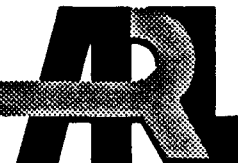


AD-A270 702



ARMY RESEARCH LABORATORY



The Modeling of Boattail Intrusion in a Lumped Parameter Interior Ballistic Code

Frederick W. Robbins
Robert T. Puhalla
Taquan S. Stewart

ARL-TR-181

August 1993



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

93-23103



93 10 1 2 3 4

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

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

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

REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE August 1993	3. REPORT TYPE AND DATES COVERED Final, Oct 89-Oct 91		
4. TITLE AND SUBTITLE The Modeling of Boattail Intrusion in a Lumped Parameter Interior Ballistic Code		5. FUNDING NUMBERS PR: 1L162618AH80		
6. AUTHOR(S) Frederick W. Robbins, Robert T. Puhalla, and Taquan S. Stewart				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-WT-PE Aberdeen Proving Ground, MD 21005-5066		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-B (Tech Lib) Aberdeen Proving Ground, MD 21005-5066		10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARL-TR-181		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This report extends both the traditional Lagrange gradient equation, and the chambrage gradient equation, to account for boattail intrusion. It compares the analytic pressure gradient with that predicted by XKTC and assesses the extent to which the effects of boattail intrusion account for the differences in boattail predictions. The gradient equation, with boattail addition, captures qualitatively, if not quantitatively, the effects for normal c/m's, although some questions remain about the physics of the gradient model.				
14. SUBJECT TERMS gradient equation; boattail; chambrage; interior ballistics		15. NUMBER OF PAGES 107		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

	<u>Page</u>
LIST OF FIGURES	v
LIST OF TABLES	v
1. INTRODUCTION	1
1.1 Models	1
1.2 Influence of Chambrage With a Boattail Intrusion	2
2. CALCULATIONS	13
3. RESULTS	15
3.1 Baselines	15
3.2 IB Calculations With Boattail and Chambrage	15
3.3 Drop in Pressure and Muzzle Velocity Due to Boattail	18
4. CONCLUSIONS	21
5. REFERENCES	23
APPENDIX A: The Derivation of a Gradient Equation With Area Change in a Tube and a Boattail	25
APPENDIX B: User's Manual and Code Listing for RGA	55
LIST OF SYMBOLS	101
DISTRIBUTION LIST	103

DTIC QUALITY INSPECTED 2

Accession For	
NTIS	CRA&I ☒
DTIC	TAB ☐
Unannounced ☐	
Justification _____	
By _____	
Distribution / _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

INTENTIONALLY LEFT BLANK.

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. The system to be modeled	3
2. Presentation of a moving area in tube	8

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Baselines (0 Boattail, 0 Chambrage)	15
2. IB Calculation With Boattail and Chambrage, C/M of .25	15
3. IB Calculation With Boattail and Chambrage, C/M of 1.0	16
4. IB Calculation With Boattail and Chambrage, C/M of 4.0	17
5. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of .25	18
6. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of 1.0	19
7. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of 4.0	20

INTENTIONALLY LEFT BLANK.

1. INTRODUCTION

During the analysis of 120-mm gun firings designed to look at the interior ballistic characteristics of combustible cartridge cases (Robbins, Koszoru, and Minor 1986), interior ballistic simulations using XNOVAKTC (XKTC) (Gough 1980) were in agreement with measured pressure-time curves and pressure-difference curves. IBHVG2 (Anderson and Fickie 1987) calculations gave a maximum breech pressure that was 42 MPa higher than measured. Parametric studies were performed using XKTC to attempt to attribute this disparity to the various processes omitted from IBHVG2. The boattail intrusion was calculated to account for 14 MPa, with effects of flamespreading and intergranular stress accounting for 3 MPa each. Subsequent calculations (Robbins 1986) indicated chambrage, propellant packaging, wave dynamics, and multiphase effects (the solid propellant velocity lag and concomitant formation of an ullage region between the projectile base and the propellant bed) as contributors to the differences between the lumped-parameter and two-phase interior ballistic codes. Further study demonstrated that the influence of chambrage and propellant velocity lag could be represented in an analytic gradient equation (Robbins, Anderson, and Gough 1990). This report extends both the traditional Lagrange gradient equation, and the chambrage gradient equation, to account for boattail intrusion. It compares the analytic pressure gradient with that predicted by XKTC and assesses the extent to which the effects of boattail intrusion account for the differences in ballistic predictions.

The family of NOVA codes, of which XKTC is the latest version, has been used with uncompromised databases to model gun systems successfully (Robbins, Koszoru, and Minor 1986; Robbins 1983; Robbins and Horst 1984). Since XKTC calculates the pressure gradient from first principles, and agrees with gun firings, XKTC simulations are assumed correct. Accordingly, all the lumped-parameter computer runs, with the different boattail gradients, are compared with equivalent XKTC computer runs.

1.1 Models. The analysis of the chambrage gradient equation can be traced back as far as Vinti (1942). The analyses of the chambrage gradient equation, the original analyses of the propellant velocity lag gradient equation, and the combination of the two were done by Gough (Robbins, Anderson, and Gough 1990), who is also responsible for the initial development of the boattail gradient equation (Gough in preparation). Similar analysis for the chambrage gradient for a different purpose has also been performed (Morrison and Wren 1989). Reasonable assumptions for the accommodation of a boattail intrusion were made, and this document represents the complete analysis.

1.2 Influence of Chambrage With a Boattail Intrusion. For the chambrage gradient equation with boattail effects, the propellant is assumed to be uniformly distributed between the breech and the base of the projectile. The variation in tube area is assumed to be confined to the chamber, while the area of the bore is constant. The boattail is assumed to be a right circular cylinder. This keeps the solution consistent with the simplicity of the lumped-parameter code, reduces computation time (cuts down on integration) and greatly simplifies the derivation.

The continuity and momentum equations for the unsteady flow of a homogeneous, inviscid substance through a tube with variable area are

$$\frac{\partial(A\rho)}{\partial t} + \frac{\partial(\rho Au)}{\partial z} = 0 \quad , \quad (1)$$

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial z} + \frac{\partial P}{\partial z} = 0 \quad , \quad (2)$$

where

A = cross-sectional area

P = pressure

ρ = density

u = velocity

t = time

z = distance.

The system to be modeled is shown in Figure 1. With reference to the figure,

z_a = distance from breech to aft end of projectile

z_p = distance from breech to base of projectile

A_A = cross-sectional area of the boattail

A_{BA} = cross-sectional area of projectile base (excluding boattail)

A_B = cross-sectional area of bore

$A_B = A_A + A_{BA}$

A = cross-sectional area of chamber

A_i = cross-sectional area of the tube in the area of the boattail

$$A = A_i + A_A$$

also

V_p = projectile velocity

$V(z,t)$ = volume up to position z , at time t

$A(z,t)$ = area at position z , at time t

$V(zp)$ = volume up to base of projectile.

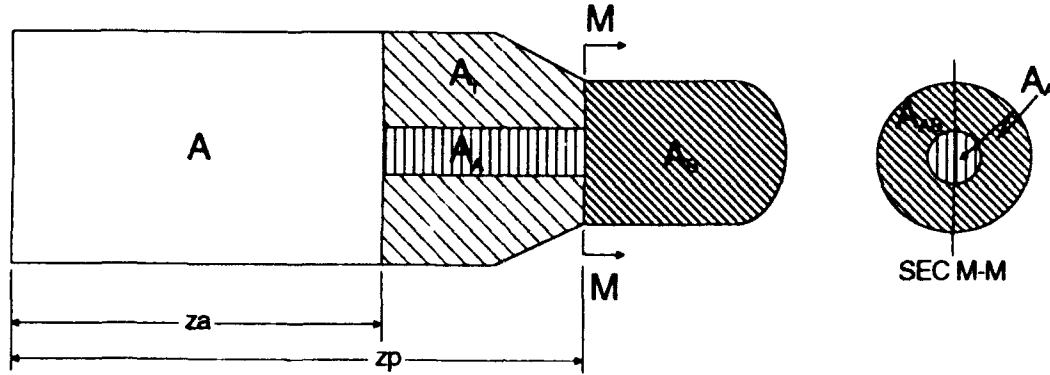


Figure 1. The system to be modeled.

The analysis, a detailed version of which is shown in Appendix A, considers two regions separately, the region from the breech to the aft end of the projectile and the region from the aft end of the projectile to base of the projectile. The analysis starts with the Lagrange assumption

$$\frac{\partial \rho}{\partial z} = 0 .$$

For

$$0 \leq z < z_a ,$$

the velocity, from the continuity equation (1), is

$$u(z,t) = \frac{A_B V_p V(z,t)}{V(zp)A(z,t)} , \quad (4)$$

and the velocity for

$$z_a < z \leq z_p$$

is

$$u(z,t) = \frac{A_B V_p V(z,t)}{V(z_p) A(z,t)} - \frac{A_A V_p}{A(z,t)}. \quad (5)$$

For the portion of space behind the projectile, $0 \leq z < z_a$, calculations of the pressure distribution from the breech to the aft end of the projectile can be performed by taking:

$P(z,t)$ = pressure at z and t

$P(z_a,t) = P(z_a)$ = pressure at aft end of projectile at time t

C = total charge mass

P_{BR} = pressure at the breech

P_B = pressure at the base of the projectile

P_{RES} = resistive pressure to the motion of the projectile

M_p = mass of the projectile,
projectile acceleration,

$$\dot{V}_p = \frac{P(z_a)A_A + P_B A_{BA} - A_B P_{RES}}{M_p}, \quad (6)$$

then using (4) and the momentum equation (2), $P(z_a)$ can be written as

$$P(z_a) = z a_0(t) P_{BR} + z a_1(t) + z a_2(t) P_B, \quad (7)$$

where

$$z a_0(t) = \frac{1}{\left[1 + \frac{C A_B A_A Q_1(z_a)}{V^2(z_p) M_p} \right]},$$

$$za_1(t) = \frac{\frac{CA_B^2 V_p^2 Q_1(za)}{V^3(zp)} + \frac{CA_B^2 P_{RES} Q_1(za)}{V^2(zp)M_p} - \frac{CA_B^2 V_p^2 Q_2(za)}{2V^3(zp)}}{\left[1 + \frac{CA_B A_A Q_1(za)}{V^2(zp)M_p} \right]},$$

$$za_2(t) = - \frac{\frac{CA_B A_{BA} Q_1(za)}{V^2(zp)M_p}}{\left[1 + \frac{CA_B A_A Q_1(za)}{V^2(zp)M_p} \right]},$$

and

$$Q_1(za) = \int_0^{za^-} \frac{V(za^-)}{A(za^-)} dz,$$

$$Q_2(za) = \frac{V^2(za^-)}{A^2(za^-)},$$

where

$$za^- = \lim_{\epsilon \rightarrow 0} za - \epsilon.$$

The pressure distribution $P(z)$ is

$$P(z) = P_{BR} + [a_3(t) + a_5(t)P_{BR} + a_4(t)P_B]Q_1(z) + b(t)Q_2(z), \quad (8)$$

where

$$a_3(t) = \frac{CA_B^2 V_p^2}{V^3(zp)} - \frac{CA_B A_A za_1(t)}{V^2(zp)M_p} + \frac{CA_B^2 P_{RES}}{V^2(zp)M_p},$$

$$a_4(t) = - \frac{CA_B A_A z a_2(t)}{V^2(z_p) M_p} - \frac{CA_B A_{BA}}{V^2(z_p) M_p} ,$$

$$a_5(t) = - \frac{CA_B A_A z a_0(t)}{V^2(z_p) M_p} ,$$

$$b(t) = - \frac{CA_B^2 V_p^2}{2 V^3(z_p)} ,$$

$$Q_1(z) = \int_0^z \frac{V(z)}{A(z)} dz ,$$

and

$$Q_2(z) = \frac{V^2(z)}{A^2(z)} .$$

Now, focusing on the portion from the aft end of the projectile to the base of the projectile...

$$z_a < z \leq z_p ,$$

$$\begin{aligned} P(z) = & z a_0(t) P_{BR} + z a_1(t) + z a_2(t) P_B + f k + a_3(t) Q_1(z a^*, t) \\ & + a_4(t) P_B Q_1(z a^*, t) + a_5(t) P_{BR} Q_1(z a^*, t) + c_3(t) Q_3(z a^*, t) \\ & + c_4(t) P_B Q_3(z a^*, t) + c_5(t) P_{BR} Q_3(z a^*, t) + b(t) Q_2(z, t) \\ & + h_1 Q_4(z, t) + j_1 Q_5(z, t) + k_1 , \end{aligned} \tag{9}$$

where

$$h_1 = \frac{CV_p^2 A_A A_B}{V^2(z_p)},$$

$$j_1 = - \frac{CA_A^2 V_p^2}{2V(z_p)},$$

$$za^+ = \lim_{\epsilon \rightarrow 0} za + \epsilon,$$

$$k_1 = \frac{CV_p^2 A_B^2 V^2(za^+)}{2V^3(z_p) A^2(za^+)} - \frac{CV_p^2 A_A A_B V(za^+)}{V^2(z_p) A^2(za^+)} + \frac{CA_A^2 V_p^2}{2V(z_p) A^2(za^+)},$$

$$Q_4(z, t) = \frac{V(z, t)}{A^2(z, t)},$$

$$Q_5(z, t) = \frac{1}{A^2(z, t)},$$

$$Q_1(za^+, t) = \int_{za^+}^z \frac{V(z, t)}{A(z, t)} dz,$$

$$Q_3(za^+, t) = \int_{za^+}^z \frac{\partial z}{A(z, t)},$$

$$c_3(t) = \frac{CA_A^2 z a_1(t)}{V(zp)M_p} - \frac{CA_A A_B P_{RES}}{V(zp)M_p} - \frac{CV_p^2 A_A A_B}{V^2(zp)},$$

$$c_4(t) = \frac{CA_A A_B A}{V(zp)M_p} + \frac{CA_A^2 z a_2(t)}{V(zp)M_p},$$

and

$$c_5(t) = \frac{CA_A^2 z a_0(t)}{V(zp)M_p}.$$

fk = jump in pressure across the boattail

An analysis by Kooker (April 1991) indicates that the pressure drop across the boattail (which in a one-dimensional analysis is equivalent to determining the pressure drop across a moving discontinuity in area, Figure 2) is given by:

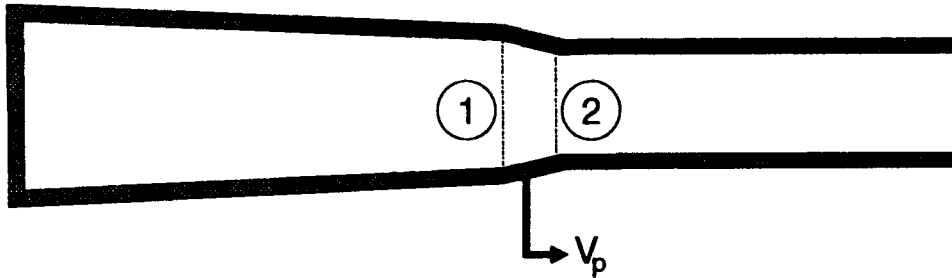


Figure 2. Presentation of a moving area in tube.

Mass balance:

$$\rho_1(u_1 - V_p)A_1 = \rho_2(u_2 - V_p)A_2 \equiv \dot{m} \quad (10)$$

Momentum balance:

$$\begin{aligned} \rho_2 u_2 (u_2 - V_p) A_2 - \rho_1 u_1 (u_1 - V_p) A_1 &= P_1 A_1 - P_2 A_2 + P_{\text{mean}} (A_2 - A_1) \\ &= - (P_2 - P_1) \left[\frac{A_1 + A_2}{2} \right] \end{aligned} \quad (11)$$

since

$$P_{\text{mean}} = \frac{1}{2} (P_1 + P_2).$$

Thus,

$$(P_2 - P_1) = \frac{2\rho(u_1 - V_p)A_1(u_2 - u_1)}{A_1 + A_2}, \quad (12)$$

where

$$A_1 = A(z a^-),$$

$$A_2 = A(z a^+),$$

$$\rho_1 = \rho_2 = \frac{C}{V(zp)},$$

$$u_1 = u(z a^-),$$

$$u_2 = u(z a^+),$$

or

$$P_2 - P_1 = \Delta k = \frac{-2 \left(\frac{C}{V(zp)} \right) \frac{V_p^2 A(z a^-) A_A}{A(z a^+)} \left[\frac{A_B V(z a)}{V(zp) A(z a^-)} - 1 \right]^2}{(A(z a^-) + A(z a^+))}. \quad (13)$$

We are now at a point where the projectile base and breech pressure can be determined. For use in lumped-parameter interior ballistic models, the gradient equation is usually cast in terms of the mean pressure (P_m):

$$P_m = \frac{\int_0^{z_p} A(z,t)P(z,t)dz}{\int_0^{z_p} A(z,t)dz} .$$

Substituting the pressure distribution and integrating,

$$P_m = b_1(z_p,t) + b_2(z_p,t)P_B + b_3(z_p,t)P_{BR} , \quad (15)$$

where

$$\begin{aligned} b_1(z_p,t) = & \frac{a_3(t)Q_7(z_a) + a_3(t)Q_9(z_p) + b(t)Q_6(z_p)}{V(z_p)} \\ & + \frac{za_1(t)(V(z_p) - V(z_a)) + fk(V(z_p) - V(z_a))}{V(z_p)} \\ & + \frac{c_3(t)Q_8(z_p) + h_1Q_1(z_p) + j_1Q_3(z_p)}{V(z_p)} \\ & + \frac{k_1(V(z_p) - V(z_a))}{V(z_p)} , \end{aligned}$$

$$\begin{aligned} b_2(z_p,t) = & \frac{a_4(t)Q_7(z_a) + a_4(t)Q_9(z_p) + za_2(t)(V(z_p) - V(z_a))}{V(z_p)} \\ & + \frac{c_4(t)Q_8(z_p)}{V(z_p)} , \end{aligned}$$

$$b_3(zp,t) = \frac{V(za) + a_5(t)Q_7(za) + a_5(t)Q_9(zp)}{V(zp)} + \frac{za_0(t)(V(zp) - V(za)) + c_5Q_8(zp)}{V(zp)},$$

$$Q_6(zp) = \int_0^{zp} \frac{V^2(z,t)}{A(z,t)} dz,$$

and

$$Q_7(za) = \int_0^{za} A(z,t) \int_0^z \frac{V(z,t)}{A(z,t)} \partial z \partial z,$$

$$Q_8(zp) = \int_{za}^{zp} A(z,t) \int_{za}^z \frac{\partial z}{A(z,t)} \partial z,$$

$$Q_9(zp) = \int_{za}^{zp} A(z,t) \int_{za}^z \frac{V(z,t)}{A(z,t)} \partial z \partial z.$$

$P(zp)$ can be determined from (9) and can be written as

$$P(zp) = P_B = \frac{l_3(zp,t)}{l_2(zp,t)} P_{BR} + \frac{l_1(zp,t)}{l_2(zp,t)}, \quad (16)$$

where

$$l_1(zp,t) = za_1(t) + \Gamma k + a_3(t)Q_1(zp) + c_3(t)Q_3(zp) + b(t)Q_2(zp) + h_1Q_4(zp) + j_1Q_5(zp) + k_1,$$

$$l_2(zp,t) = 1 - za_2(t) - a_4(t)Q_1(zp) - c_4(t)Q_3(zp),$$

$$l_3(zp,t) = za_0(t) + a_5(t)Q_1(zp) + c_5(t)Q_3(zp),$$

$$Q_1(zp) = \int_{za}^{zp} \frac{V(z,t)}{A(z,t)} dz,$$

$$Q_2(zp) = \frac{V^2(zp)}{A^2(zp)},$$

$$Q_3(zp) = \int_{za}^{zp} \frac{\partial z}{A(z,t)},$$

$$Q_4(zp) = \frac{V(zp)}{A^2(zp)},$$

and

$$Q_5(zp) = \frac{1}{A^2(zp)}.$$

Therefore, (15) and (16) are solved simultaneously:

$$P_B = \frac{\frac{P_m}{b_3(zp,t)} + \frac{b_1(zp,t)}{b_3(zp,t)} + \frac{l_1(zp,t)}{l_3(zp,t)}}{\frac{b_2(zp,t)}{b_3(zp,t)} + \frac{l_2(zp,t)}{l_3(zp,t)}} \quad (17)$$

and

$$P_{BR} = \frac{l_2(zp,t)}{l_3(zp,t)} P_B - \frac{l_1(zp,t)}{l_3(zp,t)} . \quad (18)$$

The energy in the fluid is represented by

$$dE = \frac{1}{2} u^2 dm , \quad (19)$$

$$dm = \rho A dz ,$$

and using (4) and (5),

$$E = \int_0^{zp} dE = - b(t) Q_6(zp) - h_1 Q_1(zp) - j_1 Q_3(zp) . \quad (20)$$

2. CALCULATIONS

Interior ballistic simulations were performed with IBRGAB (Appendix B—an interior ballistic code into which the boattail gradient had been incorporated) to investigate the influence on the interior ballistic trajectory of a flat-based projectile with a boattail (with and without chambrage). The same simulations were done with XKTC using XKTC databases as consistent with those of IBRGAB as the physical scope of XKTC allows.

