQ. IBEGC2 + 1) GO TO 140
      IP1 = IBEGC2 + 1
      IM1 = IENDC2 - 1
      DO 130 I = IP1,IM1
      PHIBG(I,2) = PHI0
 130  CONTINUE
C
 140  CONTINUE
      PHIBG(IENDC2,2) = 1.0 - FRACT5*(1.0 - PHI0)
C
 150  CONTINUE
      NAMELIST/PHIS/CHRGL1,CHRGL2,ENDC1,ENDC2,FRACT3,FRACT4,FRACT5,
     $  IBEGC1,IBEGC2,IENDC1,IENDC2,PHI0,PHI02
      IF(IDEBUG(32)) WRITE(6,PHIS)
C
      RETURN
C
 1000 FORMAT(//,* NOT ENOUGH ROOM FOR THE PROPELLANT*)
C
      END
      SUBROUTINE PATHS
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,THEGP,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELEFG,IPS1,IPS2,RADPS,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 CORRESPOND TO THE NEW ROUTINES
C     AS FOLLOWS--
C          2 - AXIT2,AXIT3
C          9 - BSURA2
C          10 - BSURT2,BSURT3
C
C
C
C     THE FOLLOWING LOGIC IS USED FOR BOTH CHAM2 AND CHAM3 TRUE
 80   CONTINUE
      IPATH(1,1) = 3
      IPATH(1,2) = 5
      IPATH(IENDB,1) = 4
      IPATH(IENDB,2) = 2
      DO 85 I=2,NGX
      IF (I .EQ. IENDB) GO TO85
          IPATH(I,1) = 1
          IPATH(I,2) = 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,30(I4))
      END
      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/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMP,WMBP,
     $  GAMIB,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLE
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/SPLINT/WHOLEC,WHOLEB
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL WHOLEC,WHOLEB
      DATA PIDF/.785398/
C
C *** SUBROUTINE DIMIN CALCULATES DUAL GRANULATION PROPELLANT COMBUSTION
C *** DOTMH AND DOTMM ARE COMPUTED TO DETERMINE THE AVERAGE ENERGY OF
      DOTMH = 0.0
C *** THE BURNED PROPELLANT.
      DOTMM = 0.0
      DO 100 I = 2,NX
      IF(PHI(I).GE.0.99999.AND. PHI2(I) .GE. 0.99999)GO TO 90
C *** MULTIPERF PROPELLANT COMBUSTION CALCULATIONS
      IF(PHI(I) .GE. 0.99999) GO TO 40
      R = ATPB * PG(I)**PEXP + CT
      BURNL = R * TWODT
C
C *** UPDATE GRAIN LENGTH
      UXLB(I) = XLB(I) - BURNL
C *** SEE IF GRAIN HAS SPLIT INTO SPLINTERS
C *** NOTE THAT OLD DIMENSIONS ARE BEING TESTED
      IF(DOB(I) .LE. 3.0*DIB(I)) GO TO 20
C
C *** UPDATE OTHER DIMENSIONS
      UDOB(I) = DOB(I) - BURNL
      UDIB(I) = DIB(I) + BURNL
C
```

```
C *** CALCULATE OLD AND NEW VOLUMES OF A GRAIN.
      VOLD = PIDF * XLB(I) * (DOB(I) * DOB(I) - FN * DIB(I) * DIB(I))
      VNEW = PIDF * UXLB(I) *(UDOB(I) * UDOB(I)-FN*UDIB(I) *UDIB(I))
      GO TO 30
C
C
C *** CALCULATE OLD AND NEW GRAIN VOLUMES.
C     AFTER THE GRAIN HAS SPLINTERED, VALUES FOR THE CROSS-SECTIONAL
C     AREA OF THE PARTICLES GO INTO ARRAY DOB AND VALUES FOR PERIMETER
C     GO INTO ARRAY DIB. IF THE GRAIN HAS JUST SPLINTERED, AREA AND
C     PERIMETER HAVE TO BE INITIALIZED. IF THE GRAIN HAS JUST
C     SPLINTERED, DOB(I) IS ABOUT 3.0 * DIB(I) AND IF NOT
C     DOB(I) WILL BE LESS THAN DIB(I).
 20   CONTINUE
      IF(DOB(I) .LE. DIB(I)) GO TO 25
      WHOLEC = .FALSE.
      AREA = PIDF * (DOB(I) * DOB(I) - FN * DIB(I)*DIB(I))
      DI(I) = 3.14 * (DOB(I) + FN * DIB(I))
      DO(I) = AREA
C
 25   DELR = BURNL * 0.5
      UDOB(I) = DOB(I) - DIB(I) * DELR
C
      IF(UDOB(I) .GE. 1.0E-7) GO TO 27
      UDOB(I) = 0.0
      UDIB(I) = 0.0
      UXLB(I) = 0.0
      UPHI(I) = 1.0
      GO TO 50
C
   27 CONTINUE
C *** ASSUME THAT THE RATIO OF PERIMETER SQUARED TO CROSS-SECTIONAL
C     AREA IS CONSTANT
      UDIB(I) = SQRT( DIB(I) * DIB(I) / DOB(I) * UDOB(I))
C
C *** VOLUME IS LENGTH TIMES CROSS-SECTIONAL AREA
      VOLD = XLB(I) * DOB(I)
      VNEW = UXLB(I) * UDOB(I)
C
C
 30   CONTINUE
      IF(VNEW .LE. 0.0) GO TO 40
C
      DELTAV = VOLD - VNEW
C
C     CALCULATE NUMBER OF GRAINS PER GRID/VOLUME OF GRID
      PNDV = (1.0 - PHI(I))/VOLD
      TEMP = PNDV * DELTAV
C
C     CALCULATE GAS MASS GENERATED BY BURNING PROPELLANT/GRID VOLUME
```

```
      PMDOT(I) = TEMP * RHOP
      CUMMP=CUMMP+TEMP*RHOP*DX*AMASS(I)
      DOTMH = DOTMH + PMDOT(I) * HMB1
      DOTMM = DOTMM + PMDOT(I)
C
C     UPDATE POROSITY
      UPHI(I) = PHI(I) + TEMP
      GO TO 50
 40   UPHI(I) = PHI(I)
 50   CONTINUE
C *** LOGIC FOR SINGLE-PERF PROPELLANT
      IF(PHI2(I) .GE. 0.99999) GO TO 95
      R = ATPB2 * PG(I)**PEXP2 + CT2
      BURNL = R * TWODT
C
C *** UPDATE GRAIN DIMENSIONS
      UXLB2(I) = XLB2(I) - BURNL
      UDOB2(I) = DOB2(I) - BURNL
      UDIB2(I) = DIB2(I) + BURNL
C
C *** CALCULATE OLD AND NEW VOLUMES OF A GRAIN
      VOLD = PIDF * XLB2(I) * (DOB2(I)*DOB2(I) - DIB2(I)*DIB2(I))
      VNEW = PIDF * UXLB2(I)*(UDOB2(I)*UDOB2(I) - UDIB2(I)*UDIB2(I))
      IF(VNEW .LE. 1.0E-10) GO TO 80
      DELTAV = VOLD - VNEW
      PNDV = (1.0 - PHI2(I))/VOLD
      TEMP = PNDV * DELTAV
C
C *** ADD GAS GENERATED BY SINGLE-PERF PROPELLANT/GRID-VOLUME
C     TO THAT GENERATED BY MULTI-PERF PROPELLANT
C
      PMDOT(I) = PMDOT(I) + TEMP * RHOP2
      CUMSP=CUMSP+TEMP*RHOP2*DX*AMASS(I)
      DOTMH = DOTMH + TEMP * RHOP2 * HMB2
      DOTMM = DOTMM + TEMP * RHOP2
C
C *** UPDATE POROSITY
C
      UPHI2(I) = PHI2(I) + TEMP
      GO TO 100
 80   CONTINUE
      UXLB2(I) = 0.0
      UDOB2(I) = 0.0
      UDIB2(I) = 0.0
      UPHI2(I) = 1.0
      GO TO 100
C
 90   UPHI(I) = PHI(I)
 95   UPHI2(I) = PHI2(I)
 100  CONTINUE
      IF(DOTMM.GT.0.0) HMB = DOTMH/DOTMM
C
      RETURN
      END
C
C
C
C
C
```

```
      SUBROUTINE DRAG(DRAGX,INB,I,J)
C
C    SUBROUTINE DRAG CALCULATES CURRENT AND UPDATED VALUES FOR DRAG
C    IN THE AXIAL DIRECTION AND UPDATES UTDT AND VTDT.
C    THE ACTUAL DRAG USED IN THE FINITE DIFFERENCE CALCULATIONS IS
C    AN AVERAGE OF THE CURRENT AND UPDATED VALUES.
C
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UPBAVE,
     $  UTDT,VBGAVE,VTDT
      COMMON/CLOCK/TIME,DELT
      COMMON/DRGCON/VTSG
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0.
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL INB
      DATA GRAV/32.16/
      IF(PHIAVE.GT.0.98) RETURN
C
C    THE LOGICAL VARIABLE INB IS .TRUE. IF SUBROUTINE DRAG WAS CALLED
C    FROM A ONE-DIMENSIONAL SYSTEM WHERE THERE IS NO RADIAL VELOCITY
C    AND .FALSE. IF CALLED FROM A SYSTEM WHERE THERE IS RADIAL VELOCITY.
C
C
C    SINCE A FIXED VALUE OF DM OCCURS IN COMMON BLOCK INPUTS, IF THAT
C    COMMON BLOCK IS EVER PUT INTO SUBROUTINE DRAG, THE DM HERE SHOULD
C    BE GIVEN ANOTHER NAME.
C
C
C    A AND B ARE THE SAME IN BOTH THE AXIAL AND RADIAL CALCULATIONS.
C     THE ROUTINE CALLING DRAG SHOULD HAVE ALREADY CHECKED THAT PHIAVE
C    IS NOT 1.0.
```

```
      IF(INB) GO TO 5
C    TOTAL POROSITY WAS PUT INTO PHIBG BEGORE THE PATH SUBROUTINES WERE
C    CALLED.  SEPARATE OUT THE POROSITY OF MULTI-PERF PROPELLANT.
      PHIMP = PHIBG(I,J) + 2.0 - PHIBP(I,J) - PHIBG2(I,J)
C
C    CALCULATE EFFECTIVE DIMENSIONS OF THE PROPELLANTS.
      IF(DO(I,J) .GT. DI(I,J)) DM = 1.5*DO(I,J)*DO(I,J)*XL(I,J)
      IF(DO(I,J) .LE. DI(I,J)) DM = 0.844*(DOO - DIO)**2*XL(I,J)
      IF(DM .LE. 0.) DM = 1.0E-5
      DM = DM**0.333
      VPROP =0.7854*XL(I,J)*(DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J))
      IF(VPROP .LT. 1.0E-5) VPROP = 1.0E-5
C
      DM2 = 1.5*DO2(I,J)*DO2(I,J)*XL2(I,J)
      IF(DM2 .LE. 0.) DM2 = 1.0E-5
      DM2 = DM2**0.333
      VPROP2 = 0.7854*XL2(I,J)*(DO2(I,J)*DO2(I,J) - DI2(I,J)*DI2(I,J))
      IF(VPROP2 .LT. 1.0E-5) VPROP2 = 1.0E-5
C
      VBP=4.189*BPRAD(I,J)*BPRAD(I,J)*BPRAD(I,J)
      IF(VBP   .LT.1.E-5)VBP=1.E-5
C
      TERMP = (1.0 - PHIMP)/VPROP
      TERMP2 = (1.0 - PHIBG2(I,J))/VPROP2
      TERMBP=(1.0-PHIBP(I,J))/VBP
      IF((TERMP + TERMP2 + TERMBP) .LT. 1.0E-5) TERMP = 1.0E-5
      VTSGG = GRAV*(DM*TERMP + DM2*TERMP2 + 2.0*BPRAD(I,J)*TERMBP)/
     $  (TERMP + TERMP2 + TERMBP)
      GO TO 7
 5    CONTINUE
      PHIMP=PHI(I)-PHI2(I)+1.0
       IF(DOB(I).GT.DIB(I))DM=1.5*DOB(I)*DOB(I)*XLB(I)
      IF(DOB(I).LE.DIB(I))DM=0.844*(DOO-DIO)**2*XLB(I)
      IF(DM.LE.0.0)DM=1.0E-5
      DM=DM**0.333
      VPROP=0.7854*XLB(I)*(DOB(I)*DOB(I)-FN*DIB(I)*DIB(I))
      IF(VPROP.LT.1.0E-5) VPROP=1.0E-5
C
      DM2=1.5*DOB2(I)*DOB2(I)*XLB2(I)
      IF(DM2.LE.0.0) DM2=1.0E-5
      DM2=DM2**0.333
      VPROP2=0.7854*XLB2(I)*(DOB2(I)*DOB2(I)-DIB2(I)*DIB2(I))
      IF(VPROP2.LT.1.0E-5) VPROP2=1.0E-5
C
      TERMP=(1.0-PHIMP)/VPROP
      TERMP2=(1.0-PHI2(I))/VPROP2
       IF(TERMP+TERMP2.LT.1.0E-5)TERMP=1.0E-5
      VTSGG=GRAV*(DM*TERMP+DM2*TERMP2)/(TERMP+TERMP2)
   7  CONTINUE
      IF (VTSGG.GT.VTSG)VTSGG=VTSG
      CONST=4.0*(1.0-PHIAVE)/VTSGG
C
```

```
C
      A = CONST*RHOTDT
      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
 60   DISCRP = B*B + 4.0*A*C
      VTDT = (B - SQRT(DISCRP))/(A+A)
      RETURN
C
      END
      SUBROUTINE FSURT2
C
C *** FSURT2 IS IDENTICAL TO BSURA2 EXCEPT THE BOUNDARY IS AT GRID I + 1
C *** INSTEAD OF I - 1
C
      COMMON/AVGDT/RHOTDT,PHIRHO,PHIAVE,RHOAVE,UBGAVE,UTDT,VBGAVE,VTDT
      COMMON/CHAM/I ,J ,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RR0,CV0,CVH
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
      NAMELIST/FSURCK/BUGGER,GAM,I,J,F4,F5,G4,G5,E4,C4,C5,B4,RHOTDT,
     $  PHITDT,CHBGTD,QG
C
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,J),HBG(I,J),TDUM,
     $       RHOBG(I,J),UBG(I,J),VBG(I,J),GAM,CP,2)
      BUGGER = (GAM - 1.0) / TWOGJ
      IM1 = I - 1
C
C     IN THIS SUBROUTINE PHIBG REPRESENTS THE TOTAL POROSITY, NOT JUST
C     POROSITY OF THE PROPELLANT.
      F4 = PHIBG(IM1,J)
      F5 = PHIBG(I,J)
      G4=RHOBG(IM1,J)
      G5=RHOBG(I,J)
      H4=F4*G4*UBG(IM1,J)
      EI4=HBG(IM1,J)-PCH(IM1,J)/G4/778.0
      EI5=HBG(I,J)-PCH(I,J)/G5/778.0
      C4=G4*EI4
      C5=G5*EI5
      DENOM=2.0
      PHIAVE=(F4+F5)/DENOM
       RHOAVE=(G4+G5)/DENOM
      RHOAVE = (G4 + G5) / DENOM
      UBGAVE = 0.0
      UPBAVE = (UPB(IM1,J) + UPB(I,J))*0.5
C
      PHITDT = PHIBTD(I,J) + PHI2TD(I,J) + PHIPTD(I,J) - 2.0
```