The calculations performed with IBRGAB involved databases with evenly distributed, seven-perforated propellant having an initial porosity of 0.4579 and zero barrel resistance, and assumed the propellant ignited at an initial instant. All calculations were performed for a flat-based projectile with and without a boattail and no heat loss.

The parameters used in the computer codes were:

Bore diameter	127 mm
Volume	.01 m ³
Travel	4.572 m
Propellant mass	9.0 kg
Projectile mass	2.25, 9.0, 36.0.

Propellant characteristics were:

Impetus	$.1136 \times 10^7 \text{ J/kg}$
Covolume	$.976 \times 10^{-3} \text{ m}^3/\text{kg}$
Gamma	1.23
Flame temperature	3,143 K
Molecular weight	23.0 kg/kg mole
Density	$.16605 \times 10^4 \text{ kg/m}^3$
Burning rate	$.00110519P^{1.0} \text{ m/s}$ (P is in MPa).

The maximum breech pressure studied in IBRGAB was 500 MPa. It was obtained by varying the outer diameter of the propellant grain, while holding the grain length constant at .03175m. Calculations were performed for a combination of four chambers, each with four boattail values:

- 1) 0% chambrage with 0, 5, 10, 15% boattail
- 2) 5% chambrage with 0, 5, 10, 15% boattail
- 3) 10% chambrage with 0, 5, 10, 15% boattail
- 4) 15% chambrage with 0, 5, 10, 15% boattail.

The chamber with 0% chambrage and 0% boattail is a straight tube with the chamber diameter equal to the bore diameter. The 5, 10, and 15% chambrage chambers were obtained by adding 5, 10, and 15% of the bore diameter, respectively, to the diameter of the breech for the 0% chambrage chamber. With the increase in chambrage, there is a decrease in chamber length to keep the volume constant. The breech diameter increases from .127 m to .146 m (0 to 15%), while the chamber length decreased to .78430 m from .90782 m. The boattail length is held constant at .508 m and the boattail percentages simply represent that fraction of the constant chamber volume. The different chambrage and boattail combinations are arranged in tables according to the ratio of the charge mass to the projectile mass (c/m). In the tables, the nomenclature, bt, refers to the boattail and ch refers to chambrage. Each maximum breech pressure and muzzle velocity drop is clearly seen in both XKTC and IBRGAB portions of the tables. Table 1 gives the baseline maximum breech pressures and muzzle velocities for each c/m in XKTC and IBRGAB. In IBRGAB, maximum breech pressure is forced to 500 MPa by varying the grain diameter. An equivalent XKTC database was then made to yield the values given. Tables 2 through 4 contain the explicit results of the different combinations of boattail and chambrage, emphasizing the effects of the boattail for c/m of .25, 1.0, and 4.0. For example, in any table, to go from the baseline calculation

to a 10% boattail and a 5% chambrage, a boattail is added which is 10% of the chamber volume, and the chamber description is altered by adding 5% of the bore diameter to the diameter of the breech. Arranged in the same form are tables 5 through 6 depicting the differences from the effects of the boattail. The differences are obtained by subtracting either the 5, 10, or 15% boattail result from the baseline result. These values allow each table to clearly illustrate that a larger boattail combined with a larger chambrage give a greater drop in maximum breech pressure and to show the maximum breech pressure drop increases while the c/m ratio increases.

3. RESULTS

3.1 Baselines.

Table 1. Baselines (0 Boattail, 0 Chambrage)

C/M	XKTC		IBRGAB		
	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	O.D. (m)
.25	502.6	898.8	500.0	899.5	.016599
1.0	540.3	1,641.3	499.9	1,596.8	.009879
4.0	512.8	2,538.4	499.9	2,342.6	.007237

3.2 IB Calculations With Boattail and Chambrage.

Table 2. IB Calculation With Boattail and Chambrage, C/M of .25

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
00	00	502.6	898.8	500.0	899.5
05	00	501.0	897.9	497.9	898.9
10	00	496.1	896.8	495.9	898.3

Table 2. IB Calculation With Boattail and Chambrage, C/M of .25 (continued)

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
15	00	497.8	896.1	494.0	897.8
00	05	500.6	898.2	497.8	898.8
05	05	499.2	897.6	495.5	898.1
10	05	496.0	895.7	493.4	897.4
15	05	498.8	897.1	491.4	896.8
00	10	498.8	897.8	495.8	898.1
05	10	497.2	897.2	495.2	898.1
10	10	494.8	895.9	491.0	896.7
15	10	489.4	893.2	488.8	896.0
00	15	497.0	897.3	494.1	897.6
05	15	495.7	896.8	491.3	896.7
10	15	493.3	896.0	488.7	896.0
15	15	489.0	893.6	486.3	895.2

Table 3. IB Calculation With Boattail and Chambrage, C/M of 1.0

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
00	00	540.3	1,641.3	499.9	1,596.8
05	00	530.2	1,635.2	493.4	1,593.1
10	00	516.3	1,626.9	487.1	1,589.5
15	00	509.5	1,623.7	481.3	1,586.0
00	05	531.7	1,638.3	492.8	1,592.0

Table 3. IB Calculation With Boattail and Chambrage, C/M of 1.0 (continued)

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
05	05	523.0	1,632.7	485.7	1,587.8
10	05	510.8	1,625.2	479.0	1,583.6
15	05	507.0	1,623.2	472.7	1,579.8
00	10	523.5	1,635.3	486.4	1,587.8
05	10	515.0	1,629.9	480.3	1,584.4
10	10	505.0	1,623.1	471.3	1,578.7
15	10	491.0	1,624.6	464.7	1,574.2
00	15	515.8	1,632.3	480.9	1,584.3
05	15	507.9	1,626.9	472.3	1,579.1
10	15	498.3	1,620.5	464.4	1,574.0
15	15	486.6	1,612.1	457.1	1,569.0

Table 4. IB Calculation With Boattail and Chambrage, C/M of 4.0

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
00	00	512.8	2,538.4	499.9	2,342.6
05	00	510.1	2,523.1	485.6	2,328.5
10	00	505.9	2,499.9	472.3	2,314.7
15	00	507.5	2,486.7	460.0	2,301.1
00	05	503.0	2,502.5	484.9	2,323.8
05	05	496.3	2,484.9	469.6	2,307.4
10	05	489.6	2,461.1	455.5	2,290.9

Table 4. IB Calculation With Boattail and Chambrage, C/M of 4.0 (continued)

OUTPUT		XKTC		IBRGAB	
%bt	%ch	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)	P _{MAX} BREECH (MPa)	MUZZLE VELOCITY (m/s)
15	05	490.1	2,448.9	442.5	2,276.9
00	10	494.0	2,469.9	471.7	2,306.7
05	10	484.2	2,449.0	456.6	2,290.1
10	10	476.1	2,424.0	439.9	2,270.0
15	10	464.3	2,391.7	426.2	2,252.3
00	15	485.1	2,440.3	460.3	2,291.8
05	15	472.8	2,415.9	442.2	2,270.6
10	15	463.6	2,389.1	425.8	2,250.3
15	15	452.1	2,356.9	411.1	2,230.8

3.3. Drop in Pressure and Muzzle Velocity Due to Boattail.

Table 5. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of .25

OUTPUT		XKTC		IBRGAB	
%bt	%ch	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL (m/s)	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL (m/s)
00	00	0.0	0.0	0.0	0.0
05	00	1.6	0.9	2.1	0.6
10	00	6.5	2.0	4.1	1.2
15	00	4.8	2.7	6.0	1.7
00	05	2.0	0.6	2.2	0.6
05	05	3.4	1.2	4.5	1.4
10	05	6.6	3.1	6.6	2.1
15	05	3.8	1.7	8.6	3.3
00	10	3.8	1.0	4.2	1.4

Table 5. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of .25 (continued)

OUTPUT		XKTC		IBRGAB	
%bt	%ch	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)
05	10	5.4	1.6	4.8	1.4
10	10	7.8	2.9	9.0	2.8
15	10	13.2	5.6	11.2	3.5
00	15	5.6	1.5	5.9	1.9
05	15	6.9	2.0	8.7	2.8
10	15	9.3	2.8	11.3	3.5
15	15	13.6	5.2	13.7	4.3

Table 6. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of 1.0

OUTPUT		XKTC		IBRGAB	
%bt	%ch	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)
00	00	0.0	0.0	0.0	0.0
05	00	10.1	6.1	6.5	3.7
10	00	24.0	14.4	12.8	7.3
15	00	30.8	17.6	18.6	10.8
00	05	8.6	3.0	7.1	4.8
05	05	17.3	8.6	14.2	9.0
10	05	29.5	16.1	20.9	13.2
15	05	33.3	18.1	27.2	17.0
00	10	16.8	6.0	13.5	9.0
05	10	25.3	11.4	19.6	12.4
10	10	35.3	18.2	28.6	18.1
15	10	49.3	16.7	35.2	22.6

Table 6. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of 1.0 (continued)

OUTPUT		XKTC		IBRGAB	
%bt	%ch	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)
00	15	24.5	9.0	19.0	12.5
05	15	32.4	14.4	27.6	17.7
10	15	42.0	20.8	35.5	22.8
15	15	53.7	29.2	42.8	27.8

Table 7. Drop in Pressure and Muzzle Velocity Due to Boattail, C/M of 4.0

OUTPUT		XKTC		IBRGAB	
%bt	%ch	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)	ΔP_{MAX} BRCH (MPa)	$\Delta MUZZLE$ VEL. (m/s)
00	00	0.0	0.0	0.0	0.0
05	00	2.7	15.3	14.3	14.1
10	00	6.9	38.5	27.6	27.9
15	00	5.3	51.7	39.9	41.5
00	05	9.8	35.9	15.0	18.8
05	05	16.5	53.5	30.3	35.2
10	05	23.2	77.3	44.4	51.7
15	05	22.8	89.5	57.4	65.7
00	10	18.8	68.5	28.2	35.9
05	10	28.6	89.4	43.3	52.5
10	10	36.7	114.4	60.0	72.6
15	10	48.5	146.7	73.7	90.3
00	15	27.7	98.1	39.6	50.8
05	15	40.0	122.5	57.7	72.0
10	15	49.2	149.3	74.1	92.3
15	15	60.7	181.5	88.8	111.8

It was expected that the introduction of a boattail without any volume change would lead to a lowering of the maximum breech pressure. This was expected because the aft end of the projectile is subject to a higher pressure than the base of the projectile and therefore is accelerated faster. More chamber volume is then opened up, leading to a decrease in maximum breech pressure (P_{max}).

In the baseline analysis (Table 1, 0 boattail, 0 chambrage), the calculated maximum breech pressures are within 10% of the maximum breech pressures given by XKTC for a comparable database, and the velocities are within 8%.

With a c/m of .25, IBRGAB captures the boattail effects as measured by the change in maximum breech pressure and velocity in a qualitative, if not quantitative manner. When compared to XKTC, the calculations with 15% boattail and chambrage of 0 and 5% exhibit a rise in maximum breech pressure above that with 10% boattail. This was unexpected.

Again, for a c/m of 1.0, the agreement in the change in maximum breech pressure is good. Note that the change in maximum breech pressure in IBRGAB is slightly smaller than the change in maximum breech pressure in XKTC, yet both are rather uniform.

Lastly, at c/m of 4.0, the pressure drop in XKTC for small boattails is unexpectedly small (less than for $c/m = 1$). It is also smaller than that of the analytic gradient with boattail. Similar effects of the reversal of the change in maximum breech pressures were observed for a c/m of .25 in XKTC.

4. CONCLUSIONS

A boattail has been incorporated into the gradient equation of a lumped-parameter interior ballistics model with reasonable effects.

The gradient equation, with boattail addition, captures qualitatively, if not quantitatively, the effects for normal c/m 's (.25 and 1.0).

For c/m of 4.0, or large c/m 's, some questions remain about the physics of the gradient model.

INTENTIONALLY LEFT BLANK.

5. REFERENCES

- Anderson, R. D., and K. D. Fickie. "IBHVG2—A User's Guide." BRL-TR-2829, U.S.A. ARRADCOM, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, July 1987.
- Gough, P. S. Contractor Report, DAAK11-85-D-0002, in preparation.
- Gough, P. S. "The Nova Code: A User's Manual." Indian Head Contract Report IHCR80-8, Naval Ordnance Station, Indian Head, MD, 1980.
- Morrison, W. F., and G. P. Wren. "A Lumped-Parameter Description of Liquid Injection in a Regenerative Liquid Propellant Gun." Proceedings of the 23rd JANNAF Combustion Meeting, CPIA Publication 457, vol. 2, pp. 464-489, October 1988.
- Robbins, F. W. "Comparison of TDNOVA Results With an Analytic Solution." BRL-MR-03299, U.S.A. ARRADCOM, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, July 1983.
- Robbins, F. W. "Studies Supporting Development of a Modified Gradient Equation for Lumped-Parameter Interior Ballistic Codes." Proceedings of the 12th Meeting of The Technical Cooperation Program, vol. 5, October 1986.
- Robbins, F. W., and A. W. Horst. "Detailed Characterization of the Interior Ballistics of Slotted Stick Propellant." BRL-TR-2591, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, September 1984.
- Robbins, F. W., A. A. Koszoru, and T. C. Minor. "A Theoretical and Experimental Interior Ballistic Characterization of Combustible Cases." Proceedings of the 9th International Symposium of Ballistics, Part 1, pp. 21-28, April 1986.
- Robbins, F. W., R. D. Anderson, and P. S. Gough. "New Pressure Gradient Equations for Lumped-Parameter Interior Ballistic Codes." BRL-TR-3097, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, May 1990.
- Vinti, J. P. "The Equations of Interior Ballistics." BRL Report 307, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, October 1942.

INTENTIONALLY LEFT BLANK.

APPENDIX A:
**THE DERIVATION OF A GRADIENT EQUATION
WITH AREA CHANGE IN A TUBE AND A BOATTAIL**

INTENTIONALLY LEFT BLANK.

The continuity and momentum equations for the unsteady flow of a homogeneous, inviscid substance through a tube with variable area are

$$\frac{\partial(A(z, t)\rho)}{\partial t} + \frac{\partial(\rho A(z, t)u(z, t))}{\partial z} = 0, \quad (1)$$

$$\rho \frac{\partial u(z, t)}{\partial t} + \rho u \frac{\partial u(z, t)}{\partial z} + g_o \frac{\partial P}{\partial z} = 0, \quad (2)$$

where A = cross sectional area
 P = pressure
 ρ = density
 u = velocity
 t = time
 z = distance.

If we take the boattail to be a right circular cylinder (Figure A-1),

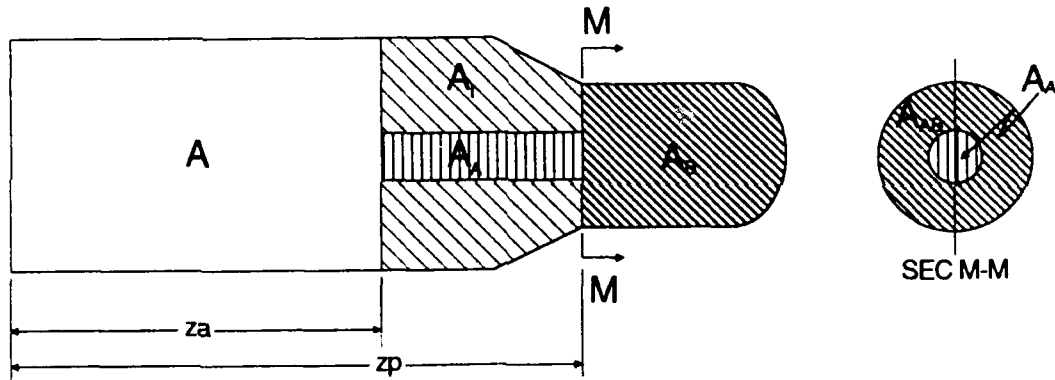


Figure A-1. The system to be modeled.

then $A_B = A_{BA} + A_A$, where

A_A = cross-sectional area of the boattail,
 A_{BA} = Area of the base of the projectile exposed to fluid,

and $A_B = A_A + A_i$, where

A = external area of the tube,
 A_i = internal area of the tube(account for boattail).

We start by examining the area from the breech to the aft end of the projectile.

$$0 \leq z < z_a,$$

where A is the area associated with the chamber and tube wall. Performing the indicated differentiation of (1) and making the Lagrange assumption,

$$\frac{\partial \rho}{\partial z} = 0 ,$$

(1) becomes

$$\frac{\partial A(z, t) u(z, t)}{\partial z} = - \frac{A(z, t)}{\rho} \frac{\partial \rho}{\partial t} - \frac{\partial A(z, t)}{\partial t} . \quad (3)$$

The density can be written as

$$\rho_{(of\ the\ fluid)} = \frac{C}{V(zp)} . \quad (4)$$

The fluid is considered to be composed of solid and gaseous propellant.

C = total mass of propellant (fluid),

or

C = initial mass of propellant (solid),

and

V(zp) = chamber volume up to the base of the projectile.

Differentiating the density gives

$$\frac{\partial \rho}{\partial t} = - \frac{\rho}{V(zp)} \frac{\partial V(zp)}{\partial t} ,$$

but

$$\frac{\partial V(zp)}{\partial t} = A_p V_p ,$$

where V_p = velocity of the projectile.

With the boundary condition

$$u(0) = 0$$

and

$$\frac{\partial A(z, t)}{\partial t} = 0$$

for

$$0 \leq z < z_a ,$$

(3) becomes

$$u(z) = \frac{A_B V_p V(z, t)}{A(z, t) V(z_p)} . \quad (5)$$

...from the aft end of the boattail to the projectile base,

$$z_a < z \leq z_p .$$

Once again performing the indicated differentiation of (1) and making the Lagrange assumption,

$$\frac{\partial(A(z, t) u(z, t))}{\partial z} = \frac{A(z, t) A_B V_p}{V(z_p)} - \frac{\partial A(z, t)}{\partial t} .$$

Since the boattail is a right circular cylinder

$$\frac{\partial A(z, t)}{\partial t} = 0 ,$$

where $A = A_i$ accounts for boattail area;
for

$$z_a < z \leq z_p$$

and

$$\int_{z_a^*}^z \partial(A(z, t) u(z, t)) = \frac{A_B V_p}{V(z_p)} \int_{z_a^*}^z A(z, t) \partial z ,$$

where

$$z_a^* = \lim_{\epsilon \rightarrow 0} z_a + \epsilon .$$

$$A(z, t) u(z, t) - A(z_a^*) u(z_a^*) = \frac{A_B V_p}{V(z_p)} [V(z, t) - V(z_a^*, t)] ,$$

and with the boundary condition

$$u(z_p) = V_p ,$$

$$V_p \frac{(V(zp)A(zp) - A_B(V(zp) - V(za^+)))}{V(zp)A(za^+)} = u(za^+) .$$

Since $A_B = A_A + A_{BA}$ and $A(zp) = A_{BA}$,

$$u(za^+) = V_p \left(\frac{A_B V(za^+) - A_A V(zp)}{V(zp)A(za^+)} \right) .$$

Therefore,

$$A(z, t) u(z, t) = \frac{A_B V_p}{V(zp)} (V(z, t) - V(za^+)) + V_p \left(\frac{A_B V(za^+) - A_A V(zp)}{V(zp)} \right) ,$$

and

$$u(z, t) = \frac{V_p A_B V(z, t)}{V(zp)A(z, t)} - \frac{A_A V_p}{A(z, t)} . \quad (6)$$

To obtain the pressure distribution, we once again examine the area from the breach to the aft end of the boattail.

$$0 \leq z < za .$$

Differentiate (5) with respect to time.

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{A_B \left(\frac{\partial V_p}{\partial t} \right) V(z, t)}{V(zp)A(z, t)} + \frac{A_B V_p \left(\frac{\partial V(z, t)}{\partial t} \right)}{V(zp)A(z, t)} \\ &- \frac{A_B V_p V(z, t)}{V^2(zp)A(z, t)} \left(\frac{\partial V(zp)}{\partial t} \right) - \frac{A_B V_p V(z, t)}{V(zp)A^2(z, t)} \left(\frac{\partial A(z, t)}{\partial t} \right) \end{aligned}$$

with

$$\frac{\partial V(z, t)}{\partial t} = 0 ,$$

$$\frac{\partial V_p}{\partial t} = \dot{V}_p ,$$

$$\frac{\partial V(zp)}{\partial t} = A_B V_p ,$$

$$\frac{\partial A(z, t)}{\partial t} = 0 ,$$

then

$$\frac{\partial u(z, t)}{\partial t} = \frac{A_B \dot{V}_p V(z, t)}{V(zp) A(z, t)} - \frac{A_B^2 V_p^2 V(z, t)}{V^2(zp) A(z, t)} . \quad (7)$$

Differentiate (5) with respect to distance.

$$\frac{\partial u(z, t)}{\partial z} = \frac{A_B V_p \left(\frac{\partial V(z, t)}{\partial z} \right)}{V(zp) A(z, t)} - \frac{A_B V_p V(z, t)}{V(zp) A^2(z, t)} \left(\frac{\partial A(z, t)}{\partial z} \right)$$

with

$$\frac{\partial V(z, t)}{\partial z} = A(z, t) ,$$

then

$$\frac{\partial u(z, t)}{\partial z} = \frac{A_B V_p}{V(zp)} - \frac{A_B V_p V(z, t)}{V(zp) A^2(z, t)} \left(\frac{\partial A(z, t)}{\partial z} \right) . \quad (8)$$

Substitute (4), (5), (7), and (8) into equation (2), the momentum equation, and we get the equation:

$$\begin{aligned}
& \frac{C}{V(zp)} \left(\frac{A_B \dot{V}_p V(z, t)}{V(zp) A(z, t)} - \frac{A_B^2 V_p^2 V(z, t)}{V^2(zp) A(z, t)} \right) \\
& + \frac{C}{V(zp)} \frac{A_B V_p V(z, t)}{V(zp) A(z, t)} \left(\frac{A_B V_p}{V(zp)} \right) \\
& - \frac{C}{V(zp)} \frac{A_B V_p V(z, t)}{V(zp) A(z, t)} \left(\frac{A_B V_p V(z, t)}{V(zp) A^2(z, t)} \left(\frac{\partial A(z, t)}{\partial z} \right) \right) + \frac{\partial P}{\partial z} = 0 .
\end{aligned}$$

Or

$$\frac{\partial P}{\partial z} = \frac{-CA_B \dot{V}_p V(z, t)}{V^2(zp) A(z, t)} + \frac{CA_B^2 V_p^2 V^2(z, t)}{V^3(zp) A^3(z, t)} \left(\frac{\partial A(z, t)}{\partial z} \right)$$

and

$$\int_0^z \partial P = \int_0^z \frac{-CA_B \dot{V}_p V(z, t)}{V^2(zp) A(z, t)} \partial z + \int_0^z \frac{CA_B^2 V_p^2 V^2(z, t)}{V^3(zp) A^3(z, t)} \left(\frac{\partial A(z, t)}{\partial z} \right) \partial z .$$

With the definition $P(0) = P_{BR}$

$$\begin{aligned}
P(z) &= P_{Br} - \frac{CA_B \dot{V}_p}{V^2(zp)} \int_0^z \left(\frac{V(z, t)}{A(z, t)} \right) \partial z \\
&+ \frac{CA_B^2 V_p^2}{V^3(zp)} \int_0^z \frac{V^2(z, t)}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z
\end{aligned} \tag{9}$$

Using Integration by Parts for the second part of equation (9),

$$\int u dv = uv - \int v du$$

with the following substitutions:

$$dv = \frac{\partial A(z, t)}{A^3(z, t)}$$

$$u = V^2(z, t)$$

$$v = -\frac{1}{2A(z, t)^2}$$

$$du = 2V(z, t) dV = 2V(z, t) A(z, t) \partial z$$

$$\int_0^z \frac{V^2(z, t)}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z = -\frac{V^2(z, t)}{2A^2(z, t)} + \int_0^z \frac{2V(z, t)A(z, t)}{2A^2(z, t)} \partial z .$$

$$\int_0^z \frac{V^2(z, t)}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z = \frac{-V^2(z, t)}{2A^2(z, t)} + \int_0^z \frac{V(z, t)}{A(z, t)} \partial z . \quad (10)$$

By substituting from equation (10) into equation (9) and factoring, the result becomes

$$P(z) = P_{Br} + \left[-\frac{CA_B \dot{V}_p}{V^2(zp)} + \frac{CA_B^2 V_p^2}{V^3(zp)} \right] \int_0^z \frac{V(z, t)}{A(z, t)} \partial z . \quad (11)$$

$$- \frac{CA_B^2 V_p^2}{2V^3(zp)} \frac{V^2(z, t)}{A^2(z, t)}$$

The acceleration of the projectile, $\partial V_p / \partial t$, is defined as

$$\frac{\partial V_p}{\partial t} = \frac{P(za)A_A + P_B A_{BA} - A_B P_{res}}{m_p} ,$$

where m_p = mass of the projectile.

Substituting $\partial V_p / \partial t$ into equation (10),

$$P(z) = P_{Br}$$

$$+ \left[\frac{CA_B^2 V_p^2}{V^3(zp)} - \frac{CA_B A_A P(za)}{V^2(zp) m_p} \right] \int_0^z \frac{V(z, t)}{A(z, t)} \partial z$$

$$+ \left[-\frac{CA_B A_{BA} P_B}{V^2(zp) m_p} + \frac{CA_B^2 P_{res}}{V^2(zp) m_p} \right] \int_0^z \frac{V(z, t)}{A(z, t)} \partial z \quad (12)$$

$$- \frac{CA_B^2 V_p^2 V^2(z, t)}{2V^3(zp) A^2(z, t)} .$$

Evaluating equation (12) at $z = z_a$ and solving for $P(z_a)$ and defining $P(z_a^-) = P(z_a)$,

$$P(z_a) = \left[P_{Br} + \frac{CA_B^2 V_p^2}{V^3(z_p)} \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z - \frac{CA_B A_{2A} P_B}{V^2(z_a) m_p} \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z \right] \left[1 + \frac{CA_B A_A \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z}{V^2(z_p) m_p} \right] + \left[\frac{CA_B^2 P_{res}}{V^2(z_p) m_p} \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z - \frac{CA_B^2 V_p^2}{2 V^3(z_p)} \frac{V^2(z_a^-)}{A^2(z_a^-)} \right] \left[1 + \frac{CA_B A_A \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z}{V^2(z_p) m_p} \right]$$

or

$$P(z_a) = z_{a0} P_{Br} + z_{a1} + z_{a2} P_B ,$$

where

$$Q_1(z_a^-) = \int_0^{z_a^-} \frac{V(z, t)}{A(z, t)} \partial z ,$$

$$Q_2(z_a^-) = \frac{V^2(z_a^-)}{A^2(z_a^-)} ,$$

$$z_{a0} = \frac{1}{\left[1 + \frac{CA_B A_A Q_1(z_a^-)}{V^2(z_p) m_p} \right]} ,$$

$$z_{a1} = \frac{\left[\frac{CA_B^2 V_p^2}{V^3(zp)} Q_1(za^-) + \frac{CA_B^2 P_{res}}{V^2(zp) m_p} Q_1(za^-) - \frac{CA_B^2 V_p^2}{2V^3(zp)} Q_2(za^-) \right]}{\left[1 + \frac{CA_B A_A}{V^2(zp) m_p} Q_1(za^-) \right]}$$

$$z_{a2} = - \frac{\frac{CA_B A_{BA} Q_1(za^-)}{V^2(zp) m_p}}{\left[1 + \frac{CA_B A_A Q_1(za^-)}{V^2(zp) m_p} \right]}$$

Then

$$P(z) = P_{Br} + [a_3(t) + a_4(t) P_B + a_5(t) P_{Br}] Q_1(z) + b(t) Q_2(z) ,$$

where

$$a_3(t) = \frac{CA_B^2 V_p^2}{V^3(zp)} - \frac{CA_B A_A z_{a1}}{V^2(zp) m_p} + \frac{CA_B^2 P_{res}}{V^2(zp) m_p} ,$$

$$a_4(t) = - \frac{CA_B A_A z_{a2}}{V^2(zp) m_p} - \frac{CA_B A_{BA}}{V^2(zp) m_p} ,$$

$$a_5(t) = - \frac{CA_B A_A z_{a0}}{V^2(zp) m_p} ,$$

$$b(t) = - \frac{CA_B^2 V_p^2}{2V^3(zp)} ,$$

$$Q_1(z) = \int_0^z \frac{V(z, t)}{A(z, t)} \partial z ,$$

$$Q_2(z) = \frac{V^2(z)}{A^2(z)} .$$

Now the area from the aft end of the projectile to the base of the projectile will be examined.

$$z_a < z \leq z_p$$

Velocity is now

$$u(z, t) = \frac{V_p A_B V(z, t)}{V(z_p) A(z, t)} - \frac{A_A V_p}{A(z, t)} \quad (13)$$

Taking the partial derivative of u with respect to t and making some substitutions

$$\begin{aligned} \frac{\partial A(z, t)}{\partial t} &= 0, \\ \frac{\partial V(z, t)}{\partial t} &= A_A V_p, \\ \frac{\partial V(z_p)}{\partial t} &= A_B V_p, \end{aligned}$$

the equation for $\partial u(z, t) / \partial t$ then becomes

$$\begin{aligned} \frac{\partial u(z, t)}{\partial t} &= \frac{\dot{V}_p A_B V(z, t)}{V(z_p) A} + \frac{V_p^2 A_B A_A}{V(z_p) A(z, t)} \\ &\quad - \frac{V_p^2 A_B^2 V(z, t)}{V^2(z_p) A(z, t)} - \frac{A_A \dot{V}_p}{A(z, t)}. \end{aligned}$$

Taking the partial derivative of equation (13) with respect to z ,

$$\begin{aligned} \frac{\partial u(z, t)}{\partial z} &= \frac{V_p A_B \frac{\partial V(z, t)}{\partial z}}{V(z_p) A(z, t)} \\ &\quad - \frac{V_p A_B V(z, t)}{V(z_p) A^2} \frac{\partial A(z, t)}{\partial z} + \frac{A_A V_p}{A^2(z, t)} \frac{\partial A(z, t)}{\partial z} \\ \frac{\partial V(z, t)}{\partial z} &= A(z, t) \end{aligned}$$

$$\frac{\partial u(z, t)}{\partial z} = \frac{V_P A_B}{V(zp)} - \frac{V_P A_B}{V(zp)} \frac{V(z, t)}{A^2(z, t)} \frac{\partial A(z, t)}{\partial z} + \frac{A_A V_P}{A^2(z, t)} \frac{\partial A(z, t)}{\partial z}.$$

Since

$$\rho \frac{\partial u(z, t)}{\partial t} + \rho u \frac{\partial u(z, t)}{\partial z} + \frac{\partial P}{\partial z} = 0, \quad (14)$$

equation (14) becomes

$$\begin{aligned} & \frac{C}{V(zp)} \left[\frac{\dot{V}_P A_B V(z, t)}{V(zp) A(z, t)} + \frac{V_P^2 A_B A_A}{V(zp) A} \right] \\ & - \frac{C}{V(zp)} \left[\frac{V_P^2 A_B^2 V(z, t)}{V^2(zp) A(z, t)} - \frac{A_A}{A(z, t)} \dot{V}_P \right] \\ & + \frac{C}{V(zp)} \left(\frac{V_P A_B V(z, t)}{V(zp) A(z, t)} - \frac{A_A V_P}{A(z, t)} \right) \frac{V_P A_B}{V(zp)} \\ & - \frac{C}{V(zp)} \left(\frac{V_P A_B V(z, t)}{V(zp) A(z, t)} - \frac{A_A V_P}{A(z, t)} \right) \left[\frac{V_P A_B V(z, t)}{V(zp) A^2} \frac{\partial A(z, t)}{\partial z} \right] \\ & + \frac{C}{V(zp)} \left(\frac{V_P A_B V(z, t)}{V(zp) A(z, t)} - \frac{A_A V_P}{A(z, t)} \right) \left[\frac{A_A V_P}{A^2(z, t)} \frac{\partial A(z, t)}{\partial z} \right] \\ & + \frac{\partial P}{\partial z} = 0. \end{aligned} \quad (15)$$

Solving equation (15) for ∂P and integrating,

$$\begin{aligned} \int_{za}^z \partial P &= - \frac{C \dot{V}_P A_B}{V^2(zp)} \int_{za}^z \frac{V(z, t)}{A(z, t)} \partial z + \frac{C A_A \dot{V}_P}{V(zp)} \int_{za}^z \frac{\partial z}{A(z, t)} \\ &+ \frac{C V_P^2 A_B^2}{V^3(zp)} \int_{za}^z \frac{V^2(z, t)}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z \\ &- \frac{2 C V_P^2 A_A A_B}{V^2(zp)} \int_{za}^z \frac{V(z, t)}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z + \frac{C A_A^2 V_P^2}{V(zp)} \int_{za}^z \frac{1}{A^3(z, t)} \frac{\partial A(z, t)}{\partial z} \partial z. \end{aligned} \quad (16)$$

Integrating portions of equation (16) by parts

$$\int v du = uv - \int u dv ,$$

let

$$v = V^2(z, t) ,$$

$$du = \frac{\partial A(z, t)}{A^3(z, t)} ,$$

$$dv = 2V(z, t)A(z, t)\partial z ,$$

$$u = -\frac{1}{2A^2(z, t)} ,$$

then

$$\int_{za}^z \frac{V^2(z, t)}{A^3(z, t)} \partial A(z, t) = -\frac{V^2(z, t)}{2A^2(z, t)} \Big|_{za}^z + \int_{za}^z \frac{V(z, t)}{A(z, t)} \partial z$$

$$= -\frac{V^2(z, t)}{2A^2(z, t)} + \frac{V^2(za^+)}{2A^2(za^+)} + \int_{za}^z \frac{V(z, t)}{A(z, t)} \partial z ,$$

and letting

$$v = V(z, t) ,$$

$$du = \frac{\partial A(z, t)}{A^3(z, t)} ,$$

$$dv = A(z, t)\partial z ,$$

$$u = -\frac{1}{2A^2(z, t)} ,$$

then

$$\int_{za}^z \frac{V(z, t)}{A^3(z, t)} \partial A(z, t) = -\frac{V(z, t)}{2A^2(z, t)} \Big|_{za}^z + \int_{za}^z \frac{\partial z}{2A(z, t)}$$

$$= -\frac{V(z, t)}{2A^2(z, t)} + \frac{V(za^*)}{2A^2(za^*)} + \int_{za^*}^z \frac{\partial z}{\partial A(z, t)}.$$

Also,

$$\int_{za^*}^z \frac{\partial A(z, t)}{A^3(z, t)} = -\frac{1}{2A^2(z, t)} \Big|_{za^*}^z = \frac{1}{2} \left(-\frac{1}{A^2(z, t)} + \frac{1}{A^2(za^*)} \right).$$

Therefore, equation (16) becomes

$$\begin{aligned} \int_{za^*}^z \partial P = & -\frac{C\dot{V}_P A_B}{V^2(zp)} \int_{za^*}^z \frac{V(z, t)}{A(z, t)} \partial z + \frac{CA_A \dot{V}_P}{V(zp)} \int_{za^*}^z \frac{\partial z}{A(z, t)} \\ & + \frac{CV_P^2 A_B^2}{V^3(zp)} \left[-\frac{V^2(z, t)}{2A^2(z, t)} + \frac{V^2(za^*)}{2A^2(za^*)} + \int_{za^*}^z \frac{V(z, t)}{A(z, t)} \partial z \right] \\ & - \frac{2CV_P^2 A_A A_B}{V^2(zp)} \left[-\frac{V(z, t)}{2A^2(z, t)} + \frac{V(za^*)}{2A^2(za^*)} + \frac{1}{2} \int_{za^*}^z \frac{\partial z}{A(z, t)} \right] \\ & + \frac{CA_A^2 V_P^2}{V(zp)} \left[\frac{1}{2} \left(-\frac{1}{A^2(z, t)} + \frac{1}{A^2(za^*)} \right) \right]. \end{aligned}$$

Using the definition for $\partial V(z, t)/\partial z$, and $P(za)$, $\partial V(z, t)/\partial z$ becomes

$$\dot{V}_P = \frac{A_A z_{a0} P_{Br} + A_A z_{a1} + A_A z_{a2} P_B + P_B A_{BA} - A_B P_{res}}{m_P}$$

$$\begin{aligned} \int_{za^*}^z \partial P = & -\frac{C\dot{V}_P A_B}{V^2(zp)} \int_{za^*}^z \frac{V(z, t)}{A(z, t)} \partial z + \frac{CA_A \dot{V}_P}{V(zp)} \int_{za^*}^z \frac{\partial z}{A(z, t)} \\ & - \frac{CV_P^2 A_B^2}{2V^3(zp)} \frac{V^2(z, t)}{A^2(z, t)} + \frac{CV_P^2 A_B^2}{2V^3(zp)} \frac{V^2(za^*)}{A^2(za^*)} \\ & + \frac{CV_P^2 A_B^2}{V^3(zp)} \int_{za^*}^z \frac{V(z, t)}{A(z, t)} \partial z + \frac{CV_P^2 A_A A_B}{V^2(zp)} \frac{V(z, t)}{A^2(z, t)} - \frac{CV_P^2 A_A A_B}{V^2(zp)} \frac{V(za^*)}{A^2(za^*)} \\ & - \frac{CV_P^2 A_A A_B}{V^2(zp)} \int_{za^*}^z \frac{\partial z}{A(z, t)} - \frac{CA_A^2 V_P^2}{2V(zp) A^2} + \frac{CA_A^2 V_P^2}{2V(zp) A^2(za^*)}. \end{aligned}$$

Therefore, $P_2(z)$ is

$$\begin{aligned}
P_2(z) = & P(za') - \frac{CA_B A_A z_{a0} P_{Br}}{V^2(zp) m_p} Q_1(za') - \frac{CA_B A_A z_{a1}}{V^2(zp) m_p} Q_1(za') \\
& - \frac{CA_B A_A z_{a2} P_B}{V^2(zp) m_p} Q_1(za') - \frac{CA_B A_{B1} P_B}{V^2(zp) m_p} Q_1(za') + \frac{CA_B^2 P_{res}}{V^2(zp) m_p} Q_1(za') \\
& + \frac{CA_A^2 z_{a0} P_{Br}}{V(zp) m_p} Q_3(za') + \frac{CA_A^2 z_{a1}}{V(zp) m_p} Q_3(za') + \frac{CA_A^2 z_{a2} P_B}{V(zp) m_p} Q_3(za') \\
& + \frac{CA_A A_{B1} P_B}{V(zp) m_p} Q_3(za') - \frac{CA_A A_B P_{res}}{V(zp) m_p} Q_3(za') \\
& - \frac{CV_p^2 A_B^2}{2V^3(zp)} \frac{V(z, t)}{A^2(z, t)} + \frac{CV_p^2 A_B^2}{2V^3(zp)} \frac{V^2(za')}{A^2(za')} + \frac{CV_p^2 A_B^2}{V^3(zp)} Q_1(za) \\
& + \frac{CV_p^2 A_A A_B}{V^2(zp)} \frac{V(z, t)}{A^2(z, t)} - \frac{CV_p^2 A_A A_B}{V^2(zp)} \frac{V(za')}{A^2(za')} - \frac{CV_p^2 A_A A_B}{V^2(zp)} \int_{za'}^z \frac{\partial z}{A(z, t)} \\
& - \frac{CA_A^2 V_p^2}{2V(zp) A^2(z, t)} + \frac{CA_A^2 V_p^2}{2V(zp) A^2(za')} .
\end{aligned}$$

Using the following substitutions,

$$\begin{aligned}
Q_1(za') &= \int_{za'}^z \frac{V(z, t)}{A(z, t)} \partial z , \\
Q_3(za') &= \int_{za'}^z \frac{\partial z}{A(z, t)} ,
\end{aligned}$$

the equation for $P_2(z)$ becomes

$$\begin{aligned}
P_2(z) = & P(za^*) + a_3(t) Q_1(za^*) + a_4(t) P_B Q_1(za^*) \\
& + a_5(t) P_{Br} Q_1(za^*) + C_3(t) Q_3(za^*) + C_4(t) P_B Q_3(za^*) \\
& + C_5 P_{Br} Q_3(za^*) + \frac{CV_P^2 A_B^2 V^2(za^*)}{2V^3(zp) A^2(za^*)} - \frac{CV_P^2 A_B^2}{2V^3(zp)} Q_2(zp) \\
& + \frac{CV_P^2 A_A A_B}{V^2(zp)} Q_4(zp) - \frac{CV_P^2 A_A A_B}{V^2(zp) A^2(za^*)} \frac{V(za^*)}{A^2(za^*)} - \frac{CA_A^2}{2V(zp)} Q_5(zp) \\
& + \frac{CA_A^2 V_P^2}{2V(zp) A^2(za^*)} ,
\end{aligned}$$

where

$$C_3(t) = \frac{CA_A^2 z_{a1}}{V(zp) m_p} - \frac{CA_A A_B P_{res}}{V(zp) m_p} - \frac{CV_P^2 A_A A_B}{V^2(zp)} ,$$

$$C_4(t) = \frac{CA_A A_{BA}}{V(zp) m_p} + \frac{CA_A^2 z_{a2}}{V(zp) m_p} ,$$

$$C_5(t) = \frac{CA_A^2 z_{a0}}{V(zp) m_p} ,$$

$$Q_2(zp) = \frac{V^2(z, t)}{A^2(z, t)} ,$$

$$Q_4(zp) = \frac{V(z, t)}{A^2(z, t)} ,$$

$$Q_5(zp) = \frac{V_P^2}{A^2(z, t)} ,$$

and a_3 , a_4 , a_5 , are the same as stated previously.