```
C
C *** DO NOT MULTIPLY DELT*DOTMIG(I)/DVAXIS BY 2.0 AS IN RSURA2 BECAUSE
C *** GRID NGX HAS VOLUME DVAXIS WHEREAS GRID 1 HAS VOLUME DVAXIS/2.
      RHOTDT=(F5*RHOAVE+DTDX*H4+DELT*DOTMIG(I)/
     $DVAXIS+DOTMB(I,J)+DOTMP(I,J))/PHITDT
      QQ = QBAG(1,J)
      IF(RHOTDT .LT. 0.0) WRITE(6,FSURCK)
C
      PHIRHO = PHITDT*RHOTDT
C
      UTDT = 0.0
      VTDT = 0.0
C
      HIGN = HBG(I,1)
      IF (DOTMIG(I) .GT. 0.00001) HIGN = HBG(I,2)
      ETDT=(F5*(C4+C5)/DENOM+DTDX*(H4*EI4
     $+PCH(IM1,J)*(F4*UBG(IM1,J)+(1.0-F4)*UPB(IM1,J))/778.0)
     $+DELT*(DOTMIG(I)*HIGN/DVAXIS-QBAG(I,J))
     $+DOTMB(I,J)*HMB+DOTMP(I,J)*HBP)/PHIRHO
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PN,       ETDT,TDUM,
     $RHOTDT,UTDT,0.0,GAM,CP,4)
      HBGTD(I,J)=ETDT+PN/RHOTDT/778.0
      CHBGTD = HBGTD(I,J)
      IF(CHBGTD .LT. 0.0) WRITE(6,FSURCK)
C
      RHOBTD(I,J) = RHOTDT
      UBGTD(I,J) = UTDT
      VBGTD(I,J) = VTDT
      RETURN
      END
      SUBROUTINE GSPROP(RO,RRO,R,CV0,CVH,CV,P,H,T,RHO,U,V,GAM,CP,IPRCP)
      COMMON/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMF,WMBP,
     $  GAM1B,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLE
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      DATA XJUL/778.0/
C
C     IPROP = 1 - GIVEN T AND P
C     IPROP = 2 - GIVEN H AND RHO
C     IPROP = 3 - GIVEN H AND P
C     IPROP = 4 - GIVEN E AND RHO.
C
      GAM=GAM1B
C
C     R, CV,CP, AND GAM ARE TO BE CALCULATED.  ALSO, OF T, P, H, AND RHO
C     THE TWO THAT ARE NOT GIVEN ARE TO BE CALCULATED.
C
      GO TO (10,20,30,40),IPROP
C
C     THE EQUATIONS FOR IPROP=1 ARE VALID ASSUMING GAS VELOCITY IS 0.
   10 CONTINUE
C *** THIS OPTION IS USED ONLY ONCE IN CHSET WHERE NEARLY
C *** IDEAL CONDITIONS EXIST AND THE CO-VOLUME IS ASSUMED EQUAL
C *** TO ZERO.
      R=1545.3/WMOLE
      RHO=P/(R*T)
      E=R*T/(GAM-1.0)/XJUL
      H=E+P/(RHO*XJUL)
      RETURN
C
   20 CONTINUE
      R=1545.3/WMOLE
      COVOL=A0+A1*RHO+A2*RHO**2+A3*RHO**3
      P=RHO*XJUL*(GAM-1.0)*(H-U*U/TWOGJ)/(GAM-RHO*COVOL)
      T=P*(1.0/RHO-COVOL)/R
      RETURN
C
   30 CONTINUE
      RHO=P/((GAM-1.0)/GAM*(H-U**2/TWOGJ))/XJUL
   31 COVOL=(A0+A1*RHO+A2*RHO**2+A3*RHO**3)
      RHOS=RHO
      E=H-P/(RHO*XJUL)-U**2/TWOGJ
      RHO=P/((GAM-1.0)*E*XJUL+COVOL*P)
      IF(ABS(RHO-RHOS)/RHO.GT.0.01) GO TO 31
       R=1545.3/WMOLE
      T=P*(1.0/RHO-COVOL)/R
      HTEMP=H-U*U/TWOGJ
      GAM=HTEMP/(HTEMP-P/(RHO*XJUL))
      R=P/(RHO*T)
      RETURN
   40 CONTINUE
C *** INTERNAL ENERGY,E, IS INPUT TO THIS OPTION RATHER THAN ENTHALPY, H
      E=H
   41 CONTINUE
      R=1545.3/WMOLE
      COVOL=A0+A1*RHO+A2*RHO**2+A3*RHO**3
      T=(GAM-1.0)*(E-U**2/TWOGJ)/R *XJUL
      P=R*T/(1.0/RHO-COVOL)
      HTEMP=H-U*U/TWOGJ
      GAM=HTEMP/(HTEMP-P/(RHO*XJUL))
      R=P/(RHO*T)
      IF(IPROP.EQ.2) H=E+P/(RHO*XJUL)
      RETURN
      END
      SUBROUTINE HOLES
C*
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $    AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $   AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVC,
     $   AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $   DX,DR,NX,GJ,TWODT,HBP
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $   AH(60),FRACT(60)
      DIMENSION TA(85)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      DATA PI/3.141593/
C
C
C*    SUBROUTINE HOLES CALCULATES THE HOLE AREA EXPOSED TO EACH AXIAL
C*    GRID.  IT FORMS ARRAY AREAH.  AREAH(I) GIVES THE EXPOSED HOLE
C*    AREA AT GRID I.  THE HOLE AREA EXPOSED TO A GRID IS CALCULATED
C*    AS THE HOLE AREA LEFT OF THE RIGHT BOUNDARY OF THE GRID MINUS
C*    THE HOLE AREA LEFT OF THE LEFT BOUNDARY OF THE GRID.
C*
C*       NROWH - NUMBER OF ROWS OF HOLES
C*       XCL(I) - CENTER LINE OF HOLES IN ROW I
C*     RADHOL(I) - RADIUS OF HOLES IN ROW I
C*       TA(I) - HOLE AREA OF HOLE ROW I THAT ALREADY IS ASSIGNED TO
C*               A GRID.  INITIALLY TA(I) IS SET AS 0.  AFTER AREAXP IS
C*               CALCULATED, TA(I) IS SET TO AREAXP.
C*
C
      CALL CLEAR(TA(1),TA(85))
      CALL CLEAR(AREAH(1),AREAH(60))
C
C*    I1 INDEXES THE GRIDS
      X = -DX
      DO 40 I1 = 1,IENDB
C*
C*    X IS THE COORDINATE OF THE CENTER OF GRID I1
      X = X + DX
      GRDEND = X + .5*DX
C*
C*    DETERMINE THE AREA OF EACH ROW OF HOLES THAT LIES BEFORE THE
C*    END OF THE GRID
C
C*    I2 INDEXES THE ROWS OF HOLES
      DO 30 I2 = 1,NROWH
C*
C*    IF THE END OF GRID I1 IS TO THE LEFT OF HOLES IN ROW I2 (AND THUS
C*    THE REST OF THE ROWS) NO MORE HOLE AREA IS EXPOSED TO GRID I1
      IF(GRDEND .LE. XCL(I2) - RADHOL(I2)) GO TO 40
      FNH = FLOAT(NHOLES(I2))
```

```
C
C*  IF THE END OF THE GRID IS TO THE RIGHT OF THE HOLES IN ROW I2, ADD
C*  THE ENTIRE AREA OF THE HOLES IN ROW I2 TO AREAXP
C*  OTHERWISE THE END OF THE GRID LIES WITHIN HOLES IN ROW I2 AND
C*  THE AMOUNT OF EXPOSED AREA WILL BE DETERMINED.
      IF(GRDEND .LT. XCL(I2) + RADHOL(I2)) GO TO 10
      AREAXP = FNH*PI*RADHOL(I2)*RADHOL(I2)
      GO TO 20
C
C
C*
 10   TERM = GRDEND - XCL(I2)
C*  FIND HALF OF THE CENTRAL ANGLE TO THE CHORD AT THE GRID END
      THETA = ACOS(ABS(TERM)/RADHOL(I2))
C*
C*  FIND AREA OF CIRCULAR SECTOR WHOSE CENTRAL ANGLE IS 2*THETA AND
C*  AREA OF TRIANGLE FORMED BY THE CHORD AND RADII
      RSQ = RADHOL(I2)*RADHOL(I2)
      AREAS = THETA*RSQ
      AREAT = SQRT(RSQ - TERM*TERM)*ABS(TERM)
C*
C*  EXPOSED AREA FROM ONE HOLE IS SECTOR AREA - TRIANGLE AREA OR THE
C*  HOLE AREA MINUS THIS
      AREAXP = (AREAS - AREAT)*FNH
      IF(TERM .GT. 0.0) AREAXP = PI*RSQ*FNH - AREAXP
C*
 20   AREAH(I1) = AREAH(I1) + AREAXP - TA(I2)
C*
      TA(I2) = AREAXP
 30   CONTINUE
C
      IF( AREAH(I1) .LT. 0.00001 ) AREAH(I1) = 0.0
 40   CONTINUE
C
C
C   CALCULATE AH(I), THE BELL TUBE AREA IN GRID 1.  AH(I) DOES NOT
C   INCLUDE AREA OF PSEUDO-HOLES.
      CALL CLEAR(AH(1),AH(60))
      DO 60 I=IBEGB,IENDB
         AH(I) = AREAH(I)
 60   CONTINUE
C
C   IF THERE IS A PSEUDO HOLE AT GRID IBEGB OR AT GRID IENDB, ITS
C   AREA SHOULD NOT HAVE BEEN INCLUDED.
C *** THERE ARE NO PSEUDOHOLES AT GRID IENDB.  THERE IS ONE AT IBEGB.
C *** IT HAS RADIUS (SHOULD BE 0.0) RADHOL(1).
      APSEUD = PI*RADHOL(1)*RADHOL(1)
      AH(IBEGB) = AH(IBEGB) - APSEUD
C
```

```
C     CALCULATE FRACT(I), THE FRACTIONAL VOLUME OF A GRID NOT UNDER THE
C     INFLUENCE OF THE BELL TUBE HOLES.  THE VOLUME CONSIDERED TO BE
C     INFLUENCED IS BASED ON THREE TIMES THE HERMISPHERICAL VOLUME
C     COMPUTED FROM THE RADIUS OF A TUBE HOLE.
      DO 70 I = 1,IENDB
           FRACT(I) = 1.0 - 2.0*SQRT(AH(I)*AH(I)*AH(I)/PI)/(AREAR(I)*CX)
           IF(FRACT(I) .LT. 0.0) FRACT(I) = 0.0
 70   CONTINUE
      RETURN
      END
      SUBROUTINE HOLSET
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $  AH(60),FRACT(60)
      DIMENSION DTEMP(50),NTEMP(50),RTEMP(50)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
C     THE DISTANCE (IN INCHES) BETWEEN THE CENTER LINE OF HOLES IN ROW I
C     AND THE CENTER LINE OF THE PREVIOUS ROW OF HOLES WAS INPUT INTO
C     XCL(I).  XCL(1) CONTAINS THE DISTANCE OF THE CENTER LINE OF HOLES
C     IN ROW 1 FROM THE BEGINNING OF THE BELL TUBE.
C     NUMBER OF HOLES IN ROW I WAS INPUT INTO NHOLES(I).
C     RADIUS OF HOLES IN ROW I WAS INPUT INTO RADHOL(I).
C     PSEUDO HOLES WILL BE PUT BEFORE AND AFTER THE BELL TUBE AND
C     TREATED THE SAME AS HOLES ON THE BELL TUBE.
C     THE POSITION OF THE CENTER LINE OF HOLES IN ROW I (IN FEET)
C     WILL BE PUT INTO XCL(I).
C
C     TEMPORARILY STORE THE INPUT DATA ABOUT THE HOLES.
      IF(NROWH .EQ. 0) GO TO 102
      DO 100 I=1,NROWH
           DTEMP(I) = XCL(I)/12.
           NTEMP(I) = NHOLES(I)
           RTEMP(I) = RADHOL(I)/12.
  100 CONTINUE
  102 CONTINUE
C
C     PUT PSEUDO HOLES BEFORE THE BELL TUBE.
C     RADPS, THE RADIUS OF THE PSEUDO HOLES, IS CALCULATED TO GIVE ABOUT
C     TWICE AS MUCH AREA AS HOLES ON THE BELL TUBE.
      RADPS = 1.0*DIAMBT*SQRT(DX)
      IPS1 = IBEGB
C
C     IF THE BELL TUBE BEGINS BEFORE GRID POINT IBEGB, THERE SHOULD BE NO
C     PSEUDOHOLE AT GRID POINT IBEGB.
      DO 105 I=1,IPS1
           XCL(I) = FLOAT(I-1)*DX
           NHOLES(I) = 1
           RADHOL(I) = RADPS
  105 CONTINUE
      IF(BELBEG.LT.DX)RADHOL(1)=RADHOL(1)*SQRT(BELBEG/DX)
C
C     SET UP HOLES ON THE BELL TUBE.
      IF(NROWH .EQ. 0) GO TO 112
      I1 = IPS1 + 1
      XCL(I1) = BELBEG + DTEMP(1)
      NHOLES(I1) = NTEMP(1)
      RADHOL(I1) = RTEMP(1)
      IF(NROWH .EQ. 1) GO TO 115
      DO 110 I=2,NROWH
           I1 = I1 + 1
           XCL(I1) = XCL(I1-1) + DTEMP(I)
           NHOLES(I1) = NTEMP(I)
           RADHOL(I1) = RTEMP(I)
           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 *** THERE SHOULD BE NO PSEUDOHOLES AT GRID IENDB.
           IF(NROWH.EQ. 85)GO TO 140
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 MFLOW
C
C
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/GASCON/RO,RRO,CVO,CVH
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $  AH(60),FRACT(60)
      COMMON/INPUTS/C1,C2,C3,C4,TO,TIGN,QCONS,RHOP,PHIO,TF,CA,RHOO,
     $  HO,FO,UO,GTRHOP,HW,DM,DM2,TIGNBP,GBCONS,TOTM,DIFFPR
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), GBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VRGTL(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHJBP(60,5),
     5 PHIFTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      DIMENSION DOTM(60)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
C
C
C     SUBROUTINE MFLOW CALCULATES THE MASS OF GAS FLOWING BETWEEN TWC
C     RADIAL ROWS THROUGH HOLES IN THE BELL IGNITER TUBE ANC THROUGH
C     PSEUDO HOLES.  MFLOW CALCULATES THE ENTRIES OF ARRAY DOTMIG ANC IF
C     CHAM3 IS TRUE, THE ENTRIES OF ARRAY DOTMBG.
C     DOTMIG(I) IS THE GAS MASS FLOWING BETWEEN GRIDS (I,1) AND (I,2).
C     + DOTMIG(I) OCCURS IN THE PATH ROUTINES FOR GRIDS WHERE J=1 ANC
C     - DOTMIG(I) WHERE J=2.
C     DOTMBG(I) IS THE GAS MASS FLOWING BETWEEN GRIDS (I,2) AND (I,3)
C     WHEN RADIAL ROWS 2 AND 3 ARE CONSIDERED INDEPENDENT ONE-DIMENSICNAL
C     SYSTEMS.
C     + DOTMBG(I) OCCURS IN THE PATH ROUTINES FOR GRIDS WHERE J=2 ANC
C     - DOTMBG(I) CCCURS WHERE J=3.
C     ARRAYS DOTMIG AND DOTMBG ARE CLEARED IN MAIN AT EACH TIME INTERVAL.
C
      LOGICAL LMFLOW(60)
      DATA LMFLOW/60*.FALSE./
C
C *** LOGICAL VARIABLE LMFLOW(I) WILL BE SET .FALSE. INITIALLY-- WHEN
C *** PRESSURE DIFFERENCE IS SUFFICIENT TO BURST LINER LMFLOW(I) WILL BE
C *** SET .TRUE. AND REMAIN .TRUE.
C
      NGX1 = NGX - 1
      ID = 1
      IF(CHAM3) ID = 2
      J = 1
      GO TO 20
```

```
C
 5     ID = 1
       J = 2
C
 20    CONTINUE
       JP1 = J + 1
       CALL CLEAR(DOTM(1),DOTM(60))
C
C
C************************************************************************
C     COMPUTE GAS MASS FLOW PER UNIT AREA
C************************************************************************
C
C *** MASS FLOW COMPUTATIONS ARE NOT DONE FOR GRIDS BEYOND THE END OF TH
C *** BELL TUBE
C
       DO 80 I = 1,IENDB
C
           PR = PCH(I,J)
           PA = PCH(I,JP1)
C
C *** IF LINER IS BROKEN DO NOT CHECK OR BURSTING. IF LINER IS STILL
C *** INTACT CHECK FOR BURSTING.
C
       IF( LMFLOW(I)) GO TO 9
       IF( ABS(PR - PA) .LT. DIFFER) GO TO 80
       LMFLOW(I) = .TRUE.
     9 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 = 0.55*PR
           IF( PCOMP .LT. PA ) PCOMP = PA
C
           CALL GSPROP(RO,RRO,R,CVO,CVR,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 CALCULATED.
            FM = 1.0
            IF (PA .LT. PSTAT) GO TO 10
            FM = SQRT( ((PR/PA)**(GMMM1/GMM) - 1.0)*CONS2 )
            PSTAT = PA
C
 10         HSTAT = HFR/(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,CV0,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
C
            FM = 1.0
            IF( PR .LT. PSTAT ) GO TO 50
            FM = SQRT( ((PA/PR)**(GMMM1/GMM) - 1.0)*CONS2 )
            PSTAT = PR
 50         HSTAT = HAA/(1.0 + FM*FM/CONS2)
C
C     SINCE GAS IS FLOWING FROM GRID (I,J+1) TO GRID (I,J),DOTM SHOULD
C     BE POSITIVE.
C****IS PHIBG(I,2) OKAY
            DOTM(I) = .203*GMM*PSTAT*FM*CA/SQRT(GMMM1*HSTAT*PHIBG(1,2))
C
 80     CONTINUE
        IF(CHAM3 .AND. ID .EQ. 1) GO TO 120
C
C
C***********************************************************************
C     FILL ARRAY DOTMIG
C***********************************************************************
```

```
C
C     AVERAGE GAS MASS FLOW AT EACH GRID.
      DOTMIG(1) = 0.5*(DOTM(1) + DOTM(2))*AREAH(1)
      DO 100 I=2,NGX1
          DOTMIG(I) = 0.25*(DOTM(I-1) + DOTM(I) + DOTM(I) + DOTM(I+1))*
     $        AREAH(I)
 100  CONTINUE
      DOTMIG(NGX) = 0.5*(DOTM(NGX1) + DOTM(NGX))*AREAH(NGX)
C
      GO TO (200,5),ID
C
C
C*************************************************************************
C     FILL ARRAY DOTMBG
C*************************************************************************
C
 120  CONTINUE
      DOTMBG(1) = 0.5*(DOTM(1) + DOTM(2))*AREAH2
      DO 130 I=2,NGX1
          DOTMBG(1) = 0.25*(DOTM(I-1) + DOTM(I) + DOTM(I) + DOTM(I+1))*
     $        AREAH2
 130  CONTINUE
      DOTMBG(NGX) = 0.5*(DOTM(NGX1) + DOTM(NGX))*AREAH2
C
 200  RETURN
C
      END
      SUBROUTINE MOTION
C
      COMMON/DXVALU/DXPS
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BOREL8,DT2BD,DTDSQ,XLEAR
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/RO,FRO,CVO,CVH
      COMMON/GRIDNX/DXPRIM
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XLO,  DOO, DIO,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/MOCON/CON3,CON4,CON5,AREAPB,ZO,WOB,XOB,FDMAX,PINER,
     $  CF,RADPB,PMASS,XINT,PINT,XLO,PLO,CON6
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL PRI1,IDEBUG
      LOGICAL NXGE2
      DATA NXGE2/.FALSE./
      DATA BPRES/2116./
      DATA CSPD/1100./
      DATA DELJ/0./
      DATA IPRI/0/
      DATA XJUL/778./
      DATA GRAV/32.16/
C
C    DETERMINE PROPULSIVE FORCE ACTING ON THE PROJECTILE
      F = PG(NX)*AREAPB
C
C *** CALCULATE ROTK AND THETA WHICH ARE RELATED TO THE TWIST OF THE
C *** RIFLING. THE TWIST OF THE RIFLING VARIES WITH THE TRAVEL DOWN THE
C ***  BARREL.
C
C     XP - XOB IS THE TRUE DISTANCE OF THE PROJECTILE DOWN THE BARREL.
      TRUXP = XP - XOB
      IF(TRUXP .LT. 8.13317) DYDX = .01042272*TRUXP + .08976038
      IF(TRUXP .GE. 8.13317) DYDX = 0.17453
      DYDX=0.15708
      THETA  = ATAN(DYDX)
      ROTK = 2.0*DYDX/BORED
      CS = COS(THETA)
      SN = SIN(THETA)
      CON1 = CS - CF*SN
      CON2 = RADPB/(PMASS/GRAV*RADPB*CON1 + PINER*ROTK*(SN + CF*CS))
```