From equations (5) and (6)

$$u(z a^+) - u(z a^-) = \frac{V_p A_B V(z a^+)}{V(z p) A(z a^+)} - \frac{A_A V_p}{A(z a^+)} - \frac{V_p A_B V(z a^-)}{V(z p) A(z a^-)} ,$$

letting

$$V(z a^+) = V(z a^-) = V(z a)$$

$$= \frac{V_p A_B V(z a)}{V(z p)} \left(\frac{1}{A(z a^+)} - \frac{1}{A(z a^-)} \right) - \frac{A_A V_p}{A(z a^+)} ,$$

and noting

$$A_A = \frac{A(z a^-) - A(z a^+)}{A(z a^+) A(z a^-)} .$$

With the Kooker analysis,

fk = jump in pressure across the boattail.

An analysis by Kooker (April 1991) indicates that the pressure drop across the boattail (which in a one-dimensional analysis is equivalent to determining the pressure drop across a moving discontinuity in area, Figure A-2) is given by:

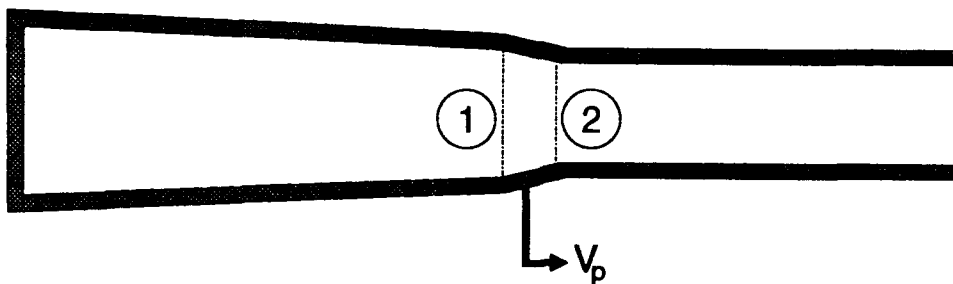


Figure A-2. Presentation of a moving area in tube.

Mass balance:

$$\rho_1(u_1 - V_p)A_1 = \rho_2(u_2 - V_p)A_2 = \dot{m} \quad (17)$$

Momentum balance:

$$\begin{aligned} \rho_2 u_2(u_2 - V_p)A_2 - \rho_1 u_1(u_1 - V_p)A_1 &= P_1 A_1 - P_2 A_2 + P_{mean}(A_2 - A_1) \\ &= -(P_2 - P_1) \left[\frac{A_1 + A_2}{2} \right] \end{aligned} \quad (18)$$

since

$$P_{mean} = \frac{1}{2}(P_1 + P_2).$$

Thus,

$$(P_2 - P_1) = \frac{2\rho(u_1 - V_p)A_1(u_2 - u_1)}{A_1 + A_2}, \quad (19)$$

where

$$A_1 = A(za^-),$$

$$A_2 = A(za^+),$$

$$\rho_1 = \rho_2 = \frac{C}{V(zp)},$$

$$u_1 = u(za^-),$$

$$u_2 = u(za^+),$$

or

$$P_2(za^+) - P_1(za^-) = fk =$$

$$\frac{\frac{-2CV_p^2 A(za^-) A_A \left[\frac{A_B V(za)}{V(zp) A(za^-)} - 1 \right]^2}{V(zp) A(za^+)}}{A(za^-) + A(za^+)}. \quad (20)$$

Therefore, $P_2(z)$ is

$$P_2(z) = P_2(za^+) + P_1(za^-) + fk$$

$$\begin{aligned} P_2(z) = & z_{a0}P_{Br} + z_{a1} + z_{a1}P_B + fk + a_3Q_1(za^+) \\ & + a_4P_BQ_1(za^+) + a_5P_{Br}Q_1(za^+) + C_3Q_3(za^+) \\ & + C_4P_BQ_3(za^+) + C_5P_{Br}Q_3(za^+) + b(t)Q_2(za^+) + h_1Q_4(za^+) \\ & + j_1Q_5(za^+) + k_1 . \end{aligned}$$

Let

$$P_B = P_2(zp)$$

$$\begin{aligned} P_B(1 - z_{a2} - a_4Q_1(zp) - C_4Q_3(zp)) = \\ P_{Br}(z_{a0} + a_5Q_1(zp) + C_5Q_3(zp)) + z_{a1} \\ + fk + a_3Q_1(zp) + C_3Q_3(zp) + b(t)Q_2(zp) + h_1Q_4(zp) \\ + j_1Q_5(zp) + k_1 \end{aligned}$$

$$P_Bl_2 = l_3P_{Br} + l_1 ,$$

where

$$\begin{aligned} l_1 = & z_{a1} + fk + a_3Q_1(zp) + C_3Q_3(zp) \\ & + b(t)Q_2(zp) + h_1Q_4(zp) + j_1Q_5(zp) + k_1 , \end{aligned}$$

$$l_2 = 1 - z_{a2} - a_4Q_1(zp) - C_4Q_3(zp) ,$$

$$l_3 = z_{a0} + a_5Q_1(zp) + C_5Q_3(zp) ,$$

$$Q_1(zp) = \int_{za^+}^{zp} \frac{V(z, t)}{A(z, t)} dz ,$$

$$Q_3(zp) = \int_{za}^{zp} \frac{\partial z}{A(z, t)} ,$$

$$Q_2(zp) = \frac{V^2(zp)}{A^2(zp)} ,$$

$$Q_4(zp) = \frac{V(zp)}{A^2(zp)} ,$$

$$Q_5(zp) = \frac{1}{A^2(zp)} ,$$

$$k_1 = \frac{-CA_B V_B^2 V(za) A_A}{V^2(zp) A^2(za)} + \frac{CA_A^2 V_P^2}{2V(zp) A^2(za)} + \frac{CV_P^2 A_B^2 V^2(za)}{2V^3(zp) A^2(za)} ,$$

$$h_1 = \frac{CA_A A_B V_P^2}{V^2(zp)} ,$$

$$j_1 = \frac{-CA_A^2 V_P^2}{2V(zp)} .$$

We are now at a point where the projectile base and breech pressure can be determined. For use in lumped-parameter interior ballistic models, the gradient equation is usually cast in terms of the mean pressure (P_m):

$$P_m = \frac{\int_0^{zp} A(z, t) P(z, t) \partial z}{\int_0^{zp} A(z, t) \partial z}$$

$$= \frac{\int_0^{za^-} A(z, t) P(z, t) \partial z + \int_{za^+}^{zp} A(z, t) P_2(z, t) \partial z}{\int_0^{za^-} A(z, t) \partial z + \int_{za^+}^{zp} A(z, t) \partial z}$$

Substituting the pressure distribution,

$$\begin{aligned}
 P_m = & \frac{\int_0^{za^-} A(z, t) P_{Br} + \int_0^{za^-} A(z, t) a_3(t) Q_1 \partial z + \int_0^{za^-} A(z, t) a_5(t) Q_1 P_{Br} \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^0 A(z, t) a_4(t) Q_1 P_B \partial z + \int_{za^-}^0 A(z, t) b(t) Q_2 \partial z + \int_{za^-}^{zp} A(z, t) z_{z0} P_{Br} \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) z_{a1} \partial z + \int_{za^-}^{zp} z_{a2} P_B \partial z + \int_{za^-}^{zp} A(z, t) f k \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) a_3 Q_1 (za^-) \partial z + \int_{za^-}^{zp} A(z, t) a_4 P_B Q_1 (za^-) \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) a_5 P_{Br} Q_1 (za^-) \partial z + \int_{za^-}^{zp} A(z, t) C_3 Q_3 (za^-) \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) C_4 P_B Q_3 (za^-) \partial z + \int_{za^-}^{zp} A(z, t) C_5 P_{Br} Q_3 (za^-) \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) b(t) Q_2 \partial z + \int_{za^-}^{zp} A(z, t) h_1 Q_4 \partial z}{V(zp)} \\
 & + \frac{\int_{za^-}^{zp} A(z, t) j_1 Q_5 \partial z + \int_{za^-}^{zp} A(z, t) k_1 \partial z}{V(zp)} .
 \end{aligned}$$

Rearranging P_m and making some substitutions, the equation becomes

$$\begin{aligned}
P_m = & \frac{a_3(t)Q_7 + b(t)Q_6 + z_{a1}(V(zp) - V(za))}{V(zp)} \\
& + \frac{a_3(t)Q_9 + fk(V(zp) - V(za)) + C_3(t)Q_8}{V(zp)} \\
& + \frac{h_1Q_1(za^*) + j_1Q_3(za) + k_1(V(zp) - V(za))}{V(zp)} \\
& + P_{Br} \frac{V(za) + a_5(t)Q_7 + a_5(t)Q_9}{V(zp)} \\
& + P_{Br} \frac{z_{a0}(V(zp) - V(za)) + C_5(t)Q_8}{V(zp)} \\
& + P_B \frac{a_4(t)Q_7 + a_4(t)Q_9 + z_{a2}(V(zp) - V(za)) + C_4(t)Q_8}{V(zp)} ,
\end{aligned}$$

where

$$Q_6 = \int_0^{zp} \frac{V^2(z, t)}{A(z, t)} \partial z ,$$

$$Q_7 = \int_0^{za} A(z, t) \int_0^z \frac{V(z, t)}{A(z, t)} \partial z \partial z ,$$

$$Q_8 = \int_{za^*}^{zp} A(z, t) \int_{za^*}^z \frac{\partial z}{A(z, t)} \partial z ,$$

$$Q_9 = \int_{za^*}^{zp} A(z, t) \int_{za^*}^z \frac{V(z, t)}{A(z, t)} \partial z \partial z ,$$

and Q_1, Q_2, Q_4 , and Q_5 are as stated previously.

Or

$$P_m = b_1 + b_2 P_B + b_3 P_{Br} ,$$

where

$$b_1 = \frac{a_3(t)Q_7(za) + a_3(t)Q_9(zp) + b(t)Q_6(zp)}{V(zp)} \\ + \frac{z_{a1}(V(zp) - V(za)) + fk(V(zp) - V(za))}{V(zp)} \\ + \frac{C_3(t)Q_8(zp) + h_1Q_1(zp) + j_1Q_3(zp)}{V(zp)} \\ + \frac{k_1(V(zp) - V(za))}{V(zp)}$$

$$b_2 = \frac{a_4(t)Q_7(za) + a_4(t)Q_9(zp)}{V(zp)} \\ + \frac{z_{a2}(V(zp) - V(za)) + C_4(t)Q_8(zp)}{V(zp)}$$

$$b_3 = \frac{V(za) + a_5(t)Q_7(za) + a_5Q_9(zp)}{V(zp)} \\ + \frac{z_{a0}(V(zp) - V(za)) + C_5Q_8(zp)}{V(zp)} .$$

Therefore,

$$\frac{P_m}{b_3} = P_{Br} + \frac{b_1}{b_3} + \frac{b_2}{b_3}P_B$$

$$P_B \frac{l_2}{l_3} = P_{Br} + \frac{l_1}{l_3} .$$

Subtracting the two previous equations and solving for P_B ,

$$\frac{P_m}{b_3} - P_B \frac{l_2}{l_3} = \frac{b_1}{b_3} - \frac{l_1}{l_3} + \frac{b_2}{b_3}P_B ,$$

$$P_B = \frac{\frac{P_m}{b_3} - \frac{b_1}{b_3} + \frac{l_1}{l_3}}{\frac{b_2}{b_3} + \frac{l_2}{l_3}} .$$

The energy of the fluid is represented by

$$dE = \frac{1}{2} u^2(z, t) dm ,$$

$$dm = \rho A(z, t) \partial z .$$

Integrating from 0 to z_p ,

$$\begin{aligned} \int_0^{z_p} dE &= \int_0^{z_p} \frac{u^2(z, t)}{2} \rho A(z, t) \partial z \\ &= \frac{\rho}{2} \int_0^{z_a^-} u^2(z, t) A(z, t) \partial z + \frac{\rho}{2} \int_{z_a^+}^{z_p} u^2(z, t) A(z, t) \partial z \end{aligned}$$

$$\begin{aligned} &= \frac{C}{2V(z_p)} \int_0^{z_a^-} \frac{A_B^2 V_p^2 V^2(z, t)}{A(z, t) V^2(z_p)} \partial z \\ &+ \frac{C}{2V(z_p)} \int_{z_a^+}^{z_p} \left(\frac{V_p A_B V(z, t)}{V(z_p) A(z, t)} - \frac{A_A V_p}{A(z, t)} \right)^2 A(z, t) \partial z \end{aligned}$$

$$\begin{aligned} \int_0^{z_p} dE &= \frac{CA_B^2 V_p^2}{2V^3(z_p)} \int_0^{z_a^-} \frac{V^2(z, t)}{A(z, t)} \partial z + \frac{CV_p^2}{2V(z_p)} \int_{z_a^+}^{z_p} \frac{A_B^2 V^2}{V^2(z_p) A(z, t)} \partial z \\ &+ \frac{CV_p^2}{2V(z_p)} \int_{z_a^+}^{z_p} \left(-\frac{2A_A A_B V(z, t)}{V(z_p) A(z, t)} + \frac{A_A^2}{A(z, t)} \right) \partial z \end{aligned}$$

$$\begin{aligned}
&= \frac{CA_B^2 V_p^2}{2V^3(zp)} \int_0^{za} \frac{V^2(z, t)}{A(z, t)} \partial z + \frac{CV_p^2 A_B^2}{2V^3(zp)} \int_{za}^{zp} \frac{V^2(z, t)}{A(z, t)} \partial z \\
&- \frac{CV_p^2 A_A A_B}{V^2(zp)} \int_{za}^{zp} \frac{V(z, t)}{A(z, t)} \partial z + \frac{CV_p^2 A_A^2}{2V(zp)} \int_{za}^{zp} \frac{\partial z}{A(z, t)} .
\end{aligned}$$

Substituting $Q_1(zp)$, $Q_3(zp)$, and Q_6 (which were previously defined) into the energy of fluid equation,

$$\int_0^{zp} dE = \frac{CA_B^2 V_p^2}{2V^3(zp)} Q_6 - \frac{CV_p^2 A_A A_B}{V^2(zp)} Q_1(zp) + \frac{CV_p^2 A_A^2}{2V(zp)} Q_3(zp) .$$

Volume and area terms for a straight tube (Lagrange tube):

$$V(za) = A_B za$$

$$V(zp) = A_B za + (zp - za) A_{BA}$$

$$Q_1(za) = \int_0^{za} \frac{V(z, t)}{A(z, t)} \partial z = \frac{za^2}{2}$$

$$Q_1(zp) = \int_{za}^{zp} \frac{V(z, t)}{A(z, t)} \partial z = \frac{(A_B za + (zp - za) A_{BA})^2}{2A_{BA}^2} - \frac{(A_B za)^2}{2A_{BA}^2}$$

$$Q_2(za) = \frac{V^2(za)}{A^2(za)} = za^2$$

$$Q_2(zp) = \frac{V^2(zp)}{A^2(zp)} = \frac{(A_B za + A_{BA}(zp - za))^2}{A_{BA}^2} - \frac{V(zp)}{A_{BA}^2}$$

$$Q_3(zp) = \int_{za}^{zp} \frac{\partial z}{A(z, t)} = \frac{zp - za}{A_{BA}}$$

$$Q_4(zp) = \frac{V(zp)}{A^2(zp)} = \frac{V(zp)}{A_{BA}^2}$$

$$Q_5(zp) = \frac{1}{A^2(z, t)} = \frac{1}{A_{BA}^2}$$

$$\begin{aligned} Q_6(zp) &= \int_0^{zp} \frac{V^2(z, t)}{A(z, t)} \partial z = \int_0^{za} \frac{A_B^2 z^2}{A_B} \partial z + \int_{za}^{zp} \frac{(A_B za + (z - za) A_{BA})^2}{A_{BA}} \\ &= \frac{A_B za^3}{3} + \frac{(A_B za + (za - za) A_{BA})^3}{3 A_{BA}^2} - \frac{(A_B za)^3}{3 A_{BA}^2} \end{aligned}$$

$$Q_7(za) = \int_0^{za} A(z, t) \int_0^z \frac{V(z, t)}{A(z, t)} \partial z \partial z = \frac{A_B za^3}{6}$$

$$Q_8(zp) = \int_{za}^{zp} A(z, t) \int_{za}^z \frac{\partial z}{A(z, t)} \partial z = \frac{(zp - za)^2}{2}$$

$$\begin{aligned} Q_9(zp) &= \int_{za}^{zp} A(z, t) \int_{za}^z \frac{V(z, t)}{A(z, t)} \partial z \partial z \\ &= \frac{(A_B za + A_{BA}(zp - za))^3}{6 A_{BA}^2} - \frac{(A_B za)^3}{6 A_{BA}^2} \\ &\quad - \frac{(A_B za)^3}{6 A_{BA}^2} - \frac{(A_B za)^2}{2 A_{BA}} (zp - za) \end{aligned}$$

The energy term for the Lagrange tube:

$$dE = \frac{1}{2} u^2(z, t) dm$$

$$dm = \rho A(z, t) \partial z$$

$$\int_0^{zp} dE = \frac{\rho}{2} \int_0^{za} u^2(z, t) A(z, t) \partial z + \frac{\rho}{2} \int_{za}^{zp} u^2(z, t) A(z, t) \partial z$$

$$= \frac{\rho}{2} \int_0^{za} \frac{A_B^2 V_P^2 V^2(z, t)}{V(zp)^2 A^2(z, t)} A(z, t) \partial z$$

$$+ \int_{za}^{zp} \left(\frac{V_P A_B V(z, t)}{V(zp) A(z, t)} - \frac{A_A V_P}{A(z, t)} \right)^2 A(z, t) \partial z$$

$$= \frac{CA_B^3 V_P^2}{2V^3(zp)} \int_0^{za} \frac{z^2 A_B^2}{A_B^2} \partial z$$

$$+ \frac{C}{2V(zp)} \int_{za}^{zp} \left(\frac{V_P A_B (A_B za + (z - za) A_{BA})}{V(zp)} - A_A V_P \right)^2 \frac{1}{A_{BA}} \partial z$$

$$= \frac{CA_B^3 V_P^2}{2V^3(zp)} \frac{za^3}{3}$$

$$+ \frac{C}{2V(zp) A_{BA}} \int_{za}^{zp} \left(\frac{V_P^2 A_B^2 V^2(z, t)}{V^2(zp)} - \frac{2A_A V_P^2 A_B V(z, t)}{V(zp)} + A_A^2 V_P^2 \right) \partial z$$

$$\begin{aligned}
&= \frac{CA_B^3 V_P^2}{2V^3(zp)} \frac{za^3}{3} \\
&+ \frac{C}{2V(zp)A_{BA}} \frac{V_P^2 A_B^2}{V^2(zp)} \int_{za}^{zp} (A_B za + (z - za)A_{BA})^2 dz \\
&- \frac{C}{2V(zp)A_{BA}} \frac{2A_A V_P^2 A_B}{V(zp)} \int_{za}^{zp} (A_B za + (z - za)) dz \\
&\quad + A_A^2 V_P^2 \int_{za}^{zp} dz
\end{aligned}$$

Substituting $Q_1(zp)$, $Q_3(zp)$, and $Q_6(zp)$ into the equation, the energy equation becomes

$$\int_0^{zp} dE = \frac{CA_B^3 V_P^2}{2V^3(zp)} Q_6(zp) - \frac{CV_P^2 A_A A_B}{V^2(zp)} Q_1(zp) + \frac{CA_A^2 V_P^2}{2V(zp)} Q_3(zp).$$

INTENTIONALLY LEFT BLANK.

APPENDIX B:
USER'S MANUAL AND CODE LISTING FOR RGA

INTENTIONALLY LEFT BLANK.

```

      program ibrga
      common nsl1, kpr, fracsl(10), dsdxsl(10), surfsl(10), nslp(10),
1  tsl(10), pbrch, pbase, pmean, bbr(10), abr(10), deltat, y(20),
1  igrad
      character cutfil * 10, bdfil * 10, style * 10
      character title(15) * 4, vsn * 4
      dimension br(10), trav(10), rp(10), tr(10), forcpl(10), temppl(10),
1  covp( 10)
      dimension chwp(10), rhop(10), gamap(10), nperfs(10), glenp(10),
1  pdp(10 ), gdiap(10), alpha(10, 10), beta(10, 10), pres(10, 10)
      dimension a(4), b(4), ak(4), d(20), p(20), z(20), frac(10),
1  surf(10 ), volp(10), dsdx(10), nbr(10), ibo(10), tbo(10),
1  d2xdt2(10), tng(10)
      real lambda, jlzp, j2zp, j3zp, j4zp, jlzb, j2zb, j3zb, j4zb
      real l1,l2,l3
      dimension chdist(6), chdiam(6), bint(10), projtr(20), projms(20)
      dimension nsl(10), surfo(10), dsdxn(10)
      data pi/3.14159/vsn/'4hboat'/

```

USER'S MANUAL FOR IBRGA

IBRGA relies on an input database consisting of all numerical parameters essential for running the code. Values may be in metric units or in Imperial units, but must be consistent throughout a dataset. Below is a compilation of a typical data base showing the name and location of each parameter. The names for the numerical values are prefixed with an alphabetical designator corresponding to the position at which the data is to appear, that is, from left to right. The data may be separated by blanks or commas. Measurement units, if any, are shown to the right of each input. In general, metric units of weight and mass are the meter and kilogram, respectively; corresponding Imperial units are the inch and pound. The only exceptions are Imperial units of propellant impetus, which are foot-pounds per pound mass.

title card - up to 60 characters of title and identification

parameter information and placement:

	A	B	C	D	E	F	G	H	I	J	K
record 1											
A. - chamber volume (when canister model is used, this will be the volume after canister bursts)								Metric (m**3)		Imperial (in**3)	
B. - groove diameter								(m)		(in)	
C. - land diameter								(m)		(in)	
D. - groove/land ratio (land, groove, and groove/land ratio used to calculate the tube bore area)											
E. - twist (units are turns/caliber)											
F. - projectile travel								(m)		(in)	
G. - gradient switch (integer value designating the gradient equation (1 = Lagrange, 2 = Chambrage,											

c 3 = Two-phase, 4 = RGA,
 c 5 = Lagrange w/ bt, 6 = Cham. w/ bt)
 c H. - variable projectile mass switch
 c (0=no, 1=yes)
 c I. - igniter canister model switch
 c (0=no, 1=yes)
 c J. - friction factor (normally 1 for
 c granular, 0.01 for stick and
 c 0.1 for partially cut propellant;
 c only used when gradient = 3 or 4)
 c
 c record 1a (read if and only if gradient = 5 or 6)
 c A. - boattail diameter (m) (in)
 c B. - boattail length (m) (in)
 c
 c record 1b (Read if and only if gradient = 2 or 4 or 6)
 c A. - number of point pairs to describe
 c chamber geometry, integer I <= 5
 c B. - initial distance from breech (m) (in)
 c (must be 0.0)
 c C. - diameter at initial distance (m) (in)
 c .
 c .
 c .
 c Ith distance from breech (m) (in)
 c (initial position of the base
 c of the projectile)
 c Ith diameter at Ith distance (m) (in)
 c (used to calculate bore area -
 c overrides record 1 groove and
 c land diameter specifications)
 c (Note: chamber geometry is used
 c to calculate the chamber volume
 c which overrides record 1 chamber
 c volume description.)
 c
 c record 2
 c A. - projectile mass (kg) (lb)
 c B. - switch to calculate energy lost
 c to air resistance, an integer
 c either 0 = no loss, or 1 = loss
 c C. - fraction of bore resistance work
 c used to heat tube (0.0<=f<=1.0)
 c D. - gas pressure ahead of projectile (MPa) (psi)
 c
 c record 2A (Read if and only if variable projectile mass
 c switch is 1)
 c A. - number of point pairs to describe
 c variable projectile mass =< 20
 c B. - initial projectile travel (m) (in)
 c (conceptually should be 0.0)
 c C. - initial projectile mass (kg) (lb)
 c (overrides value from record 2)
 c D. - projectile travel at which first (m) (in)
 c mass change occurs
 c E. - new projectile mass at travel D. (kg) (lb)
 c
 c .

c .
 c x - i-th projectile travel where mass (m) (in)
 c change occurs
 c y - i-th new projectile mass value (kg) (lb)
 c
 c record 2B (Read if and only if igniter
 c canister model switch = 1)
 c A. - pressure at which the igniter (MPa) (psi)
 c canister will burst
 c B. - volume of igniter canister (m**3) (in**3)
 c (used as chamber volume until
 c burst pressure achieved)
 c C. - canister diameter (assumes a (m) (in)
 c right circular cylinder)
 c
 c record 3
 c A. - number of pairs of barrel
 c resistance points (integer <= 10)
 c B. - bore resistance (MPa) (psi)
 c C. - travel (m) (in)
 c .
 c .
 c .
 c . - Jth bore resistance (MPa) (psi)
 c . - Jth travel (m) (in)
 c
 c record 4
 c A. - mass of recoiling parts (kg) (lb)
 c B. - number of recoil point pairs
 c (must be an integer = 2)
 c C. - recoil force (force to overcome (N) (lb)
 c before recoil start - rod preload)
 c D. - time of rod preload (must be 0.0) (s) (s)
 c E. - recoil force (constant resistive (N) (lb)
 c force after rise time)
 c F. - rise time (time to go from recoil (s) (s)
 c start to constant resistive
 c recoil force)
 c
 c record 5
 c A. - free convective heat transfer (W/m**2/K) (in-lb/in**2
 c coefficient /s/K)
 c B. - chamber wall thickness (wall (m) (in)
 c depth which is heated uniformly)
 c C. - heat capacity of chamber wall (J/kg/K) (in-lb/lb/K)
 c D. - initial temperature of tube and (K) (K)
 c chamber walls
 c E. - heat loss coefficient (usually 1.,
 c but may be set to 0.0 in order to
 c eliminate heat loss)
 c F. - density of chamber wal' (kg/m**3) (lb/in**3)
 c
 c record 6
 c A. - impetus of igniter (J/kg) (ft-lb/lb)
 c B. - adiabatic flame temperature of (K) (K)
 c igniter material
 c C. - covolume of igniter (m**3/kg) (in**3/lb)
 c D. - ratio of specific heats of igniter

```

c      E. - mass of igniter                      (kg)          (lb)
c
c      record 7
c      A. - number of propellants
c            (integer <= 10)
c
c      record 8
c      A. - impetus of propellant                 (J/kg)          (ft-lb/lb)
c      B. - adiabatic flame temperature           (K)             (K)
c      C. - covolume of propellant                (m**3/kg)       (in**3/lb)
c      D. - ratio of specific heats
c      E. - mass of propellant                     kg             (lb)
c      F. - density of propellant                 (kg/m**3)       (lb/in**3)
c      G. - propellant form function indicator
c            (integer; may be one of:
c              0 solid cylindrical grain
c              1 single-perf cylindrical grain
c              2 spherical grain
c              7 seven-perf cylindrical grain
c              15 nineteen-perf hexagonal grain
c              19 nineteen-perf cylindrical grain)
c      H. - length of propellant grain             (m)             (in)
c      I. - diameter of perforations in the        (m)             (in)
c            propellant grains (ignored if not
c            required, but must be present)
c      J. - outside diameter of propellant         (m)             (in)
c            grain (for the hexagonal grain
c            it is the distance between
c            rounded corners)
c
c      (Record 8 repeated for each propellant)
c
c      record 9
c      A. - number of burning rate triplet points
c            (integer J <= 10)
c      B. - exponent
c      C. - coefficient                           (m/s-MPa**e)    (in/s-psi**e)
c      D. - pressure (upper pressure limit         (MPa)           (psi)
c            for which the previous exponent
c            and coefficient are valid)
c
c      .
c      .
c      .
c      . - Jth exponent
c      . - Jth coefficient                       (m/s-MPa**e)    (in/s-psi**e)
c      . - Jth pressure (if pressure should        (MPa)           (psi)
c            exceed this limit, then this
c            burning rate equation is used
c            for all higher pressures)
c
c      (Record 9 repeated for each propellant)
c
c      record 10
c      A. - integration time increment             (ms)            (ms)
c      B. - print increment                       (ms)            (ms)
c      C. - upper limit on integration time        (ms)            (ms)
c            for stopping calculation
c
c*
```

```

c*      conversion factors for imperial units => metric units
c*
c      length      inches * .0254          => meters
c      mass        lb * .45359237         => kilograms
c      area        in^2 * .00064516        => m^2
c      volume      in^3 * 0.000016387064   => m^3
c      pressure    lb/in^2 * 6894.757      => pascals
c      velocity    ft/s / 3.28083         => m/s
c                      in/s * .0254        => m/s
c      energy      ft-lb * 1.3558179       => joules
c                      in-lb * 0.1129848   => joules
c      density     lb/in^3 * 27679.9       => kg/m^3
c      force/mass  (ft-lb)/lb * 2.989067   => j/kg
c      covolume    in^3/lb / 27679.9      => m^3/kg
c
c      source: engineering design handbook metric conversion guide
c              darcom pamphlet 706-470, july 1976
c
c      density     grams/cc * 1000.        => kg/m^3
c
c      call gettim(ihr,imin,ise,ihuns)
c
c      write( *, 830)
c      read( *, 840)bdfil
c      open(unit = 2, err = 810, file = bdfil, status = 'old', iostat =
1 ios)
c      nzp=0
c      rewind 2
c      write( *, 850)
c      read( *, 840)outfil
c      open(unit = 6, err = 820, file = outfil)
c      do 10 i = 1, 20
c          p(20) = 0.
c          y(20) = 0.
c          z(20) = 0.
c          d(20) = 0.
10 continue
c      write( *, 870)
c      read( *, 840 )style
c      mode = 0
c      if(style(1:1).eq.'m' .or. style(1:1).eq.'M') mode = 1
c      if(style(1:1).eq.'e' .or. style(1:1).eq.'E') mode = 2
c      if(mode.eq.0) write( *, 880)
c      if(mode.eq.0) stop
c      read(2, 885)title
c      write(6, 1236)title,vsn
c      write(6, 860)bdfil
c      read(2, *, end = 790, err = 800)cham, grve, aland, glr, twst,
1 travp, igrad, ivpm, ihl, fs0
c      if(igrad.gt.1)go to 20
c      write(6, 890)
c      igrad = 1
c      go to 140
c
c      define chambrage assumes nchpts=number of points to define
c      chamber > or = 2 < or = 5 (?),chdiam(i) defines chamber diameter
c      at chdist (i) chamber distance.  chdiam(nchpts) is assumed to be
c      the bore diameter and chdist(i) is assumed to be 0, i.e. at the

```

```

c      breech. assumes truncated cones.
c
20  if(igrad.eq.3)go to 130
    if(igrad.eq.4)go to 30
    if(igrad.GE.5)go to 25
    write(6, 900, err = 800)
    go to 40
25  read (2 , *, end = 790,err = 800) btdia,btlen
    if(mode.eq.1)then
        btvol=pi*btdia*btdia/4.*btlen
        write(6,955)btdia,btlen,btvol
955  format(/,1x,'boattail diameter m ',el6.6/1x,'boattail length
    &m ',el6.6/1x,'boattail volume m**3 ',el6.6,/)
    else
        btvol=pi*btdia*btdia/4.*btlen
        write(6,965,err=800)btdia,btlen,btvol
965  format(/,1x,'boattail diameter in ',el6.6/1x,'boattail length
    &in ',el6.6/1x,'boattail volume in**3 ',el6.6,/)
        btdia=btdia*0.0254
        btlen=btlen*0.0254
        btvol=btvol*0.0254**3
        cham=cham*1.6387064e-5
    endif
35  if(igrad.eq.5)then
    write(6,975)
975  format(1x,'using lagrange with boattail gradient')
    go to 140
    endif
    if(igrad.eq.6)then
        write(6,995)
995  format(1x,'using chambrage with boattail gradient')
        go to 40
        endif
30  write(6, 910)
    go to 40
40  read(2, *, end = 790, err = 800)nchpts, (chdist(i), chdiam(i), i
1 = 1, nchpts)
    if(mode.eq.1)then
        write(6, 920, err = 800)(chdist(i), chdiam(i), i = 1, nchpts)
        goto 60
    else
        write(6, 925, err = 800)(chdist(i), chdiam(i), i = 1, nchpts)
        do 50 i = 1, nchpts
            chdist(i) = chdist(i) * 0.0254
            chdiam(i) = chdiam(i) * 0.0254
50      continue
    endif
c
c      calculate chamber integrals and volume
c
60  if(nchpts.gt.5) write(6, 930, err = 800)
    if(nchpts.gt.5)nchpts = 5
    bore = chdiam(nchpts)
    if(chdist(1).ne.0.0)write(6, 940, err = 800)
    chdist(1) = 0.0
c      do 54 I=1,nchpts
c      chdist(I)=0.01*chdist(I)

```

```

c54  chdiam(I)=0.01*chdiam(I)
c    calculate chamber integrals and volume
    if(nchpts.gt.5) write(6,44,err=30)
44   format(1x,'use first 5 points')
    if(nchpts.gt.5)nchpts=5
    bore=chdiam(nchpts)
    if(chdist(1).ne.0.0)write(6,45,err=30)
45   format(1x,' # points ? ')
    chdist(1)=0.0
    pt1=chdist(nchpts)
    btd=btdia
    btl=btlen
    call jint(btd,btl,pt1,pt1,nchpts,chdist,chdiam,bint,bvol)
41   cham=bvol+btvol
c    write(6,47,err=30)bint(1),bint(3),bint(4)
c    format(1x,'bint 1 = ',e14.6,' bint 3 = ',e14.6,' bint 4 = ',e14.
c    &6)
    chmlen=chdist(nchpts)
    go to 140
130  write(6, 950)
140  if(mode.eq.1)then
    write(6, 960, err = 800)cham, grve, aland, glr,
& twst, travp, igrad, ivpm, ihl, fs0
    cham=cham-btvol
    endif
    if(mode.eq.2)then
    cham=cham/1.6387064e-5
    write(6, 970, err = 800)cham, grve, aland, glr, twst, travp,
1    igrad, ivpm, ihl, fs0
    cham = cham * 1.6387064e - 5
    cham=cham-btvol
    grve = grve * 0.0254
    aland = aland * 0.0254
    travp = travp * 0.0254
    endif
    read(2, *, end = 790, err = 800)prwt0, iair, htfr, pgas0
    if(mode.eq.1)then
    prwt = prwt0
    pgas = pgas0 * 1.0e6
    elseif(mode.eq.2)then
    prwt = prwt0 * 0.45359237
    pgas = pgas0 * 6894.757
    endif
    if(ivpm.eq.1)then
    read(2, *)nvpmp, (projtr(i), projms(i), i = 1, nvpmp)
    if(mode.eq.1)write(6, 980)nvpmp, (projtr(i),
1    projms(i), i = 1, nvpmp)
    if(mode.eq.2)then
    write(6, 985)nvpmp, (projtr(i), projms(i), i = 1, nvpmp)
    do 150 i = 1, nvpmp
    projtr(i) = projtr(i) * 0.0254
    projms(i) = projms(i) * 0.45359237
150    continue
    endif
    prwt = projms(1)
    prwt0 = prwt
    if(mode.eq.2) prwt0 = prwt / 0.45359237
    endif

```

```

write(6, 1050)
if(ihl.eq.1)then
  read(2, *)burstp, highv, highd
  if(mode.eq.1)write(6, 990)burstp, highv, highd
  if(mode.eq.2)then
    write(6, 1000)burstp, highv, highd
    burstp = burstp * 0.006894757
    highv = highv * 1.6387064e - 5
    highd = highd * 0.0254
  endif
  burstp = burstp * 1.e6
  areaw = 4. * highv / highd + pi * highd * highd / 2.
endif
read(2, *, end = 790, err = 800)npts, (br(i), trav(i), i = 1,npts)
read(2, *, end = 790, err = 800)rcwt, nrp, (rp(i), tr(i), i=1,nrp)
read(2, *, end = 790, err = 800)ho, tshl, cshl, twal, hl, rhocs
read(2, *, end = 790, err = 800)forcig, tempi, covi, gamai, chwi
read(2, *)nprop
read(2, *, end = 790, err = 800)(forcp(i), tempp(i), covp(i),
1 gamap(i), chwp(i), rhop(i), nperfs(i), glenp(i), pdp(i), gdiap(i)
1 , i= 1, nprop)
if(mode.eq.1)then
  write(6, 1010, err = 800)prwt0, iair, htfr, pgas0
  write(6, 1030, err = 800)npts, (br(i), trav(i), i = 1, npts)
  write(6, 1060, err = 800)rcwt, nrp, (rp(i), tr(i), i = 1, nrp)
  write(6, 1080, err = 800)ho, tshl, cshl, twal, hl, rhocs
  write(6, 1100, err = 800)forcig, tempi, covi, gamai, chwi
  write(6, 1236)title,vsu
  write(6, 1120)nprop
  write(6, 1130, err = 800)(i, forcip(i), tempp(i), covp(i),
1 gamap(i), chwp(i), rhop(i), nperfs(i), glenp(i), pdp(i),
1 gdiap(i), i = 1, nprop)
endif
if(mode.eq.2)then
  write(6, 1020, err = 800)prwt0, iair, htfr, pgas0
  write(6, 1040, err = 800)npts, (br(i), trav(i), i = 1, npts)
  write(6, 1070, err = 800)rcwt, nrp, (rp(i), tr(i), i = 1, nrp)
  write(6, 1090, err = 800)ho, tshl, cshl, twal, hl, rhocs
  write(6, 1110, err = 800)forcig, tempi, covi, gamai, chwi
  write(6, 1236)title,vsu
  write(6, 1120)nprop
  write(6, 1140, err = 800)(i, forcip(i), tempp(i), covp(i),
1 gamap(i), chwp(i), rhop(i), nperfs(i), glenp(i), pdp(i),
1 gdiap(i), i = 1, nprop)
endif
do 170 j = 1, nprop
  read(2, *,end= 790, err = 800)nbr(j), (alpha(j, i), beta(j,i)
1 , pres(j, i), i = 1, nbr(j))
  if(mode.eq.1)write(6, 1160)nbr(j)
  if(mode.eq.2)write(6, 1170)nbr(j)
  do 160 i = 1, nbr(j)
    if(mode.eq.1)write(6, 1180) alpha(j, i), beta(j, i),
1 pres(j, i)
    if(mode.eq.2)then
      write(6, 1180) alpha(j, i), beta(j, i), pres(j, i)
      rate = beta(j, i) * pres(j, i) ** alpha(j, i)
      pres(j, i) = pres(j, i) * 0.006894757
      beta(j, i) = 0.0254 * rate / pres(j, i) ** alpha(j,i)

```

```

        endif
160    continue
170    continue
c
c    convert units to program requirements
c
    do 180 i = 1, npts
        if(mode.eq.1)br(i) = br(i) * 1.e6
        if(mode.eq.2)then
            br(i) = br(i) * 6894.757
            trav(i) = trav(i) * 0.0254
        endif
180    continue
        do 200 j = 1, nprop
            if(mode.eq.2)then
                forcp(j) = forcp(j) * 2.989067
                covp(j) = covp(j) / 27679.9
                chwp(j) = chwp(j) * 0.45359237
                rhop(j) = rhop(j) * 27679.9
                glenp(j) = glenp(j) * 0.0254
                pdp(j) = pdp(j) * 0.0254
                gdiap(j) = gdiap(j) * 0.0254
            endif
            do 190 i = 1, nbr(j)
                pres(j, i) = pres(j, i) * 1.e6
190        continue
200    continue
            if(mode.eq.2)then
                do 210 i = 1, nrp
                    rp(i) = rp(i) * 0.1129848
                    tr(i) = tr(i) * 0.0254
210        continue
c
c    conversion factor for free convective heat transfer coeff
c    w/m**2 * (0.00064516 m**2/in**2) * (1.0/1.3558179 ft-lb-s/w) *
c    (12.0 in/ft) = 0.005710147 in-lb/in**2-s
c
        ho = ho / 0.005710147
        tshl = tshl * 0.0254
        cshl = cshl * 2.989067 / 12.0
        rhocs = rhocs * 27679.9
        rcwt = rcwt * 0.45359237
        forcig = forcig * 2.989067
        covi = covi / 27679.9
        chwi = chwi * 0.45359237
    endif
    tmpi = 0.0
    do 220 i = 1, nprop
        tmpi = tmpi + chwp(i)
        kpr = i
        call prf710(pdp(i), gdiap(i), glenp(i), nperfs(i), 0., frac(i)
1        , volp(i), surf(i), dsdx(i))
        tng(i) = chwp(i) / rhop(i) / volp(i)
        surfo(i) = surf(i)
        write(6, 1150)i, tng(i)
220    continue
    tmpi = tmpi + chwi
    write(6, 1050)

```

```

read(2, *, end = 790, err = 800)deltat, deltap, tstop,nzpi
write(6, 1190, err = 800)deltat, deltap, tstop
write( *, 1200)
deltat = deltat * 0.001
deltap = deltap * 0.001
tstop = tstop * .001
if(igrad.eq.2.or.igrad.eq.4.or.igrad.eq.6)go to 230
bore = (glr * grve * grve + aland * aland) / (glr + 1.)
bore = sqrt(bore)
230 areab = pi * bore * bore / 4.
areaa=pi*(btdia/2.)**2
areaba=areab-areaa
lambda = 1. / ((13.2 + 4. * log10(100. * bore)) ** 2)
iplot = 0
pltdt = deltat
pltt = 0.
pmaxm = 0.0
pmaxbr = 0.0
pmaxba = 0.0
tpmaxm = 0.0
tpmxbbr = 0.0
tpmxbba = 0.0
tpmax = 0.0
a(1) = 0.5
a(2) = 1. - sqrt(2.) / 2.
a(3) = 1. + sqrt(2.) / 2.
a(4) = 1. / 6.
b(1) = 2.
b(2) = 1.
b(3) = 1.
b(4) = 2.
ak(1) = 0.5
ak(2) = a(2)
ak(3) = a(3)
ak(4) = 0.5
vp0 = 0.0
tr0 = 0.0
tcw = 0.0
if(igrad.eq.3)chmlen = cham / areab
if(igrad.eq.5)chmlen=(cham+btvol)/areab
zb = chmlen
zp = chmlen
grlen = 0.
grdiam = 0.
egama = 0.
do 240 i = 1, nprop
    grlen = grlen + chwp(i) * glenp(i)
    grdiam = grdiam + chwp(i) * gdiap(i)
    ibo(i) = 0
    egama = egama + chwp(i) * gamap(i)
    nsl(i) = 0
    vp0 = chwp(i) / rhop(i) + vp0
240 continue
volgi = cham - vp0 - chwi * covi
grlen = grlen / (tmpi - chwi)
grdiam = grdiam / (tmpi - chwi)
egama = (egama + chwi * gamai) / tmpi
ism = 0