```
C
C     DETERMINE FDPRIM, THE ENGRAVING FORCE AND SLIDING RESISTANCE
C     XOB IS THE INITIAL PROJECTILE POSITION--POSITION WHERE ENGRAVING
C     BEGINS.  WOB IS THE LENGTH OF THE ENGRAVING BAND.
       IF(TRUXP .GT. WOB) GO TO 30
       FDPRIM = Z0 + CON3*TRUXP
       GO TO 50
C
 30    CONTINUE
       IF(TRUXP .GT. XINT) GO TO 40
       FDPRIM = FDMAX - CON4*(TRUXP - WOB)
       GO TO 50
C
 40    CONTINUE
       IF(TRUXP .GT. XLO) GO TO 45
       FDPRIM = PINT*AREAPB - CON6*(TRUXP - XINT)
       GO TO 50
C
 45    CONTINUE
       DELC = 0.2*DELU
       CSPD = CSPD + DELC
       DELP = 1.4*BPRES*DELU/(CSPD - 0.6*DELU)
       BPRES = BPRES + DELP
       FDPRIM=(BPRES+PLO)*AREAPB
C
 50    CONTINUE
       IIPRI = MOD(IPRI,50)
       IF(IDEBUG(33) .AND. IIPRI .EQ. 0) WRITE(6,5000) IPRINT,FDPRIM
       IPRI = IPRI + 1
C     DETERMINE PROJECTILE AXIAL ACCELLERATION
       ACC = CON2*(F*CON1 - FDPRIM)
C
C     IF THERE IS NO MOVEMENT, NX AND DXPRIM REMAI  THE SAME.
C****IF ACC IS USED ELSEWHERE, MAKE IT NONNEGATIVE BEFORE RETURNING
        IF(ACC.LE.0.0.AND.VP.LE.0.0) RETURN
C
C     PROJECTILE AXIAL VELOCITY
       VP0 = VP
       VP = VP0 + ACC*DELT
C
C     PROJECTILE ANGULAR VELOCITY
       OMEGA = ROTK*VP*9.5493
C
C     PROJECTILE AXIAL POSITION
       XP = XP + VP0*DELT + ACC*CON5
C
C     DETERMINE THE NUMBER OF GRIDS, NX, AND THE SIZE OF THE LAST GRID
C     NGC SAVES THE NUMBER OF GRIDS BEFORE PROJECTILE MOTION
       NGC = NX
C
C     NUMBER OF GRIDS IN DECIMAL FORM AFTER PROJECTILE MOTION
       XNG = XP/DX + 1.0
```

```
C
C    NUMBER OF TRUE GRIDS
      NX = XNG
C
C
C     SAVE THE PREVIOUS DXPRIM
      DXPS = DXPRIM
C
C     DXPRIM IS THE WIDTH OF THE LAST GRID.
C     DXPRIM WILL BE .GE. DX AND .LT. 2*DX.
      DXPRIM = (XNG - FLOAT(NX - 1))*DX
C
C
C
C    FILL THE ARRAY VALUES AT GRID NX
C
C    GET UPDATED PRESSURE AND GAM AT PREVIOUS GRID NX
      IF(NXGE2) GO TO 80
      IF(NX .EQ. 1) GO TO 100
          NXGE2 = .TRUE.
          UHG(1) = HG(1)
          URHOG(1) = RHOG(1)
          UUG(1) = UG(1)
 80   CONTINUE
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PG(NGC),UHG(NGC),TDUM,URHOG(NGC),
     $            UUG(NGC),0.0,GAM,CP,2)
      HTEMP=UHG(NGC)-UUG(NGC)*UUG(NGC)/TWOGJ
      GAM=HTEMP/(HTEMP-PG(NGC)/(RHOG(NGC)*XJUL))
      GAM1 = GAM - 1.0
      DELVP = VP - VP0
      C = SQRT(GAM1*(GJ*UHG(NGC) - VP0*VP0*0.5))
      CPRIM = C - 0.5*GAM1*DELVP
C
C    GAS VELOCITY AT GRID NX IS THAT OF THE PROJECTILE
      UUG(NX) = VP
C
      TEMP = GAM*DELVP/(C + CPRIM)
C
C    NOTE THAT PG(NGC) HOLDS UPDATED PRESSURE AT PREVIOUS NX GRID
      PRES = PG(NGC)*(1.0 - TEMP)/(1.0 + TEMP)
C
      TEMP = VP*VP/TWOGJ
      UHG(NX) = CPRIM*CPRIM/(GAM1*GJ) + TEMP
C
      CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PRES,UHG(NX),TDUM,URHOG(NX),
     1 VP,0.0,GAM,CP,3)
C
      IF(NX .GT. NGC) UUP(NX) = UUP(NGC)
      IF(UUP(NX) .GT. UUG(NX)) UUP(NX) = UUG(NX)
C****THE FOLLOWING MAY NOT BE NECESSARY
      UAMASS(NX) = UAMASS(NX-1)
      UAMOM(NX) = UAMOM(NX-1)
      UAENER(NX) = UAENER(NX-1)
C
      IF(NX .GT. NGC) GO TO 90
      UPH1(NX) = 1.0 - (1.0 - UPH1(NX))*(DXPS - DX*0.5)/
     $  (DXPRIM - DX*0.5)
      UPH12(NX)=1.0-(1.0-UPHI2(NX))*(DXPS-DX*0.5)/(DXPRIM-DX*0.5)
      GO TO 100
C
 90   CONTINUE
      UPHI(NX) = 1.0 - (1.0 - UPHI(NGC))*(DXPS - DX*0.5)/
     $  (DXPRIM + DX*0.5)
      UPHI2(NX)=1.0-(1.0-UPHI2(NGC))*(DXPS-DX*0.5)/(DXPRIM+DX*0.5)
      UXLB(NX)=UXLB(NX-1)
      UDOB(NX)=UDOB(NX-1)
      UDIB(NX)=UDIB(NX-1)
      UXLB2(NX)=UXLB2(NX-1)
      UDOB2(NX)=UDOB2(NX-1)
      UDIB2(NX)=UDIB2(NX-1)
C
C     IF A NEW GRID HAS BEEN ADDED, GRID NX - 1 DOES NOT HAVE THE PRCPER
C     VALUES EXCEPT FOR UAMASS, UAMOM, AND UAENER (IN BNDLYR AREAS
C     AT GRID NX ARE SET TO THOSE OF NX - 1).  IN THIS CASE NGC IS NCW
C     NX - 1.
      FRACT = DX/(DX + DXPRIM)
      URHOG(NX - 1) = URHOG(NX - 2) + FRACT*(URHOG(NX) - URHOG(NX - 2))
      UUG(NX - 1) = UUG(NX - 2) + FRACT*(UUG(NX) - UUG(NX - 2))
      UHG(NX - 1) = UHG(NX - 2) + FRACT*(UHG(NX) - UHG(NX - 2))
      UUP(NX - 1) = UUP(NX - 2) + FRACT*(UUP(NX) - UUP(NX - 2))
      UPHI(NX - 1) = UPHI(NX)
      UPH12(NX-1)=UPHI2(NX)
C
C
 100  CONTINUE
      IF((XP - XOB) .GE. XLBAR) PRI1 = .TRUE.
      IF(.NOT.PRI1)RETURN
      IF(.NOT. IDEBUG(23)) RETURN
               IF((XP - XOB) .GE. XLBAR) WRITE(6,3000) TIME
               WRITE(6,4000) TIME
               WRITE(6,1000)
               PRES = PG(NX)/144.
      DISP = (XP - XOB)*12.
               WRITE(6,2000) ACC,VP,DISP,OMEGA,FDPRIM,PRES
 1000 FORMAT(///,20X,*INTERIOR BALLISTICS OUTPUT *,//,5X,*PROJECTILE*,
     1  10X,*PROJECTILE*,10X,*PROJECTILE*,10X,*ROTATIONAL*,10X,
     2  *PROJECTILE*,7X,*PRESSURE AT BASE*,/,4X,*ACCELERATION*,10X,
     3  *VELOCITY*,10X,*DISPLACEMENT*,10X,*VELOCITY*,14X,*DRAG*,12X,
     4  *OF PROJECTILE*)
 2000 FORMAT(6E20.10)
 3000 FORMAT(1H1,* THE PROJECTILE HAS GONE OUT OF THE BARREL AT TIME*,
     $  E14.8)
 4000 FORMAT(1H0,* TIME IS*,E14.8)
 5000 FORMAT(/,3X,*IPRINT = *,I10,10X,*FDPRIM = *,E14.5)
      RETURN
      END
      SUBROUTINE NEWDX
C
C    SUBROUTINE NEWDX IS CALLED WHEN THE BARREL GETS A 21ST GRID.
C    THE GRID SIZE DX IS DOUBLED, THE BARREL IS CUT DOWN TO 11 GRIDS,
C    AND THE NUMBER OF CHAMBER GRIDS IS HALVED.
C    A NEW TIME INTERVAL IS ALSO CALCULATED.
C
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ,XLBAR
      COMMON/CAREAS/ARROW1,ARROW2,ARROW3,ARTOT
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0.
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/GRIDNX/DXPRIM
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
      DATA GRAV/32.16/
C
      NX = 11
C
C    PUT THE BARREL ARRAY VALUES AT THE ODD-NUMBERED GRIDS INTO GRIDS
C    1 THROUGH 11.
      J=1
      DO 10 I=2,11
```

```
          J=J+2
          UPHI(I) = UPHI(J)
      UPHI2(I)=UPHI2(J)
          URHOG(I) = URHOG(J)
          UUG(I) = UUG(J)
          UHG(I) = UHG(J)
          UUP(I) = UUP(J)
      UDRAG(I)=UDRAG(J)
      UXLB(I)=UXLB(J)
      UDOB(I)=UDOB(J)
      UDIB(I)=UDIB(J)
      UXLB2(I)=UXLB2(J)
      UDOB2(I)=UDOB2(J)
      UDIB2(I)=UDIB2(J)
          UAMASS(I) = UAMASS(J)
          UAMOM(I) = UAMOM(J)
          UAENER(I) = UAENER(J)
 10   CONTINUE
C
C    REDUCE THE NUMBER OF GRIDS IN THE CHAMBER
C    PUT THE TOTAL HOLE AREA INTO THE NEW GRIDS.
      IF(NGX .EQ. 1) GO TO 50
      AREAH(1) = AREAH(1) + AREAH(2)
      AREAH2 = AREAH2 + AREAH2
      NGX = (NGX + 1)/2
C *** FOR THE 105 ONLY
      IENDB = NGX
      IPATH(IENDB,1) = 4
      NGP1 = NGX + 1
C
C    PUT THE CHAMBER ARRAY VALUES AT THE ODD-NUMBERED GRIDS INTO THE
C     NEW CHAMBER GRIDS.
      IF(NGX .EQ. 1) GO TO 50
      J=1
      DO 40 I=2,NGX
          J=J+2
          IF((J + 1) .LE. NGX) AREAH(I) = AREAH(J) + AREAH(J + 1)
          IF ((J + 1) .GT. NGX) AREAH(I) = AREAH(J)
          DO 30 K=1,NGR
              PHIBTD(I,K) = PHIBTD(J,K)
          PHI2TD(I,K) = PHI2TD(J,K)
          XL2TDT(I,K) = XL2TDT(J,K)
          DO2TDT(I,K) = DO2TDT(J,K)
          DI2TDT(I,K) = DI2TDT(J,K)
              RHOBTD(I,K) = RHOBTD(J,K)
              UBGTD(I,K) = UBGTD(J,K)
              VBGTD(I,K) = VBGTD(J,K)
              HBGTD(I,K) = HBGTD(J,K)
              UPBDT(I,K) = UPBDT(J,K)
              XLTDT(I,K) = XLTDT(J,K)
              DOTDT(I,K) = DOTDT(J,K)
              DITDT(I,K) = DITDT(J,K)
              PHIPTD(I,K) = PHIPTD(J,K)
              TZC(I,K) = TZC(J,K)
              TZC2(I,K) = TZC2(J,K)
 30       CONTINUE
              AREAR(I) = AREAR(J)
              IF(CHAM3) AREAGP(I) = AREAGP(J)
          AREAC(I) = AREAC(J)
          TZR(I) = TZR(J)
          TZR2(I) = TZR2(J)
 40   CONTINUE
C
C     PUT THE VOLUME LOST IN THE BARREL INTO CHAMBER ROW 2.
      ARROW2 = ARROW2 + 0.5*DX*BOREA
C
C    CHANGE DX, DELT, AND CONSTANTS DEPENDING ON THEM.
 50   CONTINUE
      DXPRIM = DXPRIM + DX
      DX = 2.0*DX
      DELT = 2.0*DELT
      TWODT = 2.0*DELT
      DTDX = DELT/DX
      T2DX = 0.5*DTDX
C
C
      CON5 = 0.5*DELT*DELT
      DTDSQ = DELT*BORED*BORED
      DT2BD = -0.5*DELT/BORED
      GTRHOP = GRAV*DELT/RHOP
C
C
C     CALCULATE THE CHAMBER VOLUME.  IF IT HAS CHANGED ADJUST THE AREAS.
      IF(ONED) GO TO 150
      ARR1 = (FLOAT(NGX) - 0.5)*DX*AREAAX
C
      ARR2 = AREAR(1)*DX*0.5
      IF(NGX .EQ. 1) GO TO 80
      DO 70 I = 2,NGX
      ARR2 = ARR2 + AREAR(I)*DX
 70   CONTINUE
C
 80   CONTINUE
      IF(CHAM2) GO TO 100
C
      ARR3 = AREAGP(1)*DX*0.5
      IF(NGX .EQ. 1) GO TO 100
      DO 90 I = 2,NGX
      ARR3 = ARR3 + AREAGP(I)*DX
 90   CONTINUE
C
 100  CONTINUE
```

```
C
C     ADJUST AREAAX AND ARRAY AREAC
      ADJUST = ARROW1/ARR1
      AREAAX = AREAAX*ADJUST
      DO 105 I = 1,NGX
      AREAC(I) = AREAC(I)*ADJUST
 105  CONTINUE
C
C     ADJUST ARRAY AREAR
      ADJUST = ARROW2/ARR2
      DO 110 I = 1,NGX
      AREAR(I) = AREAR(I)*ADJUST
 110  CONTINUE
      IF(CHAM2) GO TO 200
C
C     ADJUST ARRAY AREAGP
      ADJUST = ARROW3/ARR3
      DO 120 I = 1,NGX
      AREAGP(I) = AREAGP(I)*ADJUST
 120  CONTINUE
      GO TO 200
C
C
C     LOGIC WHEN ONED IS TRUE
 150  CONTINUE
      ART = AREAC(1)*DX*0.5
      IF(NGX .EQ. 1) GO TO 170
      DO 160 I = 2,NGX
      ART = ART + AREAC(I)*DX
 160  CONTINUE
C
 170  CONTINUE
      ADJUST = ARTOT/ART
      DO 180 I = 1,NGX
      AREAC(I) = AREAC(I)*ADJUST
 180  CONTINUE
C
C
 200  CONTINUE
C    FIX AREAS AND VOLUMES
      DVAXIS = AREAAX*DX
C
      IF(ONED) AREAC(NGP1) = BOREA
      AREAR(NGP1) = BOREA
         AREAC(NGP1) = -10.E+15
      IF(IDEBUG(12)) WRITE(6,2000) IPRINT,NGX
      NAMELIST/NEWCHK/DX,DELT,TWODT,DTDX,DTDR,T2DR,T2DX,TWOTDR,
     $  DXPRIM,AREAAX,
     $  CONS,DTDSQ,DT2BD,GTRHOP,DVAXIS,DVAXIT,AREACH
      IF(IDEBUG(13)) WRITE(6,NEWCHK)
      IF(.NOT. IDEBUG(14)) RETURN
      WRITE(6,2006)
      WRITE(6,2003) (AREAGP(I),I=1,NGP1)
      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)
 2000 FORMAT('1 NEWDX CALLED, IPRINT =',I5,/,' NGX =',I5)
 2002 FORMAT(///,' ARRAY AREAR',//)
 2003 FORMAT(9X,10F11.7,/)
 2004 FORMAT(///,' ARRAY AREAC',//)
 2005 FORMAT(///,' ARRAY AREAG',//,20X,5F11.7)
 2006 FORMAT(///,' ARRAY AREAGP',//)
      RETURN
      END
      SUBROUTINE ONEDIM
C
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSG,XLBAR
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RRO,CV0,CVH
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMF(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
C
C    THIS SUBROUTINE PUTS THE CORRECT CHAMBER AREAS INTO ARRAY AREAC,
C    PUTS VOLUME-WEIGHTED AVERAGES OF THE GAS ARRAYS INTO THE ARRAY
C    ELEMENTS AT GRID 1, CLEARS THE GAS ARRAYS AT THE OTHER GRIDS,
C    AND SETS NGR TO 1.
C
      DATA PIDF/.785398/
C
C
C
C**********************************************************************
C    FILL ARRAY AREAC
C**********************************************************************
C
      IF(CHAM3) GO TO 30
      DO 20 I = 1,IENDB
          AREAC(I) = AREAR(I) + AREAAX
 20   CONTINUE
      GO TO 50
C
 30   CONTINUE
      DO 40 I=1,NGX
          AREAC(I) = AREAR(I) + AREAAX + AREAGP(I)
 40   CONTINUE
C
 50   AREAC(NGX+1) = BOREA
C
C
C************************************************************************
C     FILL EACH CHAMBER ARRAY AT (I,1) WITH THE VOLUME-WEIGHTED AVERAGE
C     OF THE ARRAY VALUES AT I.
C************************************************************************
C
C
C     CALCULATIONS FOR CHAM2 TRUE
 400  CONTINUE
      DO 450 I=1,NGX
      IF (I .GT. IENDB) GO TO 441
          TZC(I,1) = TIGN + 1.0
          PTEMP1 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAAX
          PTEMP2 = (PHIBG(I,2) + PHIBP(I,2) - 1.0)*AREAR(I)
          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))/
     $         AREAC(I)
          TEMP = (1.0 - PHIBP(I,1))*AREAC(I)
          IF(TEMP .LT. 0.0001) GO TO 410
          BPRAD(I,1) = BPRAD(I,1)/TEMP
          GO TO 420
C
 410      CONTINUE
          BPRAD(I,1) = 0.0
C
 420      CONTINUE
          TEMP1 = (1.0 - PHIBG(I,1))*AREAAX
          TEMP2 = (1.0 - PHIBG(I,2))*AREAR(I)
          PHIBG(I,1) = (PHIBG(I,1)*AREAAX + PHIBG(I,2)*AREAR(I))/
     $         AREAC(I)
          TEMP3 = (1.0 - PHIBG(I,1))*AREAC(I)
          IF(TEMP3 .LT. 0.0001) GO TO 430
          UPB(I,1) = (UPB(I,1)*TEMP1 + UPB(I,2)*TEMP2)/TEMP3
          XL(I,1) = (XL(I,1)*TEMP1 + XL(I,2)*TEMP2)/TEMP3
          DO(I,1) = (DO(I,1)*TEMP1 + DO(I,2)*TEMP2)/TEMP3
          DI(I,1) = (DI(I,1)*TEMP1 + DI(I,2)*TEMP2)/TEMP3
          GO TO 440
 430      UPB(I,1) = 0.0
          XL(I,1) = 0.0
          DO(I,1) = 0.0
          DI(I,1) = 0.0
C
 440      CONTINUE
          PTEMP3 = (PHIBG(I,1) + PHIBP(I,1) - 1.0)*AREAC(I)
C
```