```

```

odlnr = 0.
vf0 = cham - vp0
eps0 = 1. - vp0 / cham
eps = eps0
gasden = chwi / vf0
prden = tmpi / vp0
ug = 0.
up = 0.
pmean = forcig * chwi / volgi
if(ihl.eq.1)pmean = forcig * chwi /
1  (highv - vp0 - chwi * covi)
pbase = pmean
pbrch = pmean
opbase = pmean
volg = volgi
volgi = volgi + vp0
wallt = twal
tgas = tempi
told = 0.
tgaso = tgas
dtgaso = 0.
covl = covi
t = 0.
ptime = 0.0
ibrp = 12
z(3) = 1.
nde = ibrp + nprop
if(mode.eq.1)write(6, 1210)areab, pmean, vp0, volgi
if(mode.eq.2)then
    arg1 = areab / 0.00064516
    arg2 = pmean / 6894.757
    arg3 = vp0 / 0.000016387064
    arg4 = volgi / 0.000016387064
    write(6, 1220)arg1, arg2, arg3, arg4
endif
write(6, 1236)title,vsn
write(6, 1230)
if(mode.eq.1)write(6, 1232)
if(mode.eq.2)write(6, 1234)
lines = 4
linmax = 62
iswl = 0
prwt0 = prwt
250 continue
do 690 j = 1, 4
c
c
c
    find barrel resistance
        if(ivpm.ne.1)go to 270
        do 260 k = 2, nvpm
            if(y(2) + y(7).lt.projtr(k))go to 270
            prwt = projms(k)
260     continue
270     if(ihl.eq.1)go to 300
        do 280 k = 2, npts
            if(y(2) + y(7).ge.trav(k))go to 280
            go to 290
280     continue

```

```

      k = npts
290      resp = (trav(k) - y(2) - y(7)) / (trav(k) - trav(k - 1))
      resp = br(k) - resp * (br(k) - br(k - 1))
c
c      find mass fraction burned
c
300      do 320 k = 1, nprop
          kpr = k
          if(ibo(k).eq.1)goto320
          nsll = 0
          call prf710(pdp(k), ydiap(k), glenp(k), nperfs(k),
1          y(ibrp + k), frac(k), volp(k), surf(k), dsdx(k))
          nsl(k) = nsll
          if(nsl(k).eq.0)goto 310
          if(nslp(k).eq.1)go to 310
          write(6, 1240)k
          lines = lines + 1
          nslp(k) = 1
          tsl(k) = y(3)
          ism = 1
310      continue
          if(frac(k).lt..9999) go to 320
          frac(k) = 1.
          tbo(k) = y(3)
          ibo(k) = 1
          ism = 1
          write(6, 1250)k
          lines = lines + 1
320      continue
          if(ihl.eq.1)goto370
c
c      energy loss to projectile translation
c
          elpt = y(11)
c
c      elpt=prwt*y(1)*y(1)/2.
c
          eptdot = prwt * y(1) * z(1)
          z(11) = eptdot
c
c      energy loss due to projectile rotation
c
          elpr = y(12)
c
c      elpr=pi*pi*prwt*((y(1)+y(6))**2)/4.*twst*twst
c
          eprdot = pi * pi * prwt * (y(1) + y(6)) * (z(1) + z(6))
1          / 2. * twst * twst
          z(12) = eprdot
c
c      energy loss due to gas and propellant motion
c
          if(igrad.eq.1)go to 340
          if(igrad.eq.3)go to 350
          if(igrad.eq.4)go to 330
          if(igrad.eq.5)go to 352
          if(igrad.eq.6)go to 355
          pt = y(2) + y(7)

```

```

      vzb = bvol + areab * pt
      j4zp = bint(4) + ((bvol + areab * pt) ** 3 - bvol ** 3) / 3.
1      / areab / areab
      elgpm = tmpi * y(1) * y(1) * areab * areab * j4zp / 2. / vzb
1      / vzb / vzb
      go to 360
330     pb = y(7) + y(10)
      vzb = bvol + areab * pb
      j4zb = bint(4) + (vzb ** 3 - bvol ** 3) / 3. / areab / areab
      elgpm = (1. - eps) * up * up * areab ** 2 * prden * j4zb + eps
1      * ug * ug * areab ** 2 * gasden * j4zb
      elgpm = elgpm / 2. / vzb / vzb + gasden * areab * ullen / 6.
1      * (3. * y(1) * y(1) + 3. * y(1) * ullen * dlnrho + ullen ** 2
1      * dlnrho ** 2)
c
c      approximate epdot
c
      epdot = tmpi * y(1) * z(1) / 3.
      go to 360
340     elgpm = tmpi * (y(1) * y(1) - y(1) * y(6) + y(6) * y(6)) / 6.
      go to 360
350     elgpm = areab * zb / 6. * (eps * gasden * ug * ug + (1. - eps)
1      * prden * up * up)
      elgpm = elgpm + gasden * areab * ullen / 6. * (3. * y(1) *
1      y(1) + 3. * y(1) * ullen * dlnrho + ullen ** 2 * dlnrho ** 2)
c
c      approximate epdot
c
      epdot = tmpi * y(1) * z(1) / 3.
      go to 360
352     zp=y(2)+y(7)+chmlen
      za=zp-btlen
      vzb=zp*areab-btvol
      vza=za*areab
c      write(6,2050)tmpi,vzb,areaa,y(1)
c      elgpm=tmpi*y(1)*y(1)/vzb/2.
c      elgpm=elgpm*(areab*areab/3./vzb/vzb*(areab*za**3
c      & +((areab*za+areaba*(zp-za))**3-(areab*za)**3)/areaba/areaba)
c      & -areaa*areab/vzb/areaba/areaba*((areab*za+areaba*(zp-za))
c      & **2-(areab*za)**2)+areaa*areaa/areaba*(zp-za))
      elgpm1=tmpi*areab**2*y(1)**2/2./vzb/vzb/vzb
      taq1=(areab*za**3/3.+
      & (areab*za+(zp-za)*areaba)**3/3./areaba/areaba-(areab*za)**3/
      & 3./areaba/areaba)
      elgpm2=tmpi*areab*areaa*y(1)**2/vzb/vzb
      taq2=((areab*za+(zp-za)
      & *areaba)**2/2./areaba/areaba-(areab*za)**2/2./areaba/areaba)
      elgpm3=tmpi*y(1)**2*areaa**2/2./vzb
      taq3=(zp-za)/areaba
      elgpm=elgpm1*taq1-elgpm2*taq2+elgpm3*taq3
c      write(6,2050)tmpi,vzb,areaa,y(1)
c2050     format(' tmpi',e17.10,' vzb',e17.10,' areaa',e17.10,' y(1)'
c      & ,e17.10)
      go to 360

```

```

355 pt=y(2)+y(7)
    pt1=pt+chmlen
    call jint(btdia,btlen,pt1,pt1,nchpts,chdist,chdiam,bint,bvolzp)

    vzp=bvolzp
    qlzp=bint(10)
    q2zp=bint(2)
    q6zp=bint(4)
    q8zp=bint(8)
    q3zp=bint(5)
    q4zp=bint(6)
    q5zp=bint(7)
    q9zp=bint(9)
c   write(6,76)qlzp,q2zp,q3zp,q4zp,q5zp,q6zp,q7zp,q8zp,pt1,pt2
76   format(1x,10e11.4)
    delta=1.
    pt1=chmlen+y(2)+y(7)
    pt2=chmlen+y(2)+y(7)-btlen
    zp=pt1
    za=pt2
    call jint(btdia,btlen,pt1,pt2,nchpts,chdist,chdiam,bint,bvolza)
    vza = bvolza
    qlza=bint(1)
    q2za=bint(2)
    q4za=bint(6)
    q5za=bint(7)
    q6za=bint(4)
    q7za=bint(3)
    q3zpza=q3zp
    q9zpza=qlzp
c   write(6,77)qlza,q2za,q3za,q4za,q5za,q6za,q7za,q8za,pt1,pt2
77   format(1x,10e11.4)
c   elgpm=tmpi*y(1)*y(1)*areab**2*q6zp/2./vzp**3 - tmpi*areaa*areab*
c   &y(1)*y(1)*q9zpza/vzp/vzp + tmpi*y(1)*y(1)*areaa**2*q3zpza/
c   &2./vzp
    elgpm1=tmpi*y(1)*y(1)*areab**2*q6zp/2./vzp**3
    elgpm2=tmpi*areaa*areab*y(1)*y(1)*qlzp/vzp/vzp
    elgpm3=tmpi*y(1)*y(1)*areaa**2*q3zp/2./vzp
    elgpm=elgpm1-elgpm2+elgpm3
    go to 360

c
c   energy loss from bore resistance
c
360   elbr = y(4)
        z(4) = areab * resp * (y(1) + y(6))
        ebrdot = z(4)

c
c   energy loss due to recoil
c
        elrc = rcwt * y(6) * y(6) / 2.
        erdot = rcwt * y(6) * z(6)

c
c   energy loss due to heat loss
c
        areaw = cham / areab * pi * bore + 2. * areab + pi * bore *
1       (y(2) + y(7))
370   avden = 0.0
        avc = 0.0

```

```

        avcp = 0.0
        z18 = 0
        z19 = 0
        do 380 k = 1, nprop
            z18 = forcp(k) * gamap(k) * chwp(k) * frac(k) / (gamap(k)
1            - 1.) / tempp(k) + z18
            z19 = chwp(k) * frac(k) + z19
            avden = avden + chwp(k) * frac(k)
380        continue
        avcp = (z18 + forcig * gamai * chwi / (gamai - 1.) / tempi) /
1        (z19 + chwi)
        avden = (avden + chwi) / (volg + covl)
        avvel = .5 * (y(1) + y(6))
        htms = lambda * avcp * avden * avvel + ho
        z(5) = areaw * htms * (tgas - wallt) * hl
        elht = y(5)
        ehdot = z(5)
        wallt = (elht + htfr * elbr) / cshl / rhocs / areaw / tshl +
1        twal
c
c        energy loss due to air resistance in tube
c        (assume no drag resistance on air/tube interface)
c
        if(ihl.eq.1)goto 410
        air = iair
        z(8) = y(1) * pgas * air
        elar = areab * y(8)
        eddot = z(8) * areab
c
c        recoil
c
        z(6) = 0.0
        if(pbrch.le.rp(1) / areab)go to 400
        rfor = rp(2)
        if(y(3) - tr0.ge.tr(2))go to 390
        rfor = (tr(2) - (y(3) - tr0)) / (tr(2) - tr(1))
        rfor = rp(2) - rfor * (rp(2) - rp(1))
390        z(6) = areab / rcwt * (pbrch - rfor / areab - resp)
        if(y(6).lt.0.0)y(6) = 0.0
        z(7) = y(6)
        goto 410
400        tr0 = y(3)
410        continue
c
c        calculate gas temperature
c
        eprop = 0.0
        rprop = 0.0
        dmfogt = 0.0
        dmfog = 0.0
        do 420 k = 1, nprop
            eprop = eprop + forcp(k) * chwp(k) * frac(k) / (gamap(k)
1            - 1.)
            rprop = rprop + forcp(k) * chwp(k) * frac(k) / (gamap(k)
1            - 1.) / tempp(k)
            dmfogt = dmfogt + forcp(k) * rhop(k) * tng(k) * surf(k) *
1            z(ibrp + k) / ((gamap(k) - 1.) * tempp(k))
            dmfog = dmfog + forcp(k) * rhop(k) * tng(k) * surf(k) *

```

```

1          z(ibrp + k) / (gamap(k) - 1.)
420      continue
          tenerg = elpt + elpr + elgpm + elbr + elrc + elht + elar
          tgas = (eprop + forcig * chwi / (gamai - 1.) - elpt - elpr -
1          elgpm - elbr - elrc - elht - elar) / (rprop + forcig * chwi
1          / (gamai - 1.) / tempi)
          tedot = epdot + eprdot + eddot + ebrdot + erdot + ehdot + eptdot
          dtgas = (dmfog - tedot - tgas * dmfogt) / (rprop + forcig *
1          chwi / (gamai - 1.) / tempi)
c
c      find free volume
c
          vl = 0.0
          covl = 0.0
          do 430 k = 1, nprop
              vl = chwp(k) * (1. - frac(k)) / rhop(k) + vl
              covl = covl + chwp(k) * covp(k) * frac(k)
430      continue
          volg = volgi + areab * (y(2) + y(7)) - vl - covl
          if(ihl.eq.1) volg = highv - vl - covl
c
c      calculate mean pressure
c
          rl = 0.0
          do 440 k = 1, nprop
              rl = rl + forcp(k) * chwp(k) * frac(k) / temp(k)
440      continue
          pmean = tgas / volg * (rl + forcig * chwi / tempi)
          if(ihl.eq.1) go to 640
          resp = resp + pgas * air
          if(igrad.eq.1) go to 590
          if(igrad.eq.2) go to 450
          if(igrad.eq.3) go to 470
          if(igrad.eq.4) go to 540
          if(igrad.eq.5) go to 582
          if(igrad.eq.6) go to 585
450      if(iswl.ne.0) go to 460
          pbase = pmean
          pbrch = pmean
          if(pbase.gt.resp + 1.) iswl = 1
          go to 620
c
c      use chambrage pressure gradient equation
c
460      jlzp = bint(1) + (bvol * pt + areab / 2. * pt * pt) / areab
          j2zp = (bvol + areab * pt) ** 2 / areab / areab
          j3zp = bint(3) + areab * bint(1) * pt + bvol * pt * pt / 2. +
1          areab * pt * pt * pt / 6.
          a2t = - tmpi * areab * areab / prwt / vzb / vzb
          alf = 1. - a2t * jlzp
          alt = tmpi * areab * (areab * y(1) * y(1) / vzb + areab * resp
1          / prwt) / vzb / vzb
          bt = - tmpi * y(1) * y(1) * areab * areab / 2. / vzb / vzb / vzb
          bata = - alt * jlzp - bt * j2zp
          gamma = alf + a2t * j3zp / vzb

```

```

        delta = bata + alt * j3zp / vzp + bt * j4zp / vzp
c
c      calculate base pressure
c
        pbase = (pmean - delta) / gamma
c
c      calculate breech pressure
c
        pbrch = alf * pbase + bata
        go to 610
c
c      use 2 phase gradient equation
c
470      if(iswl.ne.0)goto 480
        pbase = pmean
        pbrch = pmean
        if(pbase.gt.resp + 1)iswl = 1
        go to 620
480      if(iswl.eq.2)go to 580
        vzp = cham + areab * (y(2) + y(7))
        vzb = cham + areab * (y(10) + y(7))
        phi = (
        phidot = 0.
        dmorho = 0.
        dmcov = 0.
        dmromw = 0.
        rmomw = 0.
        vfree = vzp - vl
        do 490 k = 1, nprop
            rmomw = rmomw + chwp(k) * frac(k) * forcp(k) / tempp(k)
            phi = chwp(k) * frac(k) + phi
            if(ibo(k).eq.1)go to 490
            dmorho = dmorho + tng(k) * surf(k) * z(ibrp + k)
            phidot = rhop(k) * tng(k) * surf(k) * z(ibrp + k) + phidot
            dmcov = rhop(k) * tng(k) * surf(k) * z(ibrp + k) * covp(k)
            + dmcov
            dmromw = dmromw + rhop(k) * tng(k) * surf(k) * z(ibrp + k)
            * forcp(k) / tempp(k)
1          + dmcov
1          * forcp(k) / tempp(k)
490      continue
        rmomw = rmomw + chwi * forcig / tempi
        gasmas = phi + chwi
        gasden = gasmas / vfree
        phi = (phi + chwi) / tmpi
        if (phi.gt.0.999) then
            iswl = 2
            rbm = pbase / pmean
            rbrm = pbrch / pmean
            if(phi.ge.1.)go to 580
        endif
        dmdt = phidot
        phidot = phidot / tmpi
        vdotov = (dmorho + areab * y(1)) / vfree
        dlrrho = dmdt / gasmas - vdotov
        dvoldt = dmorho + areab * y(1) - dmcov
c
c      get time derivative of mean pressure
c
        dpmdt = (dmromw * tgas - pmean * dvoldt + dtgas * rmomw) /volg

```

```

volprp = 0.
effdia = 0.
dmdmdt = 0.
dmdmor = 0.
avelen = 0.
avedia = 0.
do 500 k = 1, nprop
    if(ibo(k).eq.1)go to 500
    volprp = volprp + (1. - frac(k)) * chwp(k) / rhop(k)
    dmdmdt = dmdmdt + rhop(k) * tng(k) * dsdx(k) * z(ibrp + k)
1    * z(ibrp + k)
    dmdmdt = dmdmdt + rhop(k) * tng(k) * surf(k) * d2xdt2(k)
1    dmdmor = dmdmor + (dsdx(k) * z(ibrp + k) ** 2 + surf(k) *
    d2xdt2(k)) * tng(k)
1    effdia = effdia + 6. * volp(k) / surf(k) * (1. - frac(k))
    * chwp(k)
500 continue
clt = dmdmdt / gasmas - dmdmor / vfree + vdotov ** 2 - (dmdt
1 / gasmas) ** 2
d2lnr = clt - areab ** 2 * pbase / vfree / prwt
d2lnr = d2lnr + areab * areab * resp / vfree / prwt
zp = chmlen + y(2) + y(7)
zb = chmlen + y(10) + y(7)
ullen = zp - zb
cnow = tmpi - gasmas
vp = y(1)
effdia = effdia / cnow
prden = cnow / volprp
up = y(9)
phistr = phi - gasden * areab * ullen / tmpi
ulldot = vp - up
dphist = phidot - gasden * areab / tmpi * (ulldot + ullen *
1 dlnrho)
eps = 1. - (1. - phi) * tmpi / prden / vzb
epsdot = phidot * tmpi / prden / vzb + (1. - phi) * tmpi * up
1 * areab / prden / vzb / vzb
ug = up + (vp + ullen * dlnrho - up) / eps
alam = (1.5 * grlen / grdiam) ** .666666667
alam = (0.5 + grlen / grdiam) / alam
alam = alam ** 2.17
C
C vis kg/s/m
C
vis = .00007
ren = gasden / vis * effdia * abs(ug - up)
if(ren.lt.1.)ren = 1.
fsrg = 2.5 * alam / ren ** .081 * ((1. - eps) / (1. - eps0) *
1 eps0 / eps) ** .45
fsc = fsrg * fs0
phi2 = 1. - phi - phistr * (1. - eps) / eps
philp = dphist * ug - phidot * up - phistr * epsdot / eps /
1 eps * (vp + ullen * dlnrho - up) + phistr * ulldot * dlnrho
1 / eps + 2. * phistr * ug / zb * (ug - up)
philp = philp + phi2 * gasden / effdia / prden *
1 (ug - up) ** 2 * fsc
ak2 = 1. / (1. - phi2 * tmpi / prden / vzb)

```

```

1      phil = philp + phistr * z(1) / eps + ullen * phistr * d2lnr /
      eps
c
c      acceleration of forward boundary of propellant bed
c
1      z(9) = gasden * (ug - up) ** 2 * fsc / prden / effdia + tmpi
      * phil * ak2 / vzb / prden
      z(10) = y(9)
      e = phistr / eps * (1. - ullen * areab / vfree) * areab / prwt
      dd = ullen * phistr * clt / eps
      ak11 = tmpi * e * ak2 / zb / vzb
      ak12 = tmpi * ak2 * (philp + dd) / zb / vzb - ak11 * resp
      pbase = pmean - ak12 * zb * zb / 2. + gasden * ullen * resp *
1      areab / prwt
      pbase = pbase + ak12 * zb * zb * (zb / 3. + ullen) / 2. / zp
      pbase = pbase - gasden * ullen ** 2 * areab * resp / 2. / zp
1      / prwt
      pbase = pbase - gasden * ullen ** 2 / 2. * (1. - 2. * ullen /
1      3. / zp) * (clt - dlnrho ** 2)
      pbase = pbase - areab ** 2 * gasden * ullen ** 2 * (1. - 2. *
1      ullen / 3. / zp) * resp / prwt / vfree / 2.
      deno = - ak11 * zb ** 3 / 6. / zp - ullen * ak11 * zb * zb /
1      2. / zp
      deno = deno + gasden * ullen * areab / prwt - areab ** 2 *
1      gasden * ullen ** 2 * (1. - 2. * ullen / 3. / zp) / 2. /
1      vfree / prwt
      deno = deno - gasden * ullen ** 2 * areab / 2. / zp / prwt +
1      1. + ak11 * zb * zb / 2.
      pbase = pbase / deno
      if(ism.eq.0)goto530
      if(ism.eq.1)goto510
510     goto520
      ism = 2
      tss = sqrt(egama * rmomw / gasmas * tgas)
      write(6, *)tss
      tss = ullen / (ullen * odlnr + tss)
      tso = y(3)
      write(6, *)tss, tso
520     coefbp = (tss + tso - y(3) - deltat) / tss
      if(coefbp.gt.1.)coefbp = 1.
      if(coefbp.le.0.)then
          coefbp = 0.
          ism = 0
      endif
      pbase = coefbp * opbase + (1. - coefbp) * pbase
      if(mode.eq.1)write(6, *)coefbp, opbase, pbase, ism
      if(mode.eq.2)then
          arg1 = opbase / 6894.757
          arg2 = pbase / 6894.757
          write(6, *)coefbp, arg1, arg2, ism
      endif
530     odlnr = dlnrho
      opbase = pbase
      pbrch = pbase * (1. + ak11 * zb * zb / 2. + gasden * ullen *
1      areab / prwt - areab ** 2 * gasden * ullen ** 2 / 2. / vfree
1      / prwt)