```
C****PTEMP3 SHOULD NEVER BE 0.0
          RHOSAV = (RHOBG(I,1)*PTEMP1 + RHOBG(I,2)*PTEMP2)/PTEMP3
          HBG(I,1) = (HBG(I,1)*PTEMP1*RHOBG(1,1) +
     $        HBG(I,2)*PTEMP2*RHOBG(I,2))/(PTEMP3*RHOSAV)
          UBG(I,1) = (UBG(I,1)*PTEMP1*RHOBG(1,1) +
     $        UBG(I,2)*PTEMP2*RHOBG(I,2))/(PTEMP3*RHOSAV)
          RHOBG(I,1) = RHOSAV
C
C    CALL GSPROP TO UPDATE PCH
  441 CONTINUE
          CALL GSPROP(R0,RR0,R,CV0,CVH,CV,PCH(I,1),HBG(I,1),TDUM,
     $        RHOBG(1,1),UBG(1,1),0.0,GAM,CP,2)
 450  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 PRIMER
      COMMON/CALLP/BPLEFT
      COMMON/PRMFLO/DOTMPM, UPRM
      COMMON/CLOCK/TIME,DELT
      COMMON/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMP,WMBP,
     $  GAMIB,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLE
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBOT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL BPLEFT
      DATA FORTPI/4.188790/
C
C
C
C     CALCULATE PRIMER VELOCITY AND MASS FLOW RATE
C
      GAMBP=1.35
      UPRM=158.1*SQRT((GAMBP-1.0)*HBP)
      IF(TIME.LT.0.0004) DOTMPM=4090.0*TIME
      IF(TIME.GE.0.0004) DOTMPM=2.66*EXP(-1200.0*TIME)
C
C
C
C     BPLEFT WILL BECOME TRUE IF THERE IS SOME BLACK POWDER LEFT AND
C     THEN PRIMER WILL BE CALLED AGAIN.
      BPLEFT = .FALSE.
C
      DO 60 J = 1,NGR
      DO 60 I = 1,NGX
          IF (PHIBP(I,J) .GE. 0.999) GO TO 55
          IF(TBP(I,J) .LT. TIGNBP) GO TO 55
          IF(BPRAD(I,J) .LT. 0.0001) GO TO 35
          BPLEFT = .TRUE.
      IF(PCH(I,J) .LT.  0.0) WRITE(6,7000)  PCH(I,J),I,J
```

```
 7000 FORMAT(1H ,*PCH = *,F10.0,2I2)
            R = AGENBP*PCH(I,J)**EXPBP + BGENBP
            BURNL = R*DELT
            VOLD = FORTPI*BPRAD(I,J)**3
            BPRAD(I,J) = BPRAD(I,J) - BURNL
      IF(BPRAD(I,J).GT.0.0) GO TO 40
            WRITE(6,2000) I,J,BPRAD(I,J)
 35         BPRAD(I,J) = 0.0
            PHIBP(I,J) = 1.0
            GO TO 55
C
 40         CONTINUE
            VNEW = FORTPI*BPRAD(I,J)**3
            DELTAV = VOLD - VNEW
            PNDV =  (1.0 - PHIBP(I,J))/VOLD
            TEMP = PNDV*DELTAV
            DOTMP(I,J) = TEMP*BPDENS
      DELX=DX
      IF(I.EQ.1) DELX=DELX/2.0
      CUMBP=CUMBP+TEMP*BPDENS*DELX*AREAAX
            PHIPTD(I,J) = PHIBP(I,J) + TEMP
            GO TO 60
C
 55         PHIPTD(I,J) = PHIBP(I,J)
 60   CONTINUE
C
      RETURN
C
 2000 FORMAT(//' RADIUS OF BLACK POWDER AT GRID', I3,I3,'  IS',F10.4)
      END
      SUBROUTINE PROPEL
C*    THIS SUBROUTINE PRODUCES THE ACTUAL MOVEMENT OF PROPELLANT
C
       COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
       COMMON/GRIDNX/DXPRIM
       COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
       COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ,XLBAR
       COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $    AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
       COMMON/CLOCK/TIME,DELT
       COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
       COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
       COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
       COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
       COMMON/NEWPHI/PHI02,IENDC2
       COMMON/SPLINT/WHOLEC,WHOLEB
       COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1    VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2    DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3    PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4    VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
       LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
       LOGICAL WHOLEC,WHOLEB
C
C     ARRAY UPBDT IS CLEARED IN MAIN AT EACH TIME INTERVAL
C     UPDATED VALUES OF POROSITY (FROM REGRES) WERE PUT INTO ARRAY PHIBG
C     IN PRPVEL.  UPDATED VALUES OF UPB (FROM PRPVEL) WERE  PUT INTO
C     ARRAY UPB IN PRPVEL.  UPDATED GRAIN DIMENSIONS (FROM REGRES) WERE
C     PUT INTO ARRAYS XL, DO, DI IN REGRES.
C*
C
C
C *** THERE WILL NOT BE ANY PROPELLANT IN THE GRIDS OCCUPIED BY THE BELL
C *** IGNITER TUBE, THEREFORE DO NOT DO PROPELLANT MOTION CALCULATIONS
C *** FOR THOSE GRIDS.
      DO 10 J=1,2
```

```
C
      DO 10 I = 1,NGX
      PHIBG(I,J) = PHIBTD(I,J)
      PHIBG2(I,J) = PHI2TD(I,J)
      XL2(I,J) = XL2TDT(I,J)
      DO2(I,J) = DO2TDT(I,J)
      IF(J.EQ.1) DO2(I,J)=BPRAD(I,J)
      IF(J.EQ.1)PHIBG2(I,J)=PHIPTD(I,J)
      PHIBP(I,J)=PHIPTD(I,J)
      DI2(I,J) = DI2TDT(I,J)
      XL(I,J) = XLTDT(I,J)
      DO(I,J) = DOTDT(I,J)
      DI(I,J) = DITDT(I,J)
C
C *** LOAD ALL PROPELLANT POROSITY INTO PHIBG(I,J)
      PHIBG(I,J) = PHIBG(I,J) + PHIBG2(I,J) - 1.0
 10   CONTINUE
C
      J=2
      XL(NGX+1,J)=XLB(2)
      DO(NGX+1,J)=DOB(2)
      DI(NGX+1,J)=DIB(2)
      XL2(NGX+1,J)=XLB2(2)
      DO2(NGX+1,J)=DOB2(2)
      DI2(NGX+1,J)=DIB2(2)
      PHIBG(NGX+1,J)=PHI(2)+PHI2(2)-1.0
      UPBDT(NGX+1,J)=UP(2)
      PHIBG2(NGX+1,J)=PHI2(2)
      J=1
      PHIBG(NGX+1,J)=PHIBG(NGX-1,J)
      PHIBG2(NGX+1,J)=PHIBG2(NGX-1,J)
      DO2(NGX+1,J)=DO2(NGX-1,J)
      UPBDT(NGX+1,J)=-UPBDT(NGX-1,J)
C
      IF(UPBDT(1,2) .LE. 0.0) UPBDT(1,2) = 0.0
C     SAVE DX
      DXTEMP=DX
C
      DO 80 J=1,2
      DO 80 I = 1,NGX
C     ADJUST DX WHEN I=NGX, NX=1
      IF(I.EQ.NGX.AND.NX.EQ.1) DX=DXPRIM-(DX/2.0)
      INCRE = 1
C     TEST VELOCITY IN ITH GRID TO DETERMINE INTO WHICH ADJACENT
C     GRID THE PROPELLANT WANTS TO FLOW.
      IF(UPBDT(I,J).LT.0.0)INCRE = -1
      IF(UPBDT(I,J).EQ.0.0)INCRE = 0
      ITWO = I + 2 * INCRE
      IONE = I + INCRE
C
```

```
      UPI = UPBDT(I,J)
      IF(J.NE.1) GO TO 15
      AREAX=AREAAX
      ARONE=AREAAX
      ARTWO=AREAAX
      AREAP1=AREAAX
      AREAM1=AREAAX
      GO TO 18
   15 CONTINUE
      AREAR(NGX+1)=BOREA
      AREAR(NGX+2)=BOREA
C
      IF(I .EQ. 1) GO TO 16
C
      AREAX = AREAR(I)
      ARONE = AREAR (IONE)
      ARTWO = AREAR (ITWO)
      AREAP1 = AREAR (I+1)
      AREAM1 = AREAR (I-1)
      GO TO 18
C
   16 AREAX = AREAR (I)
      ARONE= AREAR(IONE)
      ARTWO = AREAR (ITWO)
      AREAP1 = AREAR (I+1)
C
C
   18 CONTINUE
C     CALCULATE PARAMETERS FOR THE ITH GRID MASS BALANCE.
      DMPIP1 =0.0
      DMPI = ( 1.0 - PHIBB(I,J))*DELT*AREAX*ABS ( UPBDT(I,J))
      DMPIM1 = 0.0
      XLU = XL(I+1,J)
      XLU2 = XL2(I+1,J)
      XLL2 = XL2(I-1,J)
      DOU2 = DO2(I+1,J)
      DOL2 = DO2(I-1,J)
      DIU2 = DI2(I+1,J)
      DIL2 = DI2(I-1,J)
      XLL = XL(I-1,J)
      DOU = DO(I+1,J)
      DOL = DO(I-1,J)
      DIU = DI(I+1,J)
      DIL = DI(I-1,J)
      UPU = UPB(I+1,J)
      UPL = UPB(I-1,J)
      TZCU=TZC(I+1,J)
      TZCL=TZC(I-1,J)
C     DETERMINE IF THE ITH GRID LIES ON A BOUNDARY FOR
C     SPECIAL TREATMENT.
```

```
      IF(I.EQ.1) GO TO 40
      IF(UPBDT(I+1,J) .LT. 0.0) DMPIP1 = (1.0 - PHIBG(I+1,J))
     1  *DELT*AREAP1*ABS(UPBDT(I+1,J))
      IF(UPBDT(I-1,J) .GT. 0.0) DMPIM1 = (1.0 - PHIBG(I-1,J))
     1  *DELT*AREAM1*ABS(UPBDT(I-1,J))
      GO TO 50
 40   IF(I .EQ. 1  .AND.  UPBDT(2,J) .LT. 0.0) DMPIP1 = 2.0*
     1  (1.0 - PHIBG(2,J))*DELT*AREAP1*ABS(UPBDT(2,J))
      DMPI = 2.0 * DMPI
C     MASS BALANCE ON THE ITH GRID TO DETERMINE NEW VALUE
C     OF POROSITY
   50 PHII = PHIBG (I,J) + (DMPI - DMPIP1 - DMPIM1)/ (AREAX*DX)
C
C *** DETERMINE DMPIM,DMPIPM,DMPIMM,PHIIM AND DMPIS,DMPIPS,DMPIMS,PHIIS,
C     WHICH HAVE MEANINGS SIMILAR TO DMPI,DMPIP1,DMPIM1,PHII ONLY DEAL
C     WITH MP OR SP PROPELLANT ONLY.
C
C     EXTRACT POROSITY OF MP PROPELLANT FROM PHIBG.
      PHIIJ = PHIBG(I,J) - PHIBG2(I,J) + 1.0
C
      VOLI = AREAX*DX
C
C
C     DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID I.
C     LET THE POROSITY OF PROPELLANT MOVING OUT OF THE GRID BE PHIOUT.
      PHIOUT = 1.0 - DMPI/VOLI
      IF(UPBDT(I,J) .LE. 0.0) GO TO 54
C
C     SINCE PROPELLANT IS MOVING TO THE RIGHT, MP PROPELLANT MOVES
C     BEFORE SP.
      IF(PHIOUT .LT. PHIIJ) GO TO 52
C
C     ONLY MP MOVED.
      DMPIS = 0.0
      DMPIM = DMPI
      GO TO 58
C
C     ALL MP AND SOME SP MOVED.
 52   DMPIM = (1.0 - PHIIJ)*VOLI
      DMPIS = DMPI - DMPIM
      GO TO 58
C
C     SINCE PROPELLANT IS MOVING TO THE LEFT, SP PROPELLANT MOVES
C     BEFORE MP.
 54   CONTINUE
      IF(PHIOUT .LT. PHIBG2(I,J)) GO TO 56
C
C     ONLY SP MOVED.
      DMPIM = 0.0
      DMPIS = DMPI
      GO TO 58
```

```
C
C     ALL SP AND SOME MP MOVED.
 56   DMPIS = (1.0 - PHIBG2(I,J))*VOLI
      DMPIM = DMPI - DMPIS
C
C
C     DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID
C     I + 1 INTO GRID I.
C     DMPIP1 IS GREATER THAN 0.0 ONLY IF UPBDT(I+1,J) IS LESS THAN 0.0,
C     I. E. ONLY IF SP MOVES FIRST.
 58   CONTINUE
      IF(DMPIP1 .GT. 0.0) GO TO 60
      DMPIPM = 0.0
      DMPIPS = 0.0
      GO TO 64
C
 60   PHIOUT = 1.0 - DMPIP1/(AREAP1*DX)
      IF(PHIOUT .LT. PHIBG2(I+1,J)) GO TO 62
      DMPIPM = 0.0
      DMPIPS = DMPIP1
      GO TO 64
C
 62   DMPIPS = (1.0 - PHIBG2(I+1,J))*(AREAP1*DX)
      DMPIPM = DMPIP1 - DMPIPS
C
C
C     DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID
C     I - 1 INTO GRID I.
C     DMPIM1 IS GREATER THAN 0.0 ONLY IF UPBDT(I-1,J) IS GREATER THAN
C     0.0, I. E. ONLY IF MP MOVES FIRST.
 64   CONTINUE
      IF(DMPIM1 .GT. 0.0) GO TO 66
      DMPIMM = 0.0
      DMPIMS = 0.0
      GO TO 69
C
 66   PHIOUT = 1.0 - DMPIM1/(AREAM1*DX)
      IF(PHIOUT .LT. (PHIBG(I-1,J) - PHIBG2(I-1,J) + 1.0)) GO TO 68
      DMPIMS = 0.0
      DMPIMM = DMPIM1
      GO TO 69
C
 68   DMPIMM = (1.0 - (PHIBG(I-1,J) - PHIBG2(I-1,J) + 1.0))*(AREAM1*DX)
      DMPIMS = DMPIM1 - DMPIMM
C
 69   CONTINUE
      PHIIM = PHIIJ + (DMPIM - DMPIPM - DMPIMM)/VOLI
      PHIIS = PHIBG2(I,J) + (DMPIS - DMPIPS - DMPIMS)/VOLI
C
```

```
      IF(I .NE. 1) GO TO 75
      XLL = XLU
      DOL = DOU
      DIL = DIU
      UPL = 0.0
      TZCL=TZCU
C
C
 75   IF(PHII .GE. 0.99999) GO TO 78
      UPBDT(I,J) = ( UPBDT(I,J)*(1.0 - PHIBG(I,J))*AREAX*DX
     1         + UPU*DMPIP1 + UPL*DMPIM1 - UPBDT(I,J)*DMPI )/
     2         ((1.0 - PHII)*AREAX*DX)
      IF(TZC(I,J) .LT. 0.001) TZC(I,J) = TZCL
      IF(TZC(I,J).LT.TIGN) TZC(I,J)=(TZC(I,J)*(1.0-PHIBG(I,J))
     .                          *AREAX*DX + TZCU*DMPIP1 + TZCL*DMPIM1
     *                                -TZC(I,J)*DMPI)/((1.0-PHII)*AREAX*DX)
C
      IF(PHIIS .GE. 0.99999) GO TO 76
      DENOM = (1.0 - PHIIS)*VOLI
      XL2TDT(I,J) = ( XL2(I,J)*(1.0 - PHIBG2(I,J))*VOLI + XLU2*DMPIFS +
     $  XLL2*DMPIMS - XL2(I,J)*DMPIS )/DENOM
      DO2TDT(I,J) = ( DO2(I,J)*(1.0 - PHIBG2(I,J))*VOLI + DOU2*DMPIFS +
     $  DOL2*DMPIMS - DO2(I,J)*DMPIS )/DENOM
      DI2TDT(I,J) = ( DI2(I,J)*(1.0 - PHIBG2(I,J))*VOLI + DIU2*DMPIFS +
     $  DIL2*DMPIMS - DI2(I,J)*DMPIS )/DENOM
C
 76   CONTINUE
      IF(PHIIM .GE. 0.99999) GO TO 78
      DENOM = (1.0 - PHIIM)*VOLI
      XLTDT(I,J) = ( XL(I,J)*(1.0 - PHIIJ)*VOLI + XLU*DMPIPM +
     $  XLL*DMPIMM - XL(I,J)*DMPIM )/DENOM
      IF(WHOLEC) DOTDT(I,J) = ( DO(I,J)*(1.0 - PHIIJ)*VOLI +
     $  DOU*DMPIPM + DOL*DMPIMM - DO(I,J)*DMPIM )/DENOM
      IF(WHOLEC) DITDT(I,J) = ( DI(I,J)*(1.0 - PHIIJ)*VOLI +
     $  DIU*DMPIPM + DIL*DMPIMM - DI(I,J)*DMPIM )/DENOM
C
 78   CONTINUE
      PHIBTD(I,J) = PHIIM
      PHI2TD(I,J) = PHIIS
C     RESTORE DX AFTER EACH ITERATION
      DX=DXTEMP
C
 80   CONTINUE
C
C
      DO 100 I = 1,NGX
      PHIBG(I,2)=PHIBG(I,2)-PHIBG2(I,2)+1.0
      BPRAD(I,1)=DO2TDT(I,1)
      DO2TDT(I,1)=0.0
      PHIPTD(I,1)=PHI2TD(I,1)
      PHIBG(I,1)=1.0
      PHIBTD(I,1)=1.0
      PHI2TD(I,1)=1.0
       PHIBG2(I,1)=1.0
      DO2(I,1)=0.0
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE PROPMO
C
C     SUBROUTINE PROPMO CALCULATES PROPELLANT MOTION IN THE BARREL.
C
      COMMON/CLOCK/TIME,DELT
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ,XLBAR
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/SPLINT/WHOLEC,WHOLEB
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60.5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL WHOLEC,WHOLEB
      DATA PI, PIDF /3.141593, 0.785398/
C
C *** LOAD BARREL GRID NO. 1 WITH QUANTITIES FROM CHAMBER GRID NGX
C
      PHI(1)=1.0-(AREAR(NGX)/BOREA)*(1.0-PHIBG(NGX,2))
      PHI2(1)=1.0-(AREAR(NGX)/BOREA)*(1.0-PHIBG2(NGX,2))
      XLB(1) = XL(NGX,2)
      DOB(1) = DO(NGX,2)
      DIB(1) = DI(NGX,2)
      XLB2(1) = XL2(NGX,2)
      DOB2(1) = DO2(NGX,2)
      DIB2(1) = DI2(NGX,2)
      UP(1) = UPB(NGX,2)
       UUP(1)=UP(1)
C
```

```
C *** CLEAR POROSITY AND PROP. DIMENSIONS IF THERE IS NONE IN THE GRID
      IF(PHI(1).LE.0.99999) GO TO 1
      PHI(1) = 1.0
      XLB(1) = 0.0
      DOB(1) = 0.0
      DIB(1) = 0.0
 1    CONTINUE
      IF(PHI2(1).LE.0.99999) GO TO 2
      PHI2(1) = 1.0
      XLB2(1) = 0.0
      DOB2(1) = 0.0
      DIB2(1) = 0.0
 2    CONTINUE
C *** PUT UPDATED POROSITY AND GRAIN DIMENSIONS CALCULATED IN DIMIN
C     INTO ARRAYS PHI, DOB, ETC.
      DO 3 I = 2,NX
      PHI(I) = UPHI(1)
      DOB(I) = UDOB(I)
      DIB(I) = UDIB(I)
      XLB(I) = UXLB(1)
      PHI2(I) = UPHI2(I)
      DOB2(I) = UDOB2(I)
      DIB2(I) = UDIB2(I)
      XLB2(I) = UXLB2(I)
 3    CONTINUE
      DO 10 I=2,NX
      PHI(I) = UPHI(I)
      PHI2(I) = UPHI2(I)
      XLB(I) = UXLB(I)
      DOB(I) = UDOB(1)
      DIB(I) = UDIB(I)
      XLB2(I) = UXLB2(I)
      DOB2(1) = UDOB2(I)
      DIB2(1) = UDIB2(I)
C *** LOAD ALL PROPELLANT POROSITY INTO PHI(I)
      PHI(I) = PHI(I) + PHI2(I) - 1.0
 10   CONTINUE
      PHI(1)=PHI(1)+PHI2(1)-1.0
C
C *** UPDATE PROPELLANT VELOCITY. PUT UPDATED VALUES IN ARRAY UUP
C *** USE AN AVERAGE POROSITY IN UPDATING UP.
C *** GTRHOP 1S GRAV * DELT / RHOP
C
      DO 15 I = 2,NX
      PHIAVE = (PHI(I-1) + PHI(I) + PHI(I) + PHI(I+1)) * 0.25
      IF(I.EQ.NX) PHIAVE = (PHI(1-1) + PHI(I)) * 0.5
      IF(PHIAVE .GE. 0.99999) GO TO 15
      DELUP = GTRHOP * UDRAG(I)/(1.0 - PHIAVE)
      UUP(I) = UP(I) + DELUP
 15   CONTINUE
```

```
      IF(UUP(NX).GT.VP)UUP(I) = VP
      DO 80 I=2,NX
      INCRE = 1
C *** TEST VELOCITY IN ITH GRID TO DETERMINE INTO WHICH ADJACENT
C *** GRID THE PROPELLANT WANTS TO FLOW.
      IF(UUP(I).LT.0.0) INCRE = -1
      IF(UUP(I).EQ.0.0) INCRE = 0
      ITWO = I + 2*INCRE
      IONE = I + INCRE
      UPI = UUP(I)
      AREAX = BOREA
      ARONE = BOREA
      ARTWO = BOREA
      IF(ITWO.LT.1) ARTWO = AREAR(NGX-1)
      IF(IONE.EQ.1) ARONE = AREAR(NGX)
      AREAP1 = BOREA
      AREAM1 = BOREA
      IF(I-1.EQ.1) AREAM1 = AREAR(NGX)
C *** CALCULATE THE ADJACENT GRID MASS BALANCE PARAMETERS
      DMPI1 = (1.0 - PHI(IONE)) * DELT * ARONE *ABS(UUP(IONE))
      DMPI = (1.0 - PHI(I)) * DELT * AREAX * ABS(UUP(I))
      DMPI2 = 0.0
C *** DETERMINE IF THE ADJACENT GRID LIES ON A BOUNDARY
      IF(IONE.EQ.NX+1) GO TO 25
      IF(FLOAT(INCRE)* UUP(ITWO).LE.0.0) DMPI2 = (1.0-PHI(ITWO))
     1  *ABS(UUP(ITWO)) * DELT* ARTWO
C *** PERFORM MASS BALANCE ON I+1 * INCRE GRID
      PHII1 = PHI(IONE) + (DMPI1 - DMPI2 - DMPI) / (ARONE * DX)
 25   IF(ABS(UPI).LT.1.0E-7) UPI=0.0
C *** CALCULATE PARAMETERS FOR THE ITH GRID MASS BALANCE.
      DMPIP1 = 0.0
      DMPI = (1.0-PHI(I)) * DELT * AREAX * ABS(UUP(I))
      DMPIM1 = 0.0
      XLU = XLB(I+1)
      XLU2 = XLB2(I+1)
      XLL2 = XLB2(I-1)
      DOU2 = DOB2(I+1)
      DOL2 = DOB2(I-1)
      DIU2 = DIB2(I+1)
      DIL2 = DIB2(I-1)
      XLL = XLB(I-1)
      DOU = DOB(I+1)
      DOL = DOB(I-1)
      DIU = DIB(I+1)
      DIL = DIB(I-1)
      UPU = UP(I+1)
      UPL = UP(I-1)
      IF(I.EQ.NX) GO TO 40
      IF(UUP(I+1).LT.0.0)DMPIP1 =(1.0 - PHI(I+1))
     1  * DELT * AREAP1 * ABS(UUP(I+1))
      IF(UUP(I-1).GT.0.0)DMPIM1 =(1.0 - PHI(I-1))
     1 * DELT * AREAM1 * ABS(UUP(I-1))
      GO TO 50
 40   IF(I.EQ.NX .AND. UUP(NX-1) .GT.0.0) DMPIM1 = 2.0 *
     1 (1.0-PHI1(NX-1)) * DELT * AREAM1 * ABS(UUP(NX-1))
      DMPI = 2.0* DMPI
C *** MASS BALANCE ON THE ITH GRID TO DETERMINE NEW VALUE
C *** OF POROSITY
 50   PHII = PHI(I) + (DMPI - DMPIP1 - DMPIM1)/(AREAX * DX)
C *** DETERMINE DMPIM, DMPIPM, DMPIMM, PHIIM, AND DMPIS, DMPIPS,
C *** DMPIMS, PHIIS, WHICH HAVE MEANINGS SIMILAR TO DMPI, DMPIP1,
C *** DMPIM1,PHI1, ONLY DEAL WITH MP OR SP PROPELLANT ONLY.
C ***
C *** EXTRACT POROSITY OF MP PROPELLANT FROM PHI
      PHIIJ = PHI(I) - PHI2(I) + 1.0
      VOLI = AREAX * DX
C
C
C *** DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID I.
C *** LET THE POROSITY OF PROPELLANT MOVING OUT OF THE GRID BE PHIOUT.
      PHIOUT = 1.0 - DMPI/VOLI
      IF(UUP(I).LE.0.0) GO TO 54
C
C *** SINCE PROPELLANT IS MOVING TO THE RIGHT, MP PROPELLANT MOVES
C *** BEFORE SP.
      IF(PHIOUT.LT.PHIIJ) GO TO 52
C
C *** ONLY MP MOVED.
      DMPIS = 0.0
      DMPIM = DMPI
      GO TO 58
C
C *** ALL MP AND SOME SP MOVED.
 52   DMPIM = (1.0 - PHIIJ)*VOLI
      DMPIS = DMPI - DMPIM
      GO TO 58
C
C *** SINCE PROPELLANT IS MOVING TO THE LEFT, SP PROPELLANT MOVES
C *** BEFORE MP.
 54   CONTINUE
      IF(PHIOUT .LT. PHI2(I)) GO TO 56
C
C *** ONLY SP MOVED
      DMPIM = 0.0
      DMPIS = DMPI
      GO TO 58
C
C *** ALL SP AND SOME MP MOVED.
 56   DMPIS = (1.0 - PHI2(I))* VOLI
      DMPIM = DMPI - DMPIS
C
C
```

```
C *** DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID
C *** I+1 INTO GRID I.
C *** DMPIP1 IS GREATER THAN 0.0 ONLY IF UUP(I+1) IS LESS THAN 0.0,
C *** I. E. ONLY IF SP MOVES FIRST.
C
 58   CONTINUE
      IF(DMPIP1 .GT. 0.0) GO TO 60
      DMPIPM = 0.0
      DMPIPS = 0.0
      GO TO 64
C
 60   PHIOUT = 1.0 - DMPIP1/(AREAP1 * DX)
      IF(PHIOUT .LT. PHI2(I+1)) GO TO 62
      DMPIPM = 0.0
      DMPIPS = DMPIP1
      GO TO 64
C
 62   DMPIPS = (1.0 - PHI2(I+1)) * AREAP1 * DX
      DMPIPM = DMPIP1 - DMPIPS
C
C *** DETERMINE THE MASSES OF MP AND SP PROPELLANT FLOWING FROM GRID
C *** I-1 INTO GRID I.
C *** DMPIM1 IS GREATER THAN 0.0 ONLY IF UUP(I-1) IS GREATER THAN
C *** 0.0, I. E. ONLY IF MP MOVES FIRST.
C
 64   CONTINUE
      IF(DMPIM1 .GT. 0.0) GO TO 66
      DMPIMM = 0.0
      DMPIMS = 0.0
      GO TO 69
C
 66   PHIOUT = 1.0 - DMPIM1/(AREAM1 * DX)
      IF(PHIOUT .LT. (PHI(I-1) - PHI2(I-1) + 1.0)) GO TO 68
      DMPIMS = 0.0
      DMPIMM = DMPIM1
      GO TO 69
C
  68  DMPIMM=(1.0-( PHI(I-1) - PHI2(I-1)+1.0))*AREAM1*DX
      DMPIMS = DMPIM1 - DMPIMM
 69   CONTINUE
      PHIIM=PHIIJ+(DMPIM-DMPIPM-DMPIMM)/VOLI
      PHIIS = PHI2(I) + (DMPIS - DMPIPS - DMPIMS)/VOLI
C
      IF(I.NE.NX) GO TO 75
      XLU = XLL
      DOU = DOL
      DIU = DIL
      UPU = 0.0
      XL2U = XL2L
      DO2U = DO2L
      DI2U = DI2L
C
```

```
C
 75   IF(PHII .GE.0.99999) GO TO 78
      UUP(I) = (UUP(I) * (1.0 - PHI(I)) * AREAX * DX
     1    + UPU * DMPIP1 + UPL * DMPIM1 - UUP(I) * DMPI)/
     2    ((1.0 - PHII) * AREAX * DX)
C
      IF(PHIIS .GE.0.99999) GO TO 76
      DENOM = (1.0 - PHIIS)*VOLI
      UXLB2(I) = (XLB2(I) * (1.0 - PHI2(I)) * VOLI + XLU2 * DMPIPS +
     1   XLL2 * DMPIMS - XLB2(I) * DMPIS)/DENOM
      UDOB2(I) = (DOB2(I) * (1.0 - PHI2(I)) * VOLI + DOU2 * DMPIPS +
     1DOL2 * DMPIMS - DOB2(I) * DMPIS )/DENOM
      UDIB2(I) = (DIB2(I) * (1.0 - PHI2(I)) * VOLI + DIU2 * DMPIPS +
     1  DIL2 * DMPIMS - DIB2(I) * DMPIS) /DENOM
C
 76   CONTINUE
      IF(PHIIM .GE. 0.99999) GO TO 78
      DENOM = (1.0 - PHIIM) * VOLI
      UXLB(I) = (XLB(I) * (1.0 - PHIIJ) * VOLI + XLU * DMPIPM +
     1   XLL * DMPIMM - XLB(I) * DMPIM )/DENOM
      IF(WHOLEC) UDOB(I) = (DOB(I) * (1.0 - PHIIJ) * VOLI +
     1  DOU * DMPIPM + DOL * DMPIMM - DOB(I) * DMPIM)/DENOM
      IF(WHOLEC) UDIB(I) = (DIB(I) * (1.0 - PHIIJ) * VOLI +
     1  DIU * DMPIPM + DIL * DMPIMM - DIB(I) * DMPIM)/DENOM
C
 78   CONTINUE
      UPHI(I) = PHIIM
      UPHI2(I) = PHIIS
C
 80   CONTINUE
C
C
      DO 100 I = 2,NX
      PHI(I) = PHI(I) - PHI2(I) + 1.0
 100  CONTINUE
C
      RETURN
      END
      SUBROUTINE PRPFIR
C
C     SUBROUTINE PRPFIR CALCULATES PROPELLANT HEAT TRANSFER AND
C     TEMPERATURE RISE LEADING TO IGNITION.
C     TZC(I,J) - SURFACE TEMPERATURE OF PROPELLANT NOT UNDER THE
C          INFLUENCE OF BELL TUBE HOLES AT GRID (I,J)
C     TZR(I) - SURFACE TEMPERATURE OF PROPELLANT UNDER THE INFLUENCE
C          OF BELL TUBE HOLES AT GRID (I,2)
C
C
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/CLOCK/TIME,DELT
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $  AH(60),FRACT(60)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/PROPO/ FIRE
      DIMENSION UHOLE(60)
      LOGICAL ROW2
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
      LOGICAL FIRE
      DATA FORTPI/12.566/
C      FIRE REMAINS TRU AFTER THE FIRST PROPELLANT GRID IGNITES
C
C     ARRAY QBAG IS CLEARED IN UPDATE AT EACH TIME INTERVAL.
C     QBAG IS USED IN CALCULATING UPDATED ENTHALPY IN THE PATH ROUTINES.
C     A TERM MAY HAVE ALREADY BEEN ADDED TO QBAG IN SUBROUTINE BPFIR.
C
```

```
C     IGNIT WILL REMAIN TRUE ONLY IF AT EACH GRID THERE IS NO
C     PROPELLANT OR THE PROPELLANT IS IGNITED.
       IGNIT = .TRUE.
       DO 100 J=1,NGR
           ROW2 = J .EQ. 2
       DO 97 I=1,NGX
               TTX = TBG(I,J)
               TEMP = TTX*SQRT(TTX)
C
C     GAS VISCOSITY
               TXMU = C1*TEMP/(TTX + C2)
C
C     THERMAL CONDUCTIVITY OF GAS
               TXK = C3*TEMP/(TTX + C4)
C
       DMAVG=0.0
        IF(PHIBG(I,J).GE.0.99999.AND.PHIBG2(I,J).GE.0.99999) GO TO 95
C
C      CALCULATE WEIGHTED AVERAGE OF DM AND DM2.
       DMAVG=(DM*(1.0-PHIBG(1,J))+DM2*(1.0-PHIBG2(1,J)))/
     $(2.0-PHIBG(1,J)-PHIBG2(I,J))
C      CALCULATE RISE IN TZC.
       IF(TZC(I,J).GE.TIGN) GO TO 95
C     REYNOLDS NUMBER
           TEMP1 = RHOBG(I,J)*DMAVG/TXMU
               RETX = TEMP1*ABS(UBG(I,J))
C
C     NUSSELT NUMBER
       TEMP2=0.000613*PCH(I,J)**0.556/TXK*DMAVG
        TEMP2=TEMP2*0.5
               IF(RETX .LE. 1.0E-10) TXNUS = TEMP2
               IF(RETX .GT. 1.0E-10) TXNUS = 0.3*RETX**0.62 + TEMP2
C
C     SURFACE AREA OF PROPELLANT IN GRID PER VOLUME INCREMENT
               IF(PHIBG(I,J) .LT. 0.99999) ADV = 4.0*(1.0 - PHIBG(I,J))*
     $           ( (DO(I,J) + FN*DI(I,J))/(DO(1,J)*DQ(I,J) - FN*
     1           DI(I,J)*DI(I,J)) + 0.5/XL(I,J) )
               IF(PHIBG(I,J) .GE. 0.99999) ADV = 0.0
               IF(PHIBG2(I,J) .LT. 0.99999) ADV2 =
     $            4.0*(1.0 - PHIBG2(I,J))*(1.0/(DO2(I,J) - DI2(I,J)) +
     1            0.5/XL2(I,J))
               IF(PHIBG2(I,J) .GE. 0.99999) ADV2 = 0.0
C
C     TAKE A WEIGHTED AVERAGE OF ADV AND ADV2.
               ADV = (ADV*(1.0 - PHIBG(I,J)) + ADV2*(1.0 - PHIBG2(I,J)))/
     $              (2.0 - PHIBG(I,J) - PHIBG2(I,J))
C
C     HEAT FLUX TO PROPELLANT--BTU/FT**2 - SEC
           TEMP3 = TXK/DMAVG
               QCONVA = TXNUS*TEMP3*(TTX - TZC(1,J))
               IF(QCONVA .LT. 0.001) GO TO 50
```