```

```

      pbrch = pbrch + ak12 * zb * zb / 2. - gasden * ullen * areab
1      * resp / prwt
      pbrch = pbrch + gasden * ullen ** 2 / 2. * (clt - dlnrho ** 2)
      pbrch = pbrch + areab ** 2 * gasden * ullen ** 2 * resp / 2.
1      / vfree / prwt
      go to 610

c
c      using rga gradient
c
540      if(iswl.ne.0)go to 550
          pbase = pmean
          pbrch = pmean
          if(pbase.gt.resp + 1.)iswl = 1
          go to 620
550      if(iswl.eq.2)go to 580
          vzp = cham + areab * (y(2) + y(7))
          vzb = cham + areab * (y(10) + y(7))
          jlzb = bint(1) + (bvol * pb + areab / 2. * pb * pb) / areab
          j2zb = (bvol + areab * pb) ** 2 / areab / areab
          j3zb = bint(3) + areab * bint(1) * pb + bvol * pb * pb / 2. +
1          areab / 6. * pb ** 3
          phi = 0.
          phidot = 0.
          dmorho = 0.
          dmcov = 0.
          dmromw = 0.
          rmomw = 0.
          vfree = vzp - vl
          do 560 k = 1, nprop
              rmomw = rmomw + chwp(k) * frac(k) * forcp(k) / tempp(k)
              phi = chwp(k) * frac(k) + phi
              if(ibo(k).eq.1)go to 560
              dmorho = dmorho + tng(k) * surf(k) * z(ibrp + k)
              phidot = rhop(k) * tng(k) * surf(k) * z(ibrp + k) + phidot
              dmcov = rhop(k) * tng(k) * surf(k) * z(ibrp + k) * covp(k)
1              + dmcov
              dmromw = dmromw + rhop(k) * tng(k) * surf(k) * z(ibrp + k)
1              * forcp(k) / tempp(k)
560      continue
          rmomw = rmomw + chwi * forcig / tempi
          gasmas = phi + chwi
          gasden = gasmas / vfree
          phi = (phi + chwi) / tempi
          if (phi.gt.0.99) then
              iswl = 2
              rbm = pbase / pmean
              rbrm = pbrch / pmean
              if(phi.ge.1.)go to 580
          endif
          dmdt = phidot
          phidot = phidot / tempi
          vdotov = (dmorho + areab * y(1)) / vfree
          dlnrho = dmdt / gasmas - vdotov
          dvoidt = dmorho + areab * y(1) - dmcov

c
c      get time derivative of mean pressure

```

c

```

dpmdt = (dmromw * tgas - pmean * dvoldt + dtgas * rmomw) / volg
volprp = 0.
effdia = 0.
dmdmdt = 0.
dmdmor = 0.
avelen = 0.
avedia = 0.
do 570 k = 1, nprop
    if(ibo(k).eq.1) go to 570
    volprp = volprp + (1. - frac(k)) * chwp(k) / rhop(k)
    dmdmdt = dmdmdt + rhop(k) * tng(k) * dsdx(k) * z(ibrp + k)
1    * z(ibrp + k)
    dmdmdt = dmdmdt + rhop(k) * tng(k) * surf(k) * d2xdt2(k)
1    dmdmor = dmdmor + (dsdx(k) * z(ibrp + k) ** 2 + surf(k) *
    d2xdt2(k)) * tng(k)
1    effdia = effdia + 6. * volp(k) / surf(k) * (1. - frac(k))
1    * chwp(k)
570 continue
clt = dmdmdt / gasmas - dmdmor / vfree + vdotov ** 2 - (dmdt
1 / gasmas) ** 2
d2lnr = clt - areab ** 2 * pbase / vfree / prwt
d2lnr = d2lnr + areab * areab * resp / vfree / prwt
zp = chmlen + y(2) + y(7)
zb = chmlen + y(10) + y(7)
ullen = zp - zb
cnow = tmpi - gasmas
vp = y(1)
effdia = effdia / cnow
prden = cnow / volprp
up = y(9)
phistr = phi - gasden * areab * ullen / tmpi
ulldot = vp - up
dphist = phidot - gasden * areab / tmpi * (ulldot + ullen *
1 dlnrho)
eps = 1. - (1. - phi) * tmpi / prden / vzb
epsdot = phidot * tmpi / prden / vzb + (1. - phi) * tmpi * up
1 * areab / prden / vzb / vzb
ug = up + (vp + ullen * dlnrho - up) / eps
alam = (1.5 * grlen / grdiam) ** .666666667
alam = (0.5 + grlen / grdiam) / alam
alam = alam ** 2.17

c
c vis kg/s/m
c
vis = .00007
ren = gasden / vis * effdia * abs(ug - up)
if(ren.lt.1.)ren = 1.
fsrg = 2.5 * alam / ren ** .081 * ((1. - eps) / (1. - eps0) *
1 eps0 / eps) ** .45
fsc = fsrg * fs0
phi2 = 1. - phi - phistr * (1. - eps) / eps
philp = dphist * ug - phidot * up - phistr * epsdot / eps /
1 eps * (vp + ullen * dlnrho - up) + phistr * ulldot * dlnrho
1 / eps + 2. * areab * phistr * ug / vzb * (ug - up)
philp = philp + phi2 * gasden / effdia / prden *

```

```

1      (ug - up) **2 * fsc
      ak2 = 1. / (1. - phi2 * tmpi / prden / vzb)
      phil = philp + phistr * z(1) / eps + ullen * phistr * d2lnr /
1      eps
c
c      acceleration of forward boundary of propellant bed
c
      z(9) = gasden * (ug - up) ** 2 * fsc / prden / effdia + tmpi
1      * phil * ak2 / vzb / prden
      z(10) = y(9)
      phi3 = phistr * ug * ug + (1. - phi) * up * up
      e = 1. - ullen * areab / vfree
      dd = ullen * phistr * clt / eps
      alt = tmpi * areab / vzb / vzb * (phi3 * areab / vzb - (philp
1      + dd - e * phistr * areab * resp / eps / prwt) * ak2)
      a2t = ( - tmpi * e * phistr * areab ** 2 / vzb / vzb / eps /
1      prwt) * ak2
      bt = - tmpi * phi3 * areab ** 2 / 2. / vzb / vzb / vzb
      pbase = pmean - gasden * ullen ** 2 / 2. * (clt - dlnrho ** 2
1      + areab ** 2 * resp / prwt / vfree) * (1. - 2. * areab *
1      ullen / 3. / vzb)
      pbase = pbase - alt * j3zb / vzb - bt * j4zb / vzb - areab *
1      ullen * alt * j1zb / vzb
      pbase = pbase - areab * bt * ullen * j2zb / vzb - gasden *
1      areab ** 2 * ullen ** 2 * resp / 2. / vzb / prwt + alt *
1      j1zb + bt * j2zb + areab * gasden * ullen * resp / prwt
      deno = 1. + areab * ullen * a2t * j1zb / vzb - gasden * areab
1      ** 2 * ullen ** 2 / 2. / vzb / prwt + a2t * j3zb / vzb - a2t
1      * j1zb + gasden * ullen * areab / prwt
      deno = deno - gasden * ullen ** 2 * areab ** 2 / 2. / vfree /
1      prwt + gasden * areab ** 3 * ullen ** 3 / 3. / vzb / vfree /
1      prwt
      pbase = pbase / deno
      pbrch = pbase * (1. - a2t * j1zb + gasden * ullen * areab /
1      prwt - gasden * ullen ** 2 * areab ** 2 / 2. / vfree / prwt)
1      + gasden * ullen ** 2 / 2. * (clt - dlnrho ** 2 + areab ** 2
1      * resp / prwt / vfree) - alt * j1zb - bt * j2zb - areab *
1      gasden * ullen * resp / prwt
      go to 610
580      pbase = rbm * pmean
      pbrch = rbrm * pmean
      go to 610
582      areazm=areab
      areaza=areazm- pi * btdia**2/4.
c      write(6,*)areazm, areaza
      delta=1.
      q1zp=(vzb**2-(areab*za)**2)/2./areaba**2
      q2zp=vzb**2/areaba/areaba
      q3zp=(zp-za)/areaba
      q4zp=vzb/areaba/areaba
      q5zp=1./areaba/areaba
      q6zp=(areab*za**3/3.+(areab*za+areaba*(zp-za))**3/3./areaba
& **2-(areab*za)**3/3./areaba**2)
      q8zp=(zp-za)**2/2.
      q9zp=(vzb**3-(areab*za)**3)/6./areaba**2
& -(areab*za)**2*(zp-za)/2./areaba
      q1za=za**2/2.

```

```

q2za=vza**2/areab/areab
q7za=areab*za**3/6.
vp=y(1)
bt= -tmpi * y(1)*y(1)*areab*areab/2./vzp/vzp/vzp
hl= tmpi*areab*areaa*y(1)*y(1)/(vzp*vzp)
akl=-tmpi*areab*vp*vp*vza*areaa/vzp/vzp/areaza**2
akl=akl+tmpi*areaa**2*vp**2/vzp/areaza**2/2.
akl=akl+tmpi*vp*vp*areab**2*vza**2/2./vzp**3/areaza**2
ajl=-tmpi*areaa**2*vp*vp/2./vzp
fk= -2.*tmpi*vp**2*areazm*areaa/areaza/vzp*
& (areab*vza/vzp/areazm-1.)**2
fk= fk/(areaza+areazm)
rrl= 1. + tmpi*areab*areaa*qlza/prwt/vzp**2
tal=tmpi*areab**2*y(1)**2/vzp**3
ta4=tmpi*areab**2*resp/prwt/vzp**2
zal= (bt*q2za + (tal+ta4)*qlza)/rrl
za0= 1./rrl
za2= -(tmpi*areab*areaba*qlza/prwt/vzp**2)/rrl
a3= tmpi*areab**2*y(1)**2/vzp**3 - tmpi*areab*areaa*zal/prwt/
& vzp**2 + tmpi*areab**2*resp/vzp**2/prwt
a41= -tmpi*areab*areaa*za2/vzp**2/prwt
a42= -tmpi*areab*areaba/vzp**2/prwt
a4 = a41+a42
c a4 = -tmpi*areab*areaa*za2/vzp**2/prwt - tmpi*areab*areaba/
c & vzp**2/prwt
a5 = - tmpi*areab*areaa*za0/prwt/vzp**2
c3 = tmpi*areaa**2*zal/prwt/vzp - tmpi*areaa*areab*
& resp/prwt/vzp - tmpi*areaa*areab*y(1)*y(1)/vzp/vzp
c41 = tmpi*areaa*areaa*za2/vzp/prwt
c42 = tmpi*areaa*areaba/vzp/prwt
c4 = c41+c42
c c4 = delta*tmpi*areaa*areaa*za2/vzp/prwt + delta*tmpi*areaa*
c & areaba/vzp/prwt
c5 = tmpi*areaa*areaa*za0/vzp/prwt
l1 = zal+fk+a3*qlzp+c3*q3zp+bt*q2zp+hl*q4zp+ajl*q5zp+akl
l2 = 1-za2-a4*qlzp - c4*q3zp
l3 = za0 + a5*qlzp +c5*q3zp
b1 = (a3*q7za+a3*q9zp+bt*q6zp+zal*(vzp-vza)+fk*(vzp-vza)
* +c3*q8zp+hl*qlzp+ajl*q3zp+akl*(vzp-vza))/vzp
b2 = (a4*q7za+a4*q9zp+za2*(vzp-vza)+c4*q8zp)/vzp
b3 = (vza+a5*q7za+a5*q9zp+za0*(vzp-vza)+c5*q8zp)/vzp
c calculate base pressure
pbase=(pmean/b3 - b1/b3)/( l2/l3 + b2/b3)
c calculate breech pressure
pbrch=pmean/b3 - b1/b3 - pbase*b2/b3
pza=zal + za2*pbase + za0*pbrch
c write(6,7)y(3),z(1),y(1),y(2),pmean,pbase,pbrch
c write(6,7) b1,b2,b3,l1,l2,l3
c write(6,7) a3,a4,a5,c3,c4,c5
z(1)=(areaa*pza+areaba*pbase-areab*resp)/prwt
go to 615
585 If(za.gt.chdist(nchpts))goto 275
Do 269 I=2,nchpts
If(za.lt.chdist(I).and.za.gt.(chdist(I-1)))goto 274
269 continue
274 diam =( za - chdist(I-1)) / ( chdist(I) - chdist(I-1)) *
& (chdiam(I) - chdiam(I-1)) + chdiam(I-1)
areazm = pi * diam**2/4.

```

```

        areaza = areazm - pi * btdia**2/4.
275  continue
      vp=y(1)
      bt= -tmpi * y(1)*y(1)*areab*areab/2./vzp/vzp/vzp
      hl= tmpi*areab*areaa*y(1)*y(1)/(vzp*vzp)
      ak1=-tmpi*areab*vp*vp*vza*areaa/vzp/vzp/areaza**2
      ak1=ak1+tmpi*areaa**2*vp**2/vzp/areaza**2/2.
      ak1=ak1+tmpi*vp*vp*areab**2*vza**2/2./vzp**3/areaza**2
      aj1=-tmpi*areaa**2*vp*vp/2./vzp
      fk= -2.*tmpi*vp**2*areazm*areaa/areaza/vzp*
      & (areab*vza/vzp/areazm-1.)**2
      fk= fk/(areaza+areazm)
      rrl= 1. + tmpi*areab*areaa*qlza/prwt/vzp**2
      tal=tmpi*areab**2*y(1)**2/vzp**3
      ta4=tmpi*areab**2*resp/prwt/vzp**2
      zal= (bt*q2za + (tal+ta4)*qlza)/rrl
      za0= 1./rrl
      za2= -(tmpi*areab*areaba*qlza/prwt/vzp**2)/rrl
      a3 = tmpi*areab**2*y(1)**2/vzp**3 - tmpi*areab*areaa*zal/prwt/
      & vzp**2 + tmpi*areab**2*resp/vzp**2/prwt
      a41= -tmpi*areab*areaa*za2/vzp**2/prwt
      a42= -tmpi*areab*areaba/vzp**2/prwt
      a4 = a41+a42
c    a4 = -tmpi*areab*areaa*za2/vzp**2/prwt - tmpi*areab*areaba/
c    & vzp**2/prwt
      a5 = - tmpi*areab*areaa*za0/prwt/vzp**2
      c3 = tmpi*areaa**2*zal/prwt/vzp - tmpi*areaa*areab*
      & resp/prwt/vzp - tmpi*areaa*areab*y(1)*y(1)/vzp/vzp
      c41 = tmpi*areaa*areaa*za2/vzp/prwt
      c42 = tmpi*areaa*areaba/vzp/prwt
      c4 = c41+c42
c    c4 = delta*tmpi*areaa*areaa*za2/vzp/prwt + delta*tmpi*areaa*
c    & areaba/vzp/prwt
      c5 = tmpi*areaa*areaa*za0/vzp/prwt
      l1 = zal+fk+a3*qlzp+c3*q3zp+bt*q2zp+hl*q4zp+aj1*q5zp+ak1
      l2 = 1-za2-a4*qlzp - c4*q3zp
      l3 = za0 + a5*qlzp +c5*q3zp
      b1 = (a3*q7za+a3*q9zp+bt*q6zp+zal*(vzp-vza)+fk*(vzp-vza)
      * +c3*q8zp+hl*qlzp+aj1*q3zp+ak1*(vzp-vza))/vzp
      b2 = (a4*q7za+a4*q9zp+za2*(vzp-vza)+c4*q8zp)/vzp
      b3 = (vza+a5*q7za+a5*q9zp+za0*(vzp-vza)+c5*q8zp)/vzp
c    calculate base pressure
      pbase=(pmean/b3 - b1/b3 + l1/l3)/( l2/l3 + b2/b3)
c    calculate breech pressure
      pbrch=pmean/b3 - b1/b3 - pbase*b2/b3
      pza=zal + za2*pbase + za0*pbrch
c    write(6,7) y(3), z(1), y(1), y(2), pmean, pbase, pbrch
c    write(6,7) b1, b2, b3, l1, l2, l3
c    write(6,7) a3, a4, a5, c3, c4, c5
      z(1)=(areaa*pza+areaba*pbase-areab*resp)/prwt
      go to 615
c    use lagrange pressure gradient equation
c
590    if(iswl.ne.0) go to 600
        if(pmean.lt.resp) resp = pmean
c
c    calculate base pressure
c

```

```

600      tmp2 = 1.0 + tmpi / 2.0 / prwt
      tmr2 = 1.0 + tmpi / 2.0 / rcwt
      tmr3 = 1.0 + tmpi / 3.0 / rcwt
      tmr4 = rfor / areab + resp - pgas * air
      pbase = pmean / tmr3
1      - tmpi / 2.0 / tmr2 * ( tmr4 / rcwt - resp / prwt)
2      + tmpi / 3.0 / tmr3 * ( tmr4 / rcwt - resp / prwt / 2.0)
      pbase = pbase / ( tmp2 / tmr2 - tmpi / tmr3 / prwt / 6.0)

c
      if(pbase.gt.resp + 1.)iswl = 1

c
c      calculate breech pessure
c
      pbrch = pbase * tmp2 / tmr2 + tmpi / 2.0 / tmr2 *
1      (tmr4 / rcwt - resp / prwt)

c
c      calculate projectile acceleration
c
610      z(1) = areab * (pbase - resp) / prwt
615      if(z(1).lt.0.0)go to 620
      go to 630
620      if(iswl.eq.0)z(1) = 0.0
630      if(y(1).lt.0.0)y(1) = 0.0
      z(2) = y(1)

c
c      get burning rate
c
640      do 670 m = 1, nprop
      z(ibrp + m) = 0.0
      d2xdt2(m) = 0.0
      if(ibo(m).eq.1) goto 670
      do 650 k = 1, nbr(m)
      if(pmean.gt.pres(m, k))go to 650
      go to 660
650      continue
      k = nbr(m)
660      pmix = pmean
      if(igrad.eq.3)pmix = pbrch - (ak11 * pbase + ak12) / 6. *
1      zb * zb
      if(igrad.eq.4)pmix = pbrch + (alt + a2t * pbase) * j3zb /
1      vzb + bt * j4zb / vzb
      if(pmix.lt..99 * pmean)pmix = pmean
      z(ibrp + m) = beta(m, k) * (pmix * 1.e - 6) ** alpha(m, k)
      abr(m) = alpha(m, k)
      bbr(m) = beta(m, k)
      d2xdt2(m) = beta(m, k) * alpha(m, k) * (pmix * 1.e - 6) **
1      (alpha(m, k) - 1.) * dpmdt * 1.e - 6
670      continue
      do 680 i = 1, nde
      d(i) = (z(i) - b(j) * p(i)) * a(j)
      y(i) = deltat * d(i) + y(i)
      p(i) = 3. * d(i) - ak(j) * z(i) + p(i)
680      continue
690      continue
      nzp=nzp+1
      if(igrad.ne.6)goto 2003
      if(nzp.ne.nzpi)goto 2003
      dzp=zip/50.