```
              TEMP = QCONVA*ADV
              IF(ROW2) TEMP = TEMP*FRACT(I)
              QBAG(I,J) = QBAG(I,J) + TEMP
      IF(TZC(I,J).GE.TIGN) GO TO 95
              TEMP = QCONS*QCONVA
              TEFF = ((TZC(I,J) - T0)/TEMP)**2
              IF(TEFF .GT. TIME) TEFF = TIME
C
C    HEAT TRANSFER CALCULATION USING SEMI-INFINITE HEAT CONDUCTION
C    EQUATION WITH AN EFFECTIVE TIME
              TZC(I,J) = TZC(I,J) + TEMP*
     $            (SQRT(TEFF + DELT) - SQRT(TEFF))
              IF(TZC(I,J) .LT. TIGN) GO TO 50
              FRACT(I) = 1.0
              FIRE = .TRUE.
              IF(IDEBUG(10)) WRITE(6,2000) TIME,I,J
C
C
C
C    CALCULATE RISE IN TZR
 50           CONTINUE
              IF(.NOT. ROW2) GO TO 95
              IF(FRACT(I) .GE. 0.9999) GO TO 95
              IF(TZR(I) .GE. TIGN) GO TO 95
              UHOLE(I) = DOTMIG(I)/(0.7*RHOBG(1,1)*AH(I))
C
C    RESULTANT VELOCITY
          UGAS = UBG(I,J)
          IF(I .EQ. 1) UGAS = UBG(I+1,J)/2.
          IF(I .EQ. NGX .AND. NX .EQ. 1) UGAS = UBG(I-1,J)/2. + UBG(I,J)
     •                                   /2.
          SQRTUH = SQRT(UGAS*UGAS + VBG(I,J)*VBG(I,J))
              RETXH = TEMP1*SQRTUH
              IF(RETXH .LE. 1.0E-10) TXNUSH = TEMP2
              IF(RETXH .GT. 1.0E-10) TXNUSH = 0.3*RETXH**0.62 + TEMP2
              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) - T0)/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
C     CALCULATE HEAT LOSS TO PRIMER TUBE AND CHAMBER OR CASE WALL.
C     SURFACE AREA PER UNIT VOLUME AND REPRESENTATIVE DIMENSION.
      PERGR=0.0
      IF(J.NE.1) GO TO 70
      IF(DMAVG.GT.0.0) PERGR=4.0*(1.0-PHIBP(I,J))*AREAAX/DMAVG
      DMAVG=4.0*PHIBP(I,J)*AREAAX/(PERGR+3.1416*DIAMBT)
      ADV=SQRT(FORTPI/AREAAX)
   70 CONTINUE
      IF(J.NE.2) GO TO 80
      IF(DMAVG.GT.0.0) PERGR=4.0*(1.0-PHIBG(I,J))*AREAR(I)/DMAVG
      DMAVG=4.0*PHIBG(I,J)*AREAR(I)/(PERGR+3.1416*(1.128*
     $SQRT(AREAR(I)+AREAAX)+DIAMBT))
      ADV=(SQRT(FORTPI*(AREAR(I)+AREAAX))+SQRT(FORTPI*AREAAX))/AREAR(I)
   80 CONTINUE
      RETX=RHOBG(I,J)*DMAVG/TXMU*ABS(UBG(I,J))
      TEMP2=TXK/DMAVG
      TEMP3=0.000613*PCH(I,J)**0.556/TEMP2
      TXNUS=TEMP3
       IF(RETX.GT.0.0) TXNUS=0.023*RETX**0.8+TEMP3
      QCONVA=TXNUS*TEMP2*(TTX-T0)
      QBAG(I,J)=QBAG(I,J)+QCONVA*ADV
   97 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 PRPVEL
      COMMON/PRIMV/BPDENS,BPRAD(60,5),ACEFIT,FGENCP,EXPBR
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGR,IENDR,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPCAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/FORCE/ PFORCE(60,5), PFORDT(60,5)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/NEWPHI/PHI02,IENDC2
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      DIMENSION PPROP(60,5)
      DIMENSION FCOMP(60)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      DATA GRAV/32.16/
      CALL CLEAR(PFORCE(1,1),PFORDT(60,5))
      CALL CLEAR(PPROP(1,1),PPROP(60,5))
      CALL CLEAR(FCOMP(1),FCOMP(60))
      DO 100 J=1,2
C*
C    SUBROUTINE PRPVEL CALCULATES UPDATED PROPELLANT VELOCITY
C    (ASSUMING PROPELLANT IS FREE TO MOVE)
C*
C *** FOR THIS SUBROUTINE PUT THE TOTAL POROSITY OF SP AND MP
C *** PROPELLANT INTO ARRAY PHIBG.
      DO 5 I = 1,NGX
      PHIBG(I,J)=PHIBG(I,J)+PHIBG2(I,J)+PHIBP(I,J)-2.0
      AREA=AREAR(I)
      IF(J.EQ.1)AREA=AREAAX
      COMP=(PHI0-PHIBG(I,J))/(1.0-PHI0)
      FCOMP(I)=0.0
      IF(COMP.GT.0.0)FCOMP(I)=2.446E06*COMP**1.224*AREA
 5    CONTINUE
C
C *** THERE IS NO SP OR MP PROPELLANT FOR J = 1.
C
C*
      DO 50 I=1,NGX
C
      PHIBG(I,J)=PHIBTD(I,J)+PHI2TD(I,J)+PHIPTD(I,J)-2.0
      IF(PHIBG(I,J).GE.0.99999) XDRAG(I,J)=0.0
      IF(PHIBG(I,J).GE.0.99999) GO TO 50
C*
C*   CALCULATION OF FORCE TRANSMISSION THROUGH A PACKED BED OF PROPELLAN
C    CALCULATE TOTAL PRESSURE GRADIENT
         IPA = IPATH(I,J)
      GO TO (10,10,20,15,20),IPA
   10    DPDX = (PCH(I-1,J) - PCH(I+1,J))/(2.0*DX)
      DFCOMP=(FCOMP(I-1) - FCOMP(I+1))/(2.0*DX)
         GO TO 25
C
  15  DPDX=(PCH(I-1,J)-PCH(I,J))/DX
      DFCOMP=(FCOMP(I-1)-FCOMP(I))/DX
      GO TO 25
C
   20    DPDX = (PCH(I,J) - PCH(I+1,J))/DX
      DFCOMP=(FCOMP(I)-FCOMP(I+1))/DX
 25      CONTINUE
      IF(PHIBG(I,J).GE.PHI0) DFCOMP=0.0
C
      AREA=AREAR(I)
      IF(J.EQ.1) AREA=AREAAX
      DFCOMP=DFCOMP/AREA
       PPROP(I,J)=XDRAG(I,J)+DPDX*(1.0-PHIBG(I,J))+DFCOMP
31    CONTINUE
C
      PRHO=RHOP
      IF(J.EQ.1)PRHO=BPDENS
      DELUP= PPROP(I,J)*GRAV*DELT/(PRHO*(1.0-PHIBG(I,J)) )
C
C
C    UPDATE UPB FOR USE IN PROPEL
      UPBOT(I,J)=UPB(I,J)+DELUP
C
   50 CONTINUE
C
 100  CONTINUE
C
      DO 120 J=1,2
      DO 120 I = 1,NGX
      PHIBG(I,J)=PHIBG(I,J)-PHI2TD(I,J)-PHIPTD(I,J)+2.0
      PHIBG2(I,J) = PHI2TD(I,J)
 120  CONTINUE
      RETURN
      END
      SUBROUTINE REGRES
      COMMON/BURN/ATPB,CT,PEXP
      COMMON/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMP,WMBP,
     $  GAM1B,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLF
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMB1,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),DI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/HOLEA/RADHOL(85),NROWH,NHOLES(85),XCL(85),AREAH(60),
     $  AH(60),FRACT(60)
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/SPLINT/WHOLEC,WHOLEB
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), QBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),        UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL WHOLEC,WHOLEB
      LOGICAL ROW2,PART
      DATA PIDF/.785398/
C
C     SUBROUTINE REGRES CALCULATES UPDATED GRAIN DIMENSIONS AND POROSITY
C     DUE TO BURNING OF M6 PROPELLANT.  PHIBTD IS USED IN THE PATH
C     ROUTINES AND UPDATED GRAIN DIMENSIONS ARE USED IN PROPEL.
C
C
C     ARRAY DOTMB IS CLEARED IN UPDATE AT EACH TIME INTERVAL
      DOTMH=0.0
      DOTMM=0.0
      DO 100 J=1,NGR
      ROW2 = J .EQ. 2
      DO 98 I=1,NGX
      IF(TZC(I,J).GE.TIGN)FRACT(I)=1.0
          IF(PHIBG(I,J) .GE. 0.99999  .AND.  PHIBG2(I,J) .GE. 0.99999)
     $         GO TO 90
          PART = ROW2 .AND. FRACT(I) .LE. 0.9999
C
```

```
C      IF PROPELLANT IN THE GRID IS NOT IGNITED, DO NOT DO THE BURN
C     CALCULATIONS
       IF(TZC(I,J).GE.TIGN) GO TO 10
           IF(PART .AND. TZR(I) .LT. TIGN) GO TO 90
           IF(.NOT. PART .AND. TZC(I,J) .LT. TIGN) GO TO 90
   10 CONTINUE
           R = ATPB*PCH(I,J)**PEXP + CT
           BURNL = R*TWODT
C
C
C     LOGIC FOR MULTI-PERFORATED PROPELLANT
           IF(PHIBG(I,J) .GE. 0.99999) GO TO 40
C     UPDATE GRAIN LENGTH.
           XLTDT(I,J) = XL(I,J) - BURNL
C
C     SEE IF GRAIN HAS SPLIT INTO SPLINTERS
C******NOTE THAT OLD DIMENSIONS ARE BEING TESTED.
           IF( DO(I,J) .LE. 3.0*DI(I,J) ) GO TO 20
C
C
C     UPDATE OTHER DIMENSIONS
           DOTDT(I,J) = DO(I,J) - BURNL
           DITDT(I,J) = DI(I,J) + BURNL
C
C     CALCULATE OLD AND NEW VOLUMES OF A GRAIN.
           VOLD = PIDF*XL(I,J)*( DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J) )
           VNEW = PIDF*XLTDT(I,J)*( DOTDT(I,J)*DOTDT(I,J) -
     $              FN*DITDT(I,J)*DITDT(I,J) )
           GO TO 30
C
C
C     CALCULATE OLD AND NEW GRAIN VOLUMES.
C     AFTER THE GRAIN HAS SPLINTERED, VALUES FOR THE CROSS-SECTIONAL
C     AREA OF THE PARTICLES GO INTO ARRAY DO AND VALUES FOR PERIMETER
C     GO INTO ARRAY DI.  IF THE GRAIN HAS JUST SPLINTERED, AREA AND
C     PERIMETER HAVE TO BE INITIALIZED.  IF THE GRAIN HAS JUST
C     SPLINTERED DO(I,J) IS APPROXIMATELY 3.*DI(I,J) AND IF NOT
C     DO(I,J) WILL BE LESS THAN DI(I,J).
 20        CONTINUE
           IF( DO(I,J) .LE. DI(I,J) ) GO TO 25
           WHOLEC = .FALSE.
           AREA = PIDF*(DO(I,J)*DO(I,J) - FN*DI(I,J)*DI(I,J))
           DI(I,J) = 3.14*(DO(I,J) + FN*DI(I,J))
           DO(I,J) = AREA
C
 25        DELR = BURNL*0.5
           DOTDT(I,J) = DO(I,J) - DI(I,J)*DELR
C
           IF(DOTDT(I,J) .GE. 1.0E-7) GO TO 27
           DOTDT(I,J) = 0.0
           DITDT(I,J) = 0.0
```

```
            XLTDT(I,J) = 0.0
            PHIBTD(I,J) = 1.0
            GO TO 50
C
 27      CONTINUE
C     ASSUME THE RATIO OF PERIMETER SQUARED TO CROSS-SECTIONAL AREA IS
C     CONSTANT
       DITDT(I,J)=SQRT(DI(I,J)*DI(I,J)/DO(I,J)*DOTDT(I,J))
C
C     VOLUME IS LENGTH TIMES CROSS-SECTIONAL AREA
            VOLD = XL(I,J)*DO(I,J)
            VNEW = XLTDT(I,J)*DOTDT(I,J)
C
C
 30         CONTINUE
            IF(VNEW .LE. 0.0) GO TO 40
C
            DELTAV = VOLD - VNEW
C
C     CALCULATE NUMBER OF GRAINS PER GRID/VOLUME OF GRID
            PNDV = (1.0 - PHIBG(I,J))/VOLD
            IF(PART) PNDV = PNDV*(1.0 - FRACT(I))
            TEMP = PNDV*DELTAV
C
C     CALCULATE GAS MASS GENERATED BY BURNING PROPELLANT/GRID VOLUME
            DOTMB(I,J) = TEMP*RHOP
       DELX=DX
       IF(I.EQ.1)DELX=DELX/2.0
       CUMMP=CUMMP+TEMP*RHOP*DELX*AREAR(I)
       DOTMH=DOTMH+DOTMB(I,J)*HMP1
       DOTMM=DOTMM+DOTMB(I,J)
C
C     UPDATE POROSITY
            PHIBTD(I,J) = PHIBG(I,J) + TEMP
            IF(PART) DOTDT(I,J) = DOTDT(I,J)*(1.0 - FRACT(I)) + DO(I,J)*
     $         FRACT(I)
            IF(PART) DITDT(I,J) = DITDT(I,J)*(1.0 - FRACT(I)) + DI(I,J)*
     $         FRACT(I)
            IF(PART) XLTDT(I,J) = XLTDT(I,J)*(1.0 - FRACT(I)) + XL(I,J)*
     $         FRACT(I)
            GO TO 50
C
 40         PHIBTD(I,J) = PHIBG(I,J)
C
C
C     LOGIC FOR SINGLY-PERFORATED PROPELLANT
 50         CONTINUE
            IF(PHIBG2(I,J) .GE. 0.99999) GO TO 95
        R=ATPB2*PCH(I,J)**PEXP2+CT2
       BURNL=R*TWODT
C
```

```
      HGS = HG(NX)
      PGS = PG(NX)
      UPS = UP(NX)
      RATIO = DX/DXPRIM
      RHOG(IP1) = RHOG(I) + RATIO*(RHOG(IP1) - RHOG(I))
      UG(IP1) = UG(I) + RATIO*(UG(IP1) - UG(I))
      HG(IP1) = HG(I) + RATIO*(HG(IP1) - HG(I))
      PG(IP1) = PG(I) + RATIO*(PG(IP1) - PG(I))
      UP(IP1) = UP(I) + RATIO*(UP(IP1) - UP(I))
C
C     AT GRID NX DXPRIM SHOULD BE USED RATHER THAN DX, SO T2DX MUST BE
C     CHANGED.
 30   CONTINUE
      IF(I .NE. NX) GO TO 40
      T2DXS = T2DX
      T2DX = DELT/(2.0*DXPRIM)
C
 40   CONTINUE
C
          F1 = PHI(IM1)
          F2 = PHI(IP1)
          F5 = PHI(I)
          G1 = RHOG(IM1)
          G2 = RHOG(IP1)
          G5 = RHOG(I)
          E1 = G1*UG(IM1)
          E2 = G2*UG(IP1)
          E5 = G5*UG(I)
      P1=PG(IM1)
      P5=PG(I)
      P2=PG(IP1)
      EI1=HG(IM1)-P1/G1/778.0
      EI2=HG(IP1)-P2/G2/778.0
      EI5=HG(I)-P5/G5/778.0
      C1=G1*EI1
      C2=G2*EI2
      C5=G5*EI5
      H1=F1*E1
       H2=F2*E2
C
          PHIAVE = (F1 + F5 + F5 + F2)*0.25
          RHOAVE = (G1 + G5 + G5 + G2)*0.25
          UGAVE = (UG(IM1) + UG(I) + UG(I) + UG(IP1))*0.25
      IF(UP(I).EQ.0.0.AND.PHIAVE.NE.0.)UP(I)=UGAVE
      UPAVE=(UP(IM1)+UP(I)+UP(I)+UP(IP1))*0.25
C
          UPHIT = UPHI(I)
C
          URHOT = ( F5*AMASS(I)*RHOAVE
     1          - T2DX*(F2*AMASS(IP1)*E2 - F1*AMASS(IM1)*E1)
     2          + PMDOT(I)*AMASS(I) )/(UPHIT*UAMASS(I))
```

```
C
         PHIRHO = UPHIT*URHOT
C
      GRAVA=GRAV*AMOM(I)*PHI(I)
         UUT = ( F5*AMOM(I)*(E1 + E5 + E5 + E2)*0.25
     1         - T2DX*(F2*AMOM(IP1)*E2*UG(IP1) - F1*AMOM(IM1)*E1*UG(IM1)
     2            + GRAVA*PG(IP1) - GRAVA*PG(IM1))
     3         - DELT*FRICT(I) + AMOM(I)*PMDOT(I)*UP(I) )/
     4         (PHIRHO*UAMOM(I))
C
         IF(ABS(UUT) .LT. 0.1) UUT = 0.0
      IF(PHIAVE.LT.0.999) CALL DRAG(UDRAG(I),.TRUE.,I,0)
C
      ETDT=(F5*AENER(I)*(C1+2.0*C5+C2)/4.0
     $-T2DX*(H2*AENER(IP1)*EI2+AENER(IP1)*P2/778.0*
     $(F2*UG(IP1)+(1.0-F2)* UP(IP1))-H1*AENER(IM1)*EI1
     $-AENER(IM1)*P1*(F1*UG(IM1)+(1.0-F1)* UP(IM1))/778.0)
     $-UDRAG(I)*UP(I)*DELT*AENER(I)/778.0-QCONV(I)
     $+PMDOT(I)*AENER(I)*(HMB+UP(I)**2/TWOGJ))/(PHIRHO*UAENER(I))
      CALL GSPROP(RO,RRO,R,CVO,CVH,CV,PN    ,ETDT,TDUM,URHOT,UUT,
     $0.0,GAM,CP,4)
      UHG(I)=ETDT+PN/URHOT/778.0
C
C
         URHOG(I) = URHOT
         UUG(I) = UUT
C
C     THE SAVED PROPERTIES AT GRID NX SHOULD BE PUT BACK INTO THE
C     APPROPRIATE ARRAYS BEFORE I IS SET TO NX.
         IF(I .NE. NX - 1) GO TO 100
      RHOG(NX) = RHOS
      UG(NX) = UGS
      HG(NX) = HGS
      PG(NX) = PGS
      UP(NX) = UPS
C
C
 100  CONTINUE
C
C     REPLACE T2DX BY ITS SAVED VALUE.
      T2DX = T2DXS
C
      RETURN
C
      END
      SUBROUTINE UPDATE
C
C    SUBROUTINE UPDATE UPDATES THE ARRAYS AT EACH TIME INTERVAL AND
C    PRINTS OUTPUT.
C
      COMMON/DXVALU/DXPS
      COMMON/GSTATE/A0,A1,A2,A3,A0SP,A1SP,A2SP,A3SP,
     $  A0MP,A1MP,A2MP,A3MP,A0BP,A1BP,A2BP,A3BP,WMSP,WMMP,WMBP,
     $  GAMIB,CUMSP,CUMMP,CUMBP,GAMSP,GAMMP,GAMBP,WMOLE
      COMMON/FAILED/THICKT,PHOOP,PCOMP,BTUB,XNTUB,FAIL,MFAIL(60),
     1THICK(60)
      COMMON/BARRL2/BOREA,XP,VP,BORED,BORER,BORED8,DT2BD,DTDSQ,XLBAR
      COMMON/CHAM/IX,IR,XB,RB,NGX,NGR,IBEGB,IENDB,IPATH(60,5),AREAG(5),
     $   AREACH,AREAC(60),IGNIT,ONED,DIAM1,DIAM2,DIS1,DIS2,DIS3,DIS4,
     $  AREAR(60),AREAAX,CHAM1,CHAM2,CHAM3,TOPGAP,AREAGP(60),DAVG,
     $  AREAH2,DIAMBT,BELEND,BELBEG,IPS1,IPS2,RADPS,BPIGN
      COMMON/CLOCK/TIME,DELT'
      COMMON/EQNS/DTDX,T2DR,T2DX,TWOTDR,DTDR,HMB,TWOGJ,DVAXIS,DVAXIT,
     $  DX,DR,NX,GJ,TWODT,HBP
      COMMON/GASCON/R0,RR0,CV0,CVH
      COMMON/GASES/R0M6,RR0M6,CV0M6,CVHM6,R0BP,RR0BP,CV0BP,CVHBP
      COMMON/GRAIN/ XL(60,5),  DO(60,5), DI(60,5), FN,
     1 XLTDT(60,5), DOTDT(60,5), DITDT(60,5), XL0,  DO0, DI0,
     3XLB(100),UXLB(100),XLB2(100),UXLB2(100),DOB(100),UDOB(100),
     $DOB2(100),UDOB2(100),DIB(100),UDIB(100),DIB2(100),UDIB2(100),CI02,
     3 DO02,XL02,XL2(60,5),DO2(60,5),DI2(60,5),XL2TDT(60,5),
     4  DO2TDT(60,5), DI2TDT(60,5), FN2
      COMMON/GRAIN2/HMB1, HMB2, ATPB2, CT2, RHOP2, PEXP2
      COMMON/GRIDNX/DXPRIM
      COMMON/INPUTS/C1,C2,C3,C4,T0,TIGN,QCONS,RHOP,PHI0,TF,CA,RHO0,
     $  H0,P0,U0,GTRHOP,HW,DM,DM2,TIGNBP,QBCONS,TOTM,DIFFPR
      COMMON/MOCON/CON3,CON4,CON5,AREAPB,Z0,WOB,XOB,FDMAX,PINER,
     $   CF,RADPB,PMASS,XINT,PINT,XLO,PLO,CON6
      COMMON/P/IPRINT,MODCH,MODGR,PRI1,IDEBUG(35)
      COMMON/PRIMV/BPDENS,BPRAD(60,5),AGENBP,BGENBP,EXPBP
      COMMON/BAG/PHIBG(60,5), RHOBG(60,5), HBG(60,5), UBG(60,5),
     1   VBG(60,5), UPB(60,5), PCH(60,5), TZC(60,5),
     2   DOTMIG(60), GBAG(60,5), XDRAG(60,5), DOTMB(60,5), UPBDT(60,5),
     3   PHIBTD(60,5), RHOBTD(60,5), HBGTD(60,5), UBGTD(60,5),
     4   VBGTD(60,5),TBG(60,5),DOTMBG(60),DOTMP(60,5),PHIBP(60,5),
     5 PHIPTD(60,5),TZR(60),TBP(60,5),PHI2TD(60,5),       UPB2(60,5),
     6 TZR2(60),TZC2(60,5),PHIBG2(60,5)
      COMMON/BARRL/ PHI(100), RHOG(100), HG(100), UG(100), UP(100),
     1  PG(100), TG(100), PMDOT(100), QL(100), UDRAG(100), FRICT(100),
     2  QCONV(100), UUP(100), UPHI(100), URHOG(100), UHG(100), UUG(100),
     3  AMASS(100), AMOM(100), AENER(100), UAMASS(100), UAMOM(100),
     4UAENER(100),PHI2(100),UPHI2(100)
      LOGICAL IGNIT,ONED,CHAM1,CHAM2,CHAM3,BPIGN
      LOGICAL PRI1,IDEBUG
      LOGICAL PR12,VAR
      LOGICAL FAIL
      DATA DOTMBS,DOTMPS,PMDOTS/0.0,0.0,0.0/
      DATA  I24/0/
      DATA PT1,PT2,PT3,PT4,PT5,PT6,PT7,PT8,PT9/0.0,0.0,0.0,0.0,0.0,
     *0.0,0.0,0.0,0.0/
C
C
C
C
C************************************************************************
C    UPDATE CHAMBER ARRAYS
C************************************************************************
C
      IF(IDEBUG(15) .AND. PRI1) WRITE(6,2003) IPRINT,TIME
C
C    IF THE POROSITY CONDITION IS NOT SATISFIED, VAR WILL BE SET FALSE.
      VAR = .TRUE.
C
      ITEST = 1
      J = 1
C
   9  DO 100 I = 1,NGX
      IF(I .GT. IENDB) J = 1
          PRI2 = .FALSE.
          IF(MOD(I,MODGR) .EQ. 0  .OR.  I .EQ. 1  .OR.  I .EQ. NGX)
     $        PRI2 = .TRUE.
C
      IF(I.NE.NGX) GO TO 51
      IF(NX.NE.1) GO TO 51
      DXFACT=(DXPS-DX*0.5)/(DXPRIM-DX*0.5)
      PHIBTD(I,J)=1.0-(1.0-PHIBTD(I,J))*DXFACT
      PHI2TD(I,J)=1.0-(1.0-PHI2TD(I,J))*DXFACT
  51  CONTINUE
C
              PHIBG(I,J) = PHIBTD(I,J)
          PHIBG2(I,J) = PHI2TD(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 GSPROP WITH HBG AND RHOBG
              CALL GSPROP(RO,RRO,R,CVO,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)
          XL2(I,J) = XL2TDT(I,J)
          DO2(I,J) = DO2TDT(I,J)
          DI2(I,J) = DI2TDT(I,J)
```

```
C
      IF(PCH(I,J).LE.0.0.OR.HBG(I,J).LE.0.0.OR.RHOBG(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.
      IF(PCH(I,J) .GT. 0.0 .AND. HBG(I,J) .GT. 0.0 .AND. RHOBG( I,J) .GT
     .. 0.0) GO TO 11
      WRITE(6,2016)
      WRITE(6,2003) IPRINT,TIME
  11  CONTINUE
C
C ** PRINT IF NEGATIVE IS DETECTED
C
              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           QBAG(I,J),XDRAG(I,J),DOTMB(I,J),DOTMP(I,J),PHIBG2(I,J),
     3           TBP(I,J),XL(I,J),DO(I,J),DI(I,J),XL2(I,J),DO2(I,J),
     4  DI2(I,J), BPRAD(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
      IF (ITEST .EQ. 2) GO TO 101
      ITEST = 2
      J = 2
      GO TO 9
C
  101 CONTINUE
      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
C *** AT PRESENT, ONEDIM WILL NOT BE CALLED.
```

```
C
      GO 10 150
C ***
      IF(ONED) GO TO 150
      IF(.NOT. VAR) GO TO 150
      IF(.NOT. IGNIT) GO TO 150
          ONED = .TRUE.
          CALL ONEDIM
          IF(IDEBUG(17)) WRITE(6,2006) TIME
          IF(IDEBUG(17)) WRITE(6,2009) IPRINT
          IF(.NOT. IDEBUG(18)) GO 10 150
          WRITE(6,2007)
          DO 130 I=1,NGX
              PRES = PCH(I,1)/144.
              WRITE(6,2008) I,PHIBG(I,1),RHOBG(I,1),UBG(I,1),HBG(1,1),
     $         TBG(1,1),PRES,UPB(I,1),PHIBG2(I,1),XL(I,1),DO(1,1),
     $         DI(I,1),XL2(I,1),DO2(I,1),DI2(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),
     $        QL(I),UDRAG(I),PMDOT(I),AMASS(I),AMOM(I),AENER(I),
     $PHI2(I),XLB(I),DOB(I),DIB(I),XLB2(I),DOB2(I),DIB2(I)
 170  CONTINUE
C
      IF(NX .LT. 2) GO TO 220
      DO 200 I=2,NX
      CALL GSPROP(R0,RRO,R,CV0,CVH,CV,PU,UHG(I),TU,URHOG(I),
     *UUG(I),0.0,GAM,CP,2)
      IF(URHOG(I).LE.0.0.OR.UHG(I).LE.0.0.OR.PU.LE.0.0)
     *  WRITE(6,2005) I,PHI(I),RHOG(I),UG(I),HG(I),TG(I),PG(I),
     *UP(I),QL(I),UDRAG(I),PMDOT(I),AMASS(I),AMOM(I),AENER(I),
     *PHI2(I), XLB(I),DOB(I),DIB(I),XLB2(I),DOB2(1),DIB2(I)
          PHI(I) = UPHI(I)
      PHI2(I)=UPHI2(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)
       XLB2(I)=UXLB2(I)
      DOB2(I)=UDOB2(I)
      DIB2(I)=UDIB2(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 180
C
          IF(.NOT. PRI1) GO TO 200
      IF(.NOT. IDEBUG(19)) GO TO 200
  180 CONTINUE
      IF(PG(I) .GT. 0.0 .AND. HG(I) .GT. 0.0 .AND. RHOG(I) .GT. 0.0)
     .  GO TO 181
      WRITE(6,2017)
      WRITE(6,2019)
      WRITE(6,2018) I,NX,IPRINT,VP,XP,DX,DELT
      WRITE(6,2004)
  181 PRES = PG(I)/144.
          WRITE(6,2005) I,PHI(I),RHOG(I),UG(I),HG(I),TG(I),PRES,
     $       UP(I),QL(I),UDRAG(I),PMDOT(I),AMASS(I),AMOM(I),AENER(I),
     $PHI2(I),XLB(I),DOB(I),DIB(I),XLB2(I),DOB2(I),DIB2(I)
 200  CONTINUE
 220  CONTINUE
C
C***********************************************************************
C    PRINT OUT CERTAIN PRESSURES
C***********************************************************************
C
      IF(.NOT. PRI1) GO TO 240
      IF(.NOT. IDEBUG(22)) GO TO 240
      P1 = PCH(1,1)/144.
      WRITE(6,2014) P1
C
 240  CONTINUE
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) + PHIBG2(1,1) + PHIBP(1,1) - 2.0)*RHOBG(1,1)*
     1         VOL
      PROMAS=((1.0-PHIBG(1,1))*RHOP+(1.0-PHIBG2(1,1))*RHOP2+(1.0-
     1         PHIBP(1,1))*BPDENS)*VOL
      DO 260 I=2,NGX
      VOL=DVAXIS
      IF(I.EQ.NGX.AND.NX.EQ.1)VOL=(VOL*((DXPRIM-DX)/2.0)/DX)
      VOLBPD=VOL*BPDENS
          GASMAS = GASMAS + (PHIBG(I,1) + PHIBG2(I,1) + PHIBP(I,1) - 2.0
     1             )*RHOBG(I,1)*VOL
      PROMAS=PROMAS+((1.0-PHIBG(I,1))*RHOP+(1.0-PHIBG2(I,1))*RHOP2)*VOL+
     $        (1.0 - PHIBP(I,1))*VOLBPD
 260  CONTINUE
C
      VOL = AREAR(1)*DX*0.5
          GASMAS = GASMAS + (PHIBG(1,2) + PHIBG2(1,2) + PHIBP(1,2) - 2.0
     1             )*RHOBG(1,2)*VOL
      PROMAS=PROMAS+((1.0-PHIBG(1,2))*RHOP+(1.0-PHIBG2(1,2))*RHOP2+
     $    (1.0 - PHIBP(1,2))*BPDENS )*VOL
      DO 270 I=2,NGX
          VOL = AREAR(I)*DX
      IF(I.EQ.NGX.AND.NX.EQ.1)VOL=(VOL*((DXPRIM-DX)/2.0)/DX)
          GASMAS = GASMAS + (PHIBG(I,2) + PHIBG2(I,2) + PHIBP(I,2) - 2.0
     1             )*RHOBG(I,2)*VOL
      PROMAS=PROMAS+((1.0-PHIBG(I,2))*RHOP+(1.0-PHIBG2(I,2))*RHOP2+
     $        (1.0 - PHIBP(I,2))*BPDENS )*VOL
 270  CONTINUE
      IF(CHAM2) GO TO 320
C
C    CALCULATIONS WHEN CHAMBER IS ONE DIMENSIONAL
 300  CONTINUE
      VOL = AREAC(1)*DX*0.5
      GASMAS = (PHIBG(1,1) + PHIBG2(1,1) + PHIBP(1,1) - 2.0)*RHOBG(1,1)
     1         *VOL
      PROMAS=((1.0-PHIBG(1,1))*RHOP+(1.-PHIBG2(1,1))*RHOP2+(1.-PHIBP(1,1
     1         ))*BPDENS)*VOL
      DO 310 I=2,NGX
          VOL = AREAC(I)*DX
      IF(I.EQ.NGX.AND.NX.EQ.1)VOL=(VOL*((DXPRIM-DX)/2.0)/DX)
          GASMAS = GASMAS + (PHIBG(I,1) + PHIBG2(I,1) + PHIBP(I,1) - 2.0
     1             )*RHOBG(I,1)*VOL
      PROMAS=PROMAS+((1.-PHIBG(I,1))*RHOP+(1.-PHIBG2(I,1))*RHOP2+
     $        (1.0 - PHIBP(I,1))*BPDENS )*VOL
 310  CONTINUE
C
C    BARREL CALCULATIONS
 320  CONTINUE
      IF(NX .EQ. 1) GO TO 360
      VOL = BOREA*DX
      VOLROP = VOL*RHOP
      IF(NX .EQ. 2) GO TO 340
      NX1 = NX - 1
      DO 330 I=2,NX1
      PHI(I)=PHI(I)+PHI2(I)-1.0
          GASMAS = GASMAS + PHI(I)*RHOG(I)*VOL
          PROMAS = PROMAS + (1.0 - PHI(I))*VOLROP
      PHI(I)=PHI(I)-PHI2(I)+1.0
 330  CONTINUE
C
 340  CONTINUE
      VOL = BOREA*(DXPRIM - 0.5*DX)
      VOLROP = VOL*RHOP
       PHI(NX)=PHI(NX)+PHI2(NX)-1.0
      GASMAS = GASMAS + PHI(NX)*RHOG(NX)*VOL
      PROMAS = PROMAS + (1.0 - PHI(NX))*VOLROP
      PHI(NX)=PHI(NX)-PHI2(NX)+1.0
C
 360  IF(PRI1 .AND. IDEBUG(20)) WRITE(6,2011) GASMAS,PROMAS
C
C    ADJUST DENSITIES
      ACTGAS = TOTM - PROMAS
      ADJUST = ACTGAS/GASMAS
      DO 370 J=1,NGR
      DO 370 I=1,NGX
          RHOBG(I,J) = RHOBG(I,J)*ADJUST
 370  CONTINUE
      DO 380 I=2,NX
          RHOG(I) = RHOG(I)*ADJUST
 380  CONTINUE
C
      IF(PRI1 .AND. IDEBUG(20)) WRITE(6,2013) ACTGAS,ADJUST
C
C
C***********************************************************************
C    KEEP A RUNNING SUM OF DOTMB,DOTMP, AND FMDOT TERMS AND
C    DETERMINE GAS CONSTANTS ON THE BASIS OF THESE
C***********************************************************************
C
C
      SUMB=CUMMP+CUMSP+CUMBP
      IF(SUMB.LE.0.0) GO TO 480
      FRSP=CUMSP/SUMB
      FRMP=CUMMP/SUMB
      FRBP=CUMBP/SUMB
      A0=FRSP*A0SP+FRMP*A0MP+FRBP*A0BP
      A1=FRSP*A1SP+FRMP*A1MP+FRBP*A1BP
      A2=FRSP*A2SP+FRMP*A2MP+FRBP*A2BP
      A3=FRSP*A3SP+FRMP*A3MP+FRBP*A3BP
      WMOLE=FRSP*WMSP+FRMP*WMMP+FRBP*WMBP
      GAMIB=FRSP*GAMSP+FRMP*GAMMP+FRBP*GAMBP
  480 CONTINUE
      IF(PRI1.AND.IDEBUG(21)) WRITE(6,2012) CUMSP,CUMMP,CUMBP,
     $  A0,A1,A2,A3,GAMIB,WMOLE
C***********************************************************************
C     GET READY FOR THE NEXT TIME INTERVAL
C***********************************************************************
C
 500  CONTINUE
C
C    CLEAR ARRAYS DOTMIG AND DOTMBG FOR SUBROUTINE MFLOW, ARRAY DOTMF
C    FOR PRIMER, ARRAY QBAG FOR PRPFIR, ARRAY DOTMB FOR REGRES, AND
C    ARRAY UPBDT FOR PRPVEL.
      CALL CLEAR( DOTMIG(1),QBAG( 60,5))
      CALL CLEAR( DOTMB(1,1),UPBDT( 60,5))
      CALL CLEAR(DOTMBG(1),DOTMP(60,5))
C
C
C    CLEAR ARRAY PMDOT FOR SUBROUTINE DIMIN, ARRAY QL FOR BNDLYR,
C    ARRAY UDRAG FOR RHOUH, AND ARRAY UUP FOR PROPMO.
      CALL CLEAR(PMDOT(1), UUP(100))
      P11=PCH(1,1)/144.0
      P12=PCH(1,2)/144.0
      PN1=PCH(NGX,1)/144.0
      PN2=PCH(NGX,2)/144.0
      PGN=PG(NX)/144.0
      PT1=PT2
      PT2=PT3
      PT3=PT4
      PT4=PT5
      PT5=PT6
      PT6=PT7
      PT7=PT8
      PT8=PT9
       PT9=P12
      IF(PT5.LT.1000.0) GO TO 737
      IF(PT5.GT.PT1.AND.PT5.GT.PT9.AND.IDEBUG(34)) WRITE(6,2015)IPRINT,
     *TIME,PT5
  737 CONTINUE
      XPPP = (XP - XOB)*12.0
      I24=I24+1
      II24=MOD(I24,10)
      IF(IDEBUG(24).AND.II24.EQ.0)WRITE(6,2015)IPRINT,TIME,P11,
     1  P12,PN1,PN2,PGN,XPPP,VP
C
      RETURN
```

```
C
C
C
C***********************************************************************
C
 2001 FORMAT(/,2I5,7(2X,E14.8),/,10X,7(2X,E14.8),/,10X,7(2X,E14.8),/,12X
     1    , E14.8,2X,E14.8)
 2003 FORMAT(//,' IPRINT = ',I6,
     $          /,' TIME =',E14.8,//,3X,'I',4X,'J',6X,'PHIBG',11X,
     1   'RHOBG',12X,'UBG',13X,'VBG',13X,'HBG',13X,'TBG',13X,'PCH',
     2   /,15X,'PHIBP',11X,'UPB',13X,'TZC',13X,'QBAG',11X,'XDRAG',11X,
     3   *DOTMB*,11X,*DOTMP*,/,15X,*PHIBG2*,10X,*TBP * ,14X,*XL*,14X,*DO*
     4   ,14X,*DI*,13X,*XL2*,13X,*DO2*,/,16X,*DI2*)
 2004 FORMAT(///,3X,'I',8X,'PHI',13X,'RHOG',13X,'UG',14X,'HG',14X,
     1    'TG',14X,'PG',14X,'UP',/,29X,'GL',12X,'UDRAG',11X,'PMDOT',
     2      11X,'AMASS',12X,'AMOM',11X,'AENER',
     3/,10X,*PHI2*,12X,*XLB*,13X,*DOB*,13X,*DIB*,13X,*XLB2*,
     412X,*DOB2*,12X,*DIB2*)
 2005 FORMAT(//,I5,7(2X,E14.8),/,21X,6(2X,E14.8),/,5X,7(2X,E14.8))
 2006 FORMAT('1         AT TIME ',E13.7,' THE CHAMBER WAS MADE 1-DIMENSIONA
     $L')
 2007 FORMAT(////,3X,'I',6X,'PHIBG',11X,'RHOBG',12X,'UBG',13X,'HBG',
     $  13X,*TBG*,13X,*PCH*,13X,*UPB*,/,10X,*PHIBG2*,12X,*XL*,14X,*DO*,
     $  14X,*DI*,13X,*XL2*,13X,*DO2*,13X,*DI2*)
 2008 FORMAT(/I5,7(2X,E14.8),/,7X,E14.8,6(2X,E14.8))
 2009 FORMAT(//,' IPRINT = ',I5)
 2010 FORMAT('  LOGIC FOR SUMMING GAS MASS AND PROPELLANT MASS HAS NOT B
     $EEN WRITTEN FOR CHAM1 TRUE')
 2011 FORMAT('  THE MASS OF GAS IN THE SYSTEM IS',F10.4,//,
     $  '  THE MASS OF PROPELLANT AND BLACK POWDER IN THE SYSTEM IS',
     $  F10.4)
 2012 FORMAT(//,*  CUMSP = *,F10.4,*  CUMMP = *,F10.4,*  CUMBP = *,
     $F10.4,//,* FOR THE NEXT INTERVAL  A0 = *,E10.4,* A1 = *,
     $E10.4,* A2 = *,E10.4,* A3 = *, E10.4,* GAMIB = *,F10.4,
     $ * WMOLE = *,F10.4,/)
 2013 FORMAT(' ACTUAL GAS IN THE SYSTEM IS', F10.4,
     $  '  ADJUSTING FRACTION IS',F10.4,/)
 2014 FORMAT(1H0,*PRESSURE AT GRID (1,1) IS*, E20.10)
 2015 FORMAT(2X, I5, 3X, 8(E13.6,2X))
 2016 FORMAT(//,*1 NEGATIVE PRESSURE,ENTHALPY OR DENSITY DETECTED BY UPD
     .ATE IN THE CHAMBER*)
 2017 FORMAT(//,*1 NEGATIVE PRESSURE,ENTHALPY OR DENSITY DETECTED BY UPD
     .ATE IN THE BARREL*)
 2019 FORMAT(10X,*I*,9X,*NX*,4X,*IPRINT*,10X,*VP*,13X,
     *  *XP*,13X,*DX*,11X,*DELT*)
 2018 FORMAT(2X,3I10,4E15.6)
      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 primer tube 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 M2A2 105mm howitzer with the M67 propelling charge loaded with special lot PAD-PE-490-1-F, at Zone 7 with the M1 projectile. The computed peak pressure and muzzle velocity for these conditions are 46200 psi and 1603 ft/sec., respectively.

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

A0BP
A1BP
A2BP
A3BP

The co-volume curve fit coefficients for black powder, for the equation of state

$$p(\frac{1}{\rho} - \eta) = RT$$

where the co-volume η in units of in^3/lbm, is given by

$$\eta = A_0 + A_1\rho + A_2\rho^2 + A_3\rho^3$$

The values used for these terms are:

A0BP = 15.0 in^3/lbm

A1BP = 0.0

A2BP = 0.0

A3BP = 0.0

A0MP
A1MP
A2MP
A3MP

The co-volume curve fit coefficients for multiperf propellant for the equation of state given above. The values generated by the BLAKE code for special lot F are:

A0MP = 33.579 in^3/lbm

A1MP = -26.083

A2MP = 20.755

A3MP = -20.301

A0SP
A1SP
A2SP
A3SP

The co-volume curve fit coefficients for single perf propellant for the equation of state given above. The values generated by the BLAKE code for propellant lot 68108 are:

A0SP = 34.192 in^3/lbm

A1SP = -25.578

A2SP = 16.500

A3SP = -16.214

AGEN,
AGEN2

The "A" term in the propellant burning rate equation

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

In this expression, p is in psi, T_o in °R, and $\dot{r}$ is in in/sec. AGEN is input for the multi-perf reference propellant and AGEN2 for the single-perf reference propellant. Temperature dependence was not used for M1 propellant, and this term was set equal to zero.

AGENBP

Pressure coefficient for black powder burn rate. This term was also set equal to 0.744.

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 M1 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
BGEN2

The "B" term in the propellant burning rate equation (see AGEN). A value of 0.280 x 10^{-2} was generated by closed bomb tests for BGEN, the multiperf propellant, and 0.214 x 10^{-2} was assumed for BGEN2, the single perf propellant.

BGENBP

Constant value for black powder burn rate. This term was set equal to 0.0.

BØRED Average barrel diameter, taking lands and grooves of the rifling into account, = 4.168 in.

BPDENS Black powder granule density, = 1.75 gm/cc.

B0
B1
B2
B3
B4
Coefficients to regression equation that expresses bed density in terms of physical grain parameters,

$$\rho_b = B0 + B1 \cdot x_{10} + B2 \cdot D0 + B3\, d_o + B4 \cdot nperf$$

The values for these coefficients have yet to be evaluated and are set to zero, except B0, which is set equal to 45.0.

BPRADO Initial effective radius of black powder granules assuming a spherical configuration, = 0.03 in.

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

CGEN
CGEN2
The "C" term in the burning rate equation (see AGEN), set equal to zero for M1 propellant because of inadequate temperature dependency information.

CHAM2 Logical variable set .TRUE, when the chamber grid matrix consists of two parallel one-dimensional networks. Otherwise, it is set .FALSE. For 105mm howitzer this is normally set .TRUE.

CHAM3 Logical variable set .TRUE. when the chamber grid matrix consists of three parallel one-dimensional networks. Otherwise it is set . FALSE. Normally set FALSE for 105mm howitzer simulations.

CHWT2( ) Array of propellant charge weight for each of the 7 zones of the M67 charge, set equal to 0.5175, 0.0875, 0.168, 0.231, 0.323, 0.538, and 0.880.

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

C3
C4

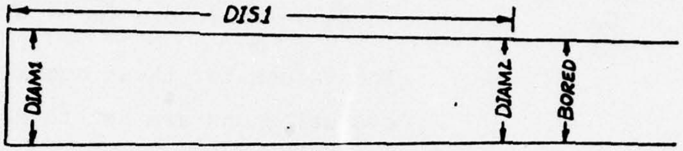
Constants in Sutherland's equation for thermal conductivity in Btu/ft-sec-°R. C3 = 0.291×10^{-6}; C4 = 170.1.

DIAMBT

Inside diameter of the igniter tube, equal to 0.387 in.

DIAM1
DIAM2
DIS1

Chamber dimensions as defined by the following sketch:



DIAM1 = 4.178

DIAM2 = 4.178

DIS1 = 11.16

DIO

Multi-perf propellant grain minor or perforation diameter, equal to 0.0220 in. for M1 propellant, lot F.

DIO2

Single-perf propellant grain minor or perforation diameter, equal to 0.0187 in. for M1 propellant lot 681-08.

DIFFPR

Differential pressure required to cause the primer tube liner to fail, assumed equal to 1000 psi.

DIR1

Perforation diameter for multi-perf reference propellant, assumed equal to 0.0220 in.

DIR2

Perforation diameter for single-perf reference propellant, assumed equal to 0.0198 in.

DØO

Multi-perf propellant grain major or exterior diameter, equal to 0.142 in. for M1 propellant lot F.

DØ02

Single-perf propellant grain major, or exterior diameter, equal to 0.0467 in. for M1 propellant lot 68-108.

DØR1

Exterior diameter for multi-perf reference propellant, assumed equal to 0.142 in.

DØR2

Exterior diameter for single-perf reference propellant, assumed equal to 0.0467 in.

EXPBP Black powder burn rate pressure exponent, = 0.24.

GAMBP
GAMMP
GAMSP
Values of I.B. GAMMA as generated by the BLAKE code for black powder, multiperf propellant, and single perf propellant, respectively. The value for black powder was estimated at 1.08 from closed bomb data while the BLAKE code generated values of 1.257 for the multiperf propellant lot F and 1.261 for the single-perf propellant lot 68-108.

HBP Heat of combustion of black powder, assumed equal to 1375 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 1500 Btu/lbm is good for the expected range of calculations.

HMB
HMB2
The energy added by burning propellant. HMB represents the multiperf propellant, 1586 Btu/lbm for lot F, and HMB2 represents the single perf propellant, 1531 Btu/lbm for lot 68-108. These values were generated by dividing the BLAKE code generated output parameters of impetus (ft-lbf/lbm) by [(I.B. GAMMA - 1.0) times 778 (ft-lbm/Btu]. The actual numbers were selected from equilibrium calculations for these propellants at 45000 psi, close to the peak pressure generated experimentally by lot F.

NGR Number of radial grids in the chamber matrix. This number must coincide with the matrix selected by CHAM2, or CHAM3. Currently, this number can be 3. Set at 2 or 3, with 2 being used normally.

NGX Number of axial grids in the chamber matrix. This number cannot exceed 59, which is one less than the number currently allocated the variables in common storage. 8 grids were used in the standard run.

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 M28 primer tube has 2 holes at each axial position and NHØLES is filled with 22 * 2, 63 * 0, to fill up the entire array of 85 potential rows.

NPERF Number of perforations in the multi-perf grains of the main charge. The M1 propellant used in the 105mm howitzer has 7 perforations.

NPERF2 Number of perforations in the single-perf grains, equal to 1.

NRØWH Number of rows of holes in the igniter tube, equal to 22 for the 105mm howitzer.

PEXP
PEXP2 Pressure exponent to the propellant burn rate equation (see AGEN) with PEXP equal to 0.715 for lot F as generated by a closed bomb test and PEXP2 assumed equal to 0.71 for lot 68-108.

PO Initial pressure, - 14.7 psi.

RADHØL Igniter tube hole radius array, equal to 0.1875 in. for the M28 primer tubes. The entire array is filled by 22 * 0.065, 63 * 0.0.

RF1
RF2 Relative force for multi-perf (1) and single-perf propellants (2), assumed equal to 1.0 for the standard run.

RHØP Grain density of M1 propellant where RHØP designates the Lot F multi-perf propellant equal to 97.7 lbm/ft^3 for lot 68-108 and RHØP2, single-perf, equal to 97.5 lbm/ft^3.

RHØP1R
RHØP2R Reference propellant density, RHØP1R representing multi-perf propellant and RHØP2R single-perf propellant, equal to 97.7 and 97.5 lbm/ft^3 respectively.

RQ1
RQ2 Relative quickness for multi-perf (1) and single-perf (2) propellants, assumed equal to 1.0.

TF Time extent of the calculation. This provides an alternate method to terminate a computer run. A value of 0.05 sec. is sufficient for most runs.

TIGN Propellant grain surface temperature at which ignition occurs, assumed ≈800°R for M1 propellant.

TIGNBP Ignition temperature for black powder, assumed equal to 1100°R.

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

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

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

WMBP
WMMP
WMSP Values for molecular weight of combusted black powder, multiperf propellant and single-perf propellant, respectively. The following values were assigned to these terms:

WMBP = 75 (estimated from closed bomb data)
WMMP = 22.33 (BLAKE code for lot F)
WMSP = 21.97 (BLAKE code for lot 68-108)

XCL Array that specifies the axial position of M28 primer 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 = 1.535, 21 * 0.375, 63 * 0.0 for the 105mm howitzer, with all distances given in inches.

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

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

XLBEL Length of the primer tube, equal to 9.64 in.

XLO Average initial multi-perf propellant grain length, equal to 0.318 in. for M1 propellant lot F.

XLO2 Average initial single-perf propellant grain length, equal to 0.199 in. for M1 propellant lot 68-108.

XLR1 Average initial multi-perf reference propellant grain length, equal to 0.318 in.

XLR2 Average initial single-perf reference propellant grain length, equal to 0.199 in.

BLACK POWDER INPUTS - NAMELIST BPINP

BEGTC Axial position of the beginning of the black powder primer charge, assumed equal to 0.0 in.

CHTC Black powder charge weight equal to 0.206 oz.

XLTC Length of the black powder primer charge, equal to 9.6 in.

BARREL INPUTS - NAMELIST BARINP

CF Coefficient of friction between projectile rotating band and the barrel in theory but acts as a coefficient that links projectile acceleration forces to barrel resistance. Assumed equal to zero.

PDMAX Equivalent reactive pressure experienced when the rotating band is finished being engraved, assumed equal to 0.0 psi (see sketch following page).

PINER Projectile moment of inertia - approximately equal to 0.0172 lb ft/sec^2-ft.

PINT Intermediate resistive pressure (see sketch on following page) assumed equal to 0.0 psi.

PLØ — Final resistive pressure (see sketch below) equal to 0.0 psi.

PMASS — Projectile mass, equal to 33 lbs. for the M1 projectile.

PZO — Initial or shot start pressure, assumed equal to 1000 psi (see sketch below).

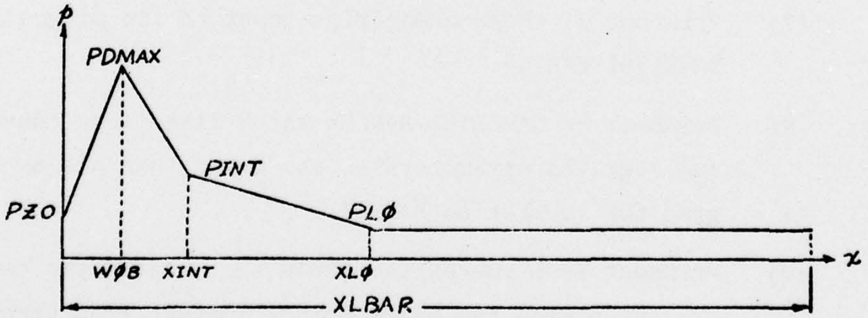
RADPB — Effective radius of the projectile cross-section upon which the pressure acts, equal to 2.089 in.

WØB — Width of the obturator, or rotating band, equal to 0.8 in. (see sketch below).

XINT — Distance from the origin of rifling at which PINT acts, assumed equal to 5.0 in. (see sketch below).

XLBAR — Total effective length of the barrel equal to 78 in. for the M2A2 105mm howitzer. This should be increased to 82 in. to account for the total travel during which the pressure force acts on the projectile.

XLØ — Distance from the origin of rifling at which PLØ acts, assumed equal to 20 in. (see sketch below).



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 and the print frequency of many items is controlled by input MØDCH in NAMELIST MØDS. 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 ntervals 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 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)
to
(29) Not used at present.

(30) Printout from subroutine BPINIT of certain parameters pertaining to initial black powder distribution calculations performed in that subroutine.

(31) Printout from CHSET that gives initial amounts of propellant and black powder in the system as calculated from porosity.

(32) Printout from DETPHI that gives charge lengths, first and last grids containing propellant and other information used in determining propellant porosities.

(33) Printout from MOTION that gives FDPRIM, the projectile resistive force, and the calculation interval energy 50th time interval.

(34) Printout of time and pressure (PCH(1,2)) in the region of the peak pressure.

(35) Not used at present.

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

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

FORMATS OF TYPICAL PRINTOUTS

```
IDEBUG(15)                                          CHAMBER OUTPUT
  IPRINT =       0
  TIME =0.

  I    I     PHIBG          PHOBG          UBG                VBG                HBG             TBG              PCH
             RHIBB          UBB            TSC                QBAG               XDRAG           DOTMB            DOTMP
             THP            XI             DO                 DI
   0.                .55833E-02

   1    1   .10000000E+01   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .44300000E+00   0.             0.                 0.               0.               0.               0.
            .53500000E+03   0.             0.                 0.
   0.                .55526E-02

   1    2   .73633442E+00   .61390653E-01  0.                 0.                .18574250E+03   .68214570E+03   .19227716E+02
            .97480948E+00   0.            .53500000E+03       0.               0.               0.              .10990958E-02
            .74500000E+03   .85333333E-01 .365833333E-01     .36666667E-02
   0.        0.

   1    3   .10000000E+01   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .10000000E+01   0.             0.                 0.               0.               0.               0.
         0.                 0.             0.                 0.
   0.                .55833E-02

   4    1   .10000000E+01   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .83562634E+00   0.             0.                 0.               0.               0.               0.
            .53500000E+03   0.             0.                 0.
   0.        0.

   4    2   .58096656E+00   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .10000000E+01   0.            .53500000E+03       0.               0.               0.               0.
         0.                 .85333333E-01 .365833333E-01     .36666667E-02
   0.        0.

   4    3   .10000000E+01   .59845591E-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   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .75135648E+00   0.             0.                 0.               0.               0.               0.
            .53500000E+03   0.             0.                 0.
   0.        0.

   8    2   .58096656E+00   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .10000000E+01   0.            .53500000E+03       0.               0.               0.               0.
         0.                 .85333333E-01 .365833333E-01     .36666667E-02
   0.        0.

   8    3   .10000000E+01   .59845591E-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   .59845591E-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   .58096656E+00   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .10000000E+01   0.            .53500000E+03       0.               0.               0.               0.
         0.                 .85333333E-01 .365833333E-01     .36666667E-02
   0.        0.

  12    3   .10000000E+01   .59845591E-01  0.                 0.                .14444781E+03   .53500000E+03   .14700000E+02
            .10000000E+01   0.             0.                 0.               0.               0.               0.
         0.                 0.             0.                 0.
   0.                .55833E-02
   16    1   .10000000E+01   .59845918E-01   0.              0.                 .14444781E+03   .53500000E+03   .14700000E+02
             .98157986E+00   0.              0.              0.               0.              0.              0.
             .53500000E+03   0.              0.              0.
     0.            0.

   16    2   .59019420E+00   .59845918E-01   0.              0.                 .14444781E+03   .53500000E+03   .14700000E+02
             .10000000E+01   0.              .53500000E+03   0.               0.              0.              0.
           0.                .87233333E-01   .36583333E-01   .36666667E-02
     0.            0.

   16    3   .10000000E+01   .59845918E-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   .59845918E-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   .59845918E-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   .59845918E-01   0.              0.                 .14444781E+03   .53500000E+03   .14700000E+02
             .10000000E+01   0.              0.              0.               0.              0.              0.
           0.                0.              0.              0.
IDEBUG(16)
  $DOT

  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,

  DOTMBG  =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
             0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
             0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
             0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,

  TZP     =  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
  --                             GL              UDRAG          PHDOT             AMASS           AMOM            AENER
                                 XLH             DOB            DIB

   1     .10000000E+01   .59845918E-01   0.              .14444781E+03   0.               .14700000E+02   0.
                         0.              0.              0.               .264359B4E+00   .264359B4E+00   .264359B4E+00
                         0.              0.              0.
IDEBUG(20)
  PRESSURE AT GRID (1,1) IS      .14700000E+02
   THE MASS OF GAS IN THE SYSTEM IS      .0804
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)
        9      .149102E-03    .700412E+02    .396655E+02    .228290E+02    .227536E+02    .227557E+02   0.              0.
       19      .295771E-03    .135279E+03    .461916E+02    .228311E+02    .227544E+02    .227621E+02   0.              0.
```