```

```

open(unit=11,file='dist.dat')
open(unit=12,file='press.dat')
do 2001 i=1,50
  ddzp=i*dzp
  call jint(btd,btl,zp,ddzp,nchpts,chdist,chdiam,bint,bvol)
  if(ddzp.lt.za)then
    pz=pbrch-tmpi*areab*z(1)*bint(1)/vzp/vzp
    & +tal*bint(1)+bt*bint(2)
    pz1=pbrch+(a3+a5*pbrch+a4*pbase)*bint(1)+bt*bint(2)
  else
    pz=pza +fk-tmpi*areab*z(1)*bint(10)/vzp**2
    & +tmpi*areaa*z(1)*bint(5)/vzp+bt*bint(2)+ak1
    & +h1*bint(6)-bt*2.*bint(10)-h1*bint(5)+aj1*bint(7)
    pz1=za0*pbrch+za1+za2*pbase+fk+a3*bint(10)
    & +a4*pbase*bint(10)+a5*pbrch*bint(10)
    & +c3*bint(5)+c4*pbase*bint(5)+c5*pbrch*bint(5)
    & +bt*bint(2)+h1*bint(6)+aj1*bint(7)+ak1
  endif
  write(6,*)'za ',za,' ddzp ',ddzp,' pza ',pza
  write(6,*)' pz ',pz,' pz1 ',pz1
  write(11,*)ddzp
  write(12,*)pz/1.e6
2001 continue
2003 if(prwt0.ne.prwt)then
  if(mode.eq.1)write(6, 1450)prwt
  if(mode.eq.2)then
    arg1 = prwt / 0.45359237
    write(6, 1450)arg1
  endif
  prwt0 = prwt
  lines = lines + 1
endif
t = t + deltat
told = y(3)
if(ihl.eq.1 .and. pmean.gt.burstp)then
  write(6, 1440)
  ihl = 2
  lines = lines + 1
endif
if(pmaxm.gt.pmean)go to 700
pmaxm = pmean
tpmaxm = y(3)
700 if(pmaxba.gt.pbase)go to 710
pmaxba = pbase
tpmaxba = y(3)
710 if(pmaxbr.gt.pbrch)go to 720
pmaxbr = pbrch
tpmaxbr = y(3)
720 continue
if(y(3).lt.ptime)go to 730
ptime = ptime + deltap
pjt = y(2) + y(7)
arg0 = y(3) * 1000.
if(mode.eq.1)write(6, 1270)arg0, z(1), y(1), pj, pmean, pbase,
1 pbrch
if(nzp.ne.nzpi)goto 2004
write(6,*)'dpza ',dpza,' pza ',pza,' dl ',dl,' gl ',gl
c write(6,*)areaa*pza,areaba*pbase,areaa*pza+areaba*pbase

```

```

write(6,*)'areaa ','areaa,' areab ','areab,' areaba ','areaba
write(6,*)' vza ','vza,' vzp ','vzp
write(6,*)'qlza-ql0za'
write(6,*)qlza,q2za,q3za,q4za
write(6,*)q5za,q6za,q7za,q8za
write(6,*)q9za,ql0za
write(6,*)' qlzp-q9zp'
write(6,*)qlzp,q2zp,q3zp,q4zp
write(6,*)q5zp,q6zp,q7zp,q8zp
write(6,*)q9zp
write(6,*)
write(6,*)'zao,za1,za2,resp ','zao,za1,za2,resp
write(6,*)'b1,b2,b3,l1 ','b1,b2,b3,l1
write(6,*)'l2,l3,a3,a4 ','l2,l3,a3,a4
write(6,*)' a5,c3,c4,c5 ','a5,c3,c4,c5
write(6,*)'bt,h1,ak1,aj1,fk ','bt,h1,ak1,aj1,fk
2004 if(mode.eq.2)then
      arg1 = y(1) / 0.0254
      arg2 = pjt / 0.0254
      arg3 = pmean / 6894.757
      arg4 = pbase / 6894.757
      arg5 = pbrch / 6894.757
      arg6 = z(1) / 0.0254
      write(6, 1270)arg0, arg6, arg1, arg2, arg3, arg4, arg5
endif
lines = lines + 1
if(igrad.gt.2)then
  pjt = y(2) + y(7)
  prt = y(10) + y(7)
  if(mode.eq.1)write(6, 1280)prt, pjt
  if(mode.eq.2)then
    arg1 = prt / 3.28083
    arg2 = pjt / 3.28083
    write(6, 1280)arg1, arg2
  endif
  lines = lines + 1
endif
730 continue
if(lines.gt.linmax)then
  write(6, 1236) title,vsn
  write(6, 1230)
  if(mode.eq.1)write(6, 1232)
  if(mode.eq.2)write(6, 1234)
  lines = 4
endif
if(t.gt.tstop)goto 740
if(y(2) + y(7).gt.travp)go to 740
rmvelo = y(1)
tmvelo = y(3)
rcvelo = y(6)
disto = y(2) + y(7)
go to 250
740 if(lines.gt.linmax-nprop-25) write(6, 1236) title,vsn
write(6, 1290)t, y(3)
if(mode.eq.1) write(6, 1300)maxm, tpmaxm, pmaxba, tpmxba, pmaxbr,
1 tpmxbr
if(mode.eq.2)then
  pmaxm = pmaxm / 6894.757

```

```

        pmaxba = pmaxba / 6894.757
        pmaxbr = pmaxbr / 6894.757
        write(6, 1310)pmaxm, tpmaxm, pmaxba, tpmxba, pmaxbr, tpmxbr
    endif
    if(y(2) + y(7).le.travp)goto 750
    dfraction = (travp - disto) / (y(2) + y(7) - disto)
    rmvel = (y(1) - rmvelo) * dfraction + rmvelo
    tmvel = (y(3) - tmvelo) * dfraction + tmvelo
    rcvel = (y(6) - rcvelo) * dfraction + rcvelo
    if(mode.eq.1)write(6, 1320)rmvel, tmvel, rcvel
    if(mode.eq.2)then
        rmvel = rmvel * 3.28083
        rcvel = rcvel * 3.28083
        write(6, 1330)rmvel, tmvel, rcvel
    endif
    go to 760
750 if(mode.eq.1)write(6, 1340)y(1), y(3)
    if(mode.eq.2)then
        arg1 = y(1) * 3.28083
        write(6, 1350)arg1, y(3)
    endif
760 efi = chwi * forcig / (gamai - 1.)
    efp = 0.0
    do 770 i = 1, nprop
        efp = efp + chwp(i) * forcp(i) / (gamap(i) - 1.0)
770 continue
    tenerg = efi + efp
    if(mode.eq.1)write(6, 1360)tenerg
    if(mode.eq.2)tenerg = tenerg / 0.1129848
    if(mode.eq.2)write(6, 1370)tenerg
    tengas = chwi * forcig * tgas / (gamai - 1.) / tempi
    do 780 i = 1, nprop
        tengas = (frac(i) * chwp(i) * forcp(i) * tgas / tempp(i) /
1      (gamap(i) - 1.)) + tengas
780 continue
    write(6, 1380)(i, frac(i), tbo(i), i = 1, nprop)
    if(mode.eq.1)write(6, 1390)
    if(mode.eq.2)then
        tengas = tengas / 0.1129848
        elpt = elpt / 0.1129848
        elpr = elpr / 0.1129848
        elgpm = elgpm / 0.1129848
        elbr = elbr / 0.1129848
        elrc = elrc / 0.1129848
        elht = elht / 0.1129848
        elar = elar / 0.1129848
        write(6, 1400)
    endif
    pcten1 = tengas / tenerg * 100.0
    pcten2 = elpt / tenerg * 100.0
    pcten3 = elpr / tenerg * 100.0
    pcten4 = elgpm / tenerg * 100.0
    pcten5 = elbr / tenerg * 100.0
    pcten6 = elrc / tenerg * 100.0
    pcten7 = elht / tenerg * 100.0
    pcten8 = elar / tenerg * 100.0
    write(6, 1410)tengas, pcten1, elpt, pcten2, elpr, pcten3,
1    elgpm, pcten4, elbr, pcten5, elrc, pcten6, elht, pcten7,

```

```

2    elar, pten8
    stop
790 write( *, 1420)
    stop
800 write( *, 1430)
810 continue
820 continue
    stop
830 format (' input name of data file to be used as input ')
840 format (a10)
850 format (' input name of output file ')
860 format (' the input file is ',a10/)
870 format (' input data units - "metric" or "english"')
880 format (' must use quoted "m" or "e" as first character of',
1    ' input file'/' to specify metric or english input units')
885 format (15a4)
890 format (1x,'using lagrange pressure gradient')
900 format (1x,'using chambrage pressure gradient')
910 format (1x,'using rga gradient')
920 format (///,' chamber distance m chamber diameter m',/
1    (5x,e14.6,5x,e14.6))
925 format (///,' chamber distance in chamber diameter in',/
1    (5x,e14.6,6x,e14.6))
930 format (1x,'use first 5 points')
940 format (1x,' # points ? ')
950 format (1x,'using 2 phase gradient equation')
960 format (' chamber volume m**3 ',e16.6/
1    ' groove diam m ',e16.6/' land diam m ',e16.6/
1    ' groove/land ratio ',e16.6/' twist turns/caliber ',e16.6/
1    ' projectile travel m',e16.6/' gradient # ',i3,//
1    ' variable mass switch',i3/' container switch',i7/
1    ' friction factor ',e18.6/)
970 format (' chamber volume in**3 ',e16.6/
1    ' groove diam in ',e16.6/' land diam in ',e16.6/
1    ' groove/land ratio ',e16.6/' twist turns/caliber ',e16.6/
1    ' projectile travel in',e16.6/' gradient # ',i3,//
1    ' variable mass switch',i3/' container switch',i7/
1    ' friction factor ',e18.6/)
980 format (1x,'number of variable projectile mass points ',i2,/
1    1x,' travel (m) projectile mass (kg)'/
1    (1x,e14.6,e14.6))
985 format (1x,'number of variable projectile mass points ',i2,/
1    1x,' travel (in) projectile mass (lb)'/
1    (1x,e14.6,e14.6))
990 format (1x,'canister burst pressure (mpa)',e14.6/
1    1x,'canister volume (m^3) ',e14.6/
1    1x,'canister diameter (m) ',e14.6)
1000 format (1x,'canister burst pressure (psi)',e14.6/
1    1x,'canister volume (in^3) ',e14.6/
1    1x,'canister diameter (in) ',e14.6)
1010 format (' projectile mass kg',34x,e14.6/
1    ' switch to calculate energy lost to air resistance ',i3/
1    ' fraction of work against bore used to heat the tube',e14.6/
1    ' gas pressure mpa ',e14.6)
1020 format (' projectile mass lb',34x,e14.6/
1    ' switch to calculate energy lost to air resistance ',i3/
1    ' fraction of work against bore used to heat the tube',e14.6/
1    ' gas pressure psi ',e14.6)

```

```

1030 format (' number barrel resistance points ',i2/
1      ' bore resistance mpa - travel m'/(3x,e14.6,8x,e14.6))
1040 format (' number barrel resistance points ',i2/
1      ' bore resistance psi - travel inches'/(3x,e14.6,e22.6))
1050 format (1x)
1060 format (/
1      ' mass of recoiling parts kg ',e14.6/
1      ' number of recoil point pairs ',i3/
1      ' recoil force J', ' recoil time sec'/(1x,e14.6,3x,e14.6))
1070 format (/
1      ' mass of recoiling parts lb ',e14.6/
1      ' number of recoil point pairs ',i3/
1      ' recoil force in-lb', ' recoil time sec' /
1      (1x,e14.6,3x,e14.6))
1080 format (/
1      ' free convective heat transfer coefficient w/m^2 k ',e14.6/
1      ' chamber wall thickness m ',e14.6/
1      ' heat capacity of steel of chamber wall j/kg k ',e14.6/
1      ' initial temperature of chamber wall k ',e14.6/
1      ' heat loss coefficient ',e14.6/
1      ' density of chamber wall steel kg/m^3 ',e14.6/)
1090 format (/
1      ' free convective heat transfer coef in-lb/in^2-s-k ',e14.6/
1      ' chamber wall thickness (inches) ',e14.6/
1      ' heat capacity of steel of chamber wall in-lb/lb-k ',e14.6/
1      ' initial temperature of chamber wall k ',e14.6/
1      ' heat loss coefficient ',e14.6/
1      ' density of chamber wall steel lb/in^3 ',e14.6/)
1100 format (
1      ' impetus of igniter propellant j/kg ',e14.6/
1      ' adiabatic flame temperature of igniter propellant k',e14.6/
1      ' covolume of igniter m^3/kg ',e14.6/
1      ' ratio of specific heats for igniter ',e14.6/
1      ' initial mass of igniter kg ',e14.6/)
1110 format (
1      ' impetus of igniter propellant ft-lb/lb ',e14.6/
1      ' adiabatic flame temperature of igniter propellant k',e14.6/
1      ' covolume of igniter ft^3/lb ',e14.6/
1      ' ratio of specific heats for igniter ',e14.6/
1      ' initial mass of igniter lb ',e14.6/)
1120 format (' there are ',i2,' propellants'//)
1130 format ((' for propellant number',i2//
1      ' impetus of propellant j/kg ',e14.6/
1      ' adiabatic temperature of propellant k ',e14.6/
1      ' covolume of propellant m^3/kg ',e14.6/
1      ' ratio of specific heats for propellant ',e14.6/
1      ' initial mass of propellant kg ',e14.6/
1      ' density of propellant kg/m^3 ',e14.6/
1      ' number of perforations of propellant ',i3/
1      ' length of propellant grain m ',e14.6/
1      ' perforation diameter m ',e14.6/
1      ' outside diameter of propellant grain m ',e14.6//))
1140 format ((' for propellant number',i2//
1      ' impetus of propellant ft-lb/lb ',e14.6/
1      ' adiabatic temperature of propellant k ',e14.6/
1      ' covolume of propellant in^3/lb ',e14.6/
1      ' ratio of specific heats for propellant ',e14.6/
1      ' initial mass of propellant lb ',e14.6/

```

```

1 ' density of propellant lb/in^3 ',e14.6/
1 ' number of perforations of propellant ',i3/
1 ' length of propellant grain in ',e14.6/
1 ' perforation diameter in ',e14.6/
1 ' outside diameter of propellant grain in',e14.6/))
1150 format (' for propellant ',i2,' the total number of grains'
1 ', is ',e14.6)
1160 format (' number of burning rate points',i2/
1 ' exponent coefficient pressure'/
1 ' - m/sec-mpa^ai mpa')
1170 format (' number of burning rate points',i2/
1 ' exponent coefficient pressure'/
1 ' - in/sec-psi^ai psi')
1180 format (1x,e14.6,5x,e14.6,4x,e14.6)
1190 format (' time increment msec ',e14.6/
1 ' print increment msec ',e14.6/
1 ' time to stop calculation msec',e14.6)
1200 format (1x,'end input data -- i.b. calculation start')
1210 format (' area bore m^2 ',e27.6/' pressure from ign pa',e21.6/
1 ', volume of unburnt prop m^3 ',e14.6/
1 ' init cham vol-cov ign m^3 ',e15.6)
1220 format (' area bore in^2 ',e29.6/' pressure from ign psi',e23.6/
1 ', volume of unburnt prop in^3 ',e16.6/
1 ' init cham vol-cov ign in^3 ',e17.6)
1230 format (' time accel velocity distance pr mean',
1 ' pr base pr brch')
1232 format (' (ms) (m/s/s) (m/s) (m) (Pa) ',
1 ' (Pa) (Pa)')
1234 format (' (ms) (in/s/s) (in/s) (in) (psi) ',
1 ' (psi) (psi)')
1236 format (1h1,3x,15a4,' rga.',a4)
1240 format (' propellant',i2,' has slivered')
1250 format (' propellant',i2,' has burned out')
1270 format (1x,1p7e11.4)
1280 format (1x,'prop travel',e11.4,'proj travel',e11.4)
1290 format (/1x,' deltat t', e14.6, ' intg t',e14.6)
1300 format (' pmaxmean pa ',1pe14.7,' time at pmaxmean sec ',
1 0pe16.6/' pmaxbase pa ',1pe14.7,' time at pmaxbase sec ',
1 0pe16.6/' pmaxbreech pa ',1pe14.7,' time at pmaxbreech sec ',
1 0pe14.6)
1310 format (' pmaxmean psi',f14.3,' time at pmaxmean sec ',
1 0pe16.6/' pmaxbase psi',f14.3,' time at pmaxbase sec ',
1 0pe16.6/' pmaxbreech psi',f14.3,' time at pmaxbreech sec ',
1 0pe14.6)
1320 format (/1x,'muzzle velocity m/s ',e14.6,' time of muzzle exit ',
1 e14.6,' sec'//1x,'recoil velocity m/s ',e14.6)
1330 format (/1x,'muzzle velocity ft/s ',e14.6,' time of muzzle exit ',
1 e14.6,' sec'//1x,'recoil velocity ft/s ',e14.6)
1340 format (/ ' velocity of projectile m/s ',e14.6,' at time ',e14.6,
1 ' sec')
1350 format (/ ' velocity of projectile ft/s ',e14.6,' at time ',e14.6,
1 ' sec')
1360 format (/1x,'total initial energy available j = ',e14.6/)
1370 format (/1x,'total initial energy available in-lb = ',e14.6/)
1380 format (' propellant mass fraction burnt time (sec)'/
1 (4x,i2,9x,e14.6,5x,e14.6))
1390 format (' ** energy summary **',23x,'joules',11x,'')
1400 format (' ** energy summary **',23x,' in-lb',11x,'%')

```

```

1410 format (1x,'total energy remaining in gas',11x,' = ',e14.6,f11.4
1    /1x,'energy loss from projectile translation = ',e14.6,f11.4
1    /1x,'energy loss from projectile rotation   = ',e14.6,f11.4
1    /1x,'energy lost to gas and propellant motion = ',e14.6,f11.4
1    /1x,'energy lost to bore resistance         = ',e14.6,f11.4
1    /1x,'energy lost to recoil                  = ',e14.6,f11.4
1    /1x,'energy loss from heat transfer         = ',e14.6,f11.4
1    /1x,'energy lost to air resistance          = ',e14.6,f11.4)
1420 format (1x,'end of file encounter')
1430 format (1x,'read or write error')
1440 format ('    canister burst pressure achieved')
1450 format ('    projectile mass transition point - new mass = ',
1    1p11.4)
    end
    subroutine prf710(pd, gd, gl, np, x, frac, vol, surf, dsdx)
    common nsl, kpr, fracsl(10), dsdxsl(10), surfsl(10), nslp(10),
1    tsl(10), pbrch, pbase, pmean, bbr(10), abr(10), deltat, yar(20),
1    igrad
    dimension ts(10), coef(10)
    pi = 3.141593
    nsl = 0

c
c    pd=perforation diameter
c    gd=outer dia
c    gl=grain length
c    np=number of perfs
c
c    surf=output surface area
c    frac=output mass fraction of propellant burned
c
c    w = web = distance between perforation edges
c    = distance between outside perf edge and edge of grain
c
c    p = distance between perforation centers
c
c    x1 = distance to inner sliver burnout
c
c    x2 = distance to outer sliver burnout (frac=1)
c
    if(np.eq.0) go to 70
    if(np.eq.1)go to 90
    if(np.eq.2)go to 210
    if(np.eq.7)go to 10
    if(np.eq.19)go to 110
    if(np.eq.15)go to 180
    write(6, 220)
    stop
10  w = (gd - 3. * pd) / 4.
    d = w + pd
    sqr3 = sqrt(3.)
    x1 = d / sqr3 - pd / 2.
    x2 = (14. - 3. * sqr3) * d / 13. - pd / 2.
    v0 = pi / 4. * gl * (gd * gd - 7. * pd * pd)
    s0 = 2. * v0 / gl + pi * gl * (gd + 7. * pd)
    if (x.gt.w / 2. + .0000001) goto 20
    vol = pi / 4. * (gl - 2. * x) * ((gd - 2. * x) ** 2 - 7. * (pd +
1 2. * x) ** 2)
    surf = 2. * vol / (gl - 2. * x) + pi * (gl - 2. * x) * ((gd - 2.

```

```

1 * x) + 7. * (pd + 2. * x))
frac = 1. - vol / v0
dsdx = - 4 * pi * (gd + 7. * pd - 3. * gl + 18. * x)
dsdxsl(kpr) = dsdx
fracsl(kpr) = frac
surfsl(kpr) = surf
return
20 nsl = 1
coef(kpr) = 0.
if(igrad.eq.1.or.igrad.eq.2)go to 40
if(nslp(kpr).eq.1)goto 30
tsl(kpr) = yar(3)
ts(kpr) = w / 2. * ( - 1. + (pbrch / pmean) ** abr(kpr)) /
1 (bbr(kpr) * (pbase * 1.e - 6) ** abr(kpr))
30 continue
coef(kpr) = (ts(kpr) + tsl(kpr) - (deltat + yar(3))) / ts(kpr)
if(coef(kpr).gt.1.)coef(kpr) = 1.
if(coef(kpr).lt.0.)coef(kpr) = 0.
40 if(x.ge.x2)goto 60
s1 = 0.
s2 = 0.
v1 = 0.
v2 = 0.
dsldx = 0.
ds2dx = 0.
y = sqrt((pd + 2. * x) ** 2 - d * d)
theta = atan(y / d)
a1 = theta / 4. * (pd + 2. * x) ** 2 - d / 4. * y
if(x.ge.x1)goto 50
v1 = 3. / 4. * (gl - 2. * x)
v1 = v1 * (2. * sqr3 * d * d - pi * (pd + 2. * x) ** 2 + 24. * a1)
s1 = 2. * v1 / (gl - 2. * x)
s1 = s1 + 3. * (gl - 2. * x) * (pi - 6. * theta) * (pd + 2. * x)
50 y1 = sqrt((gd - 2. * x) ** 2 - (5. * d - 2. * (pd + 2. * x)) ** 2)
chi = atan(y1 / (5. * d - 2. * (pd + 2. * x)))
y2 = sqrt((pd + 2. * x) ** 2 - (3. * d - 2. * (pd + 2. * x)) ** 2)
phi = atan(y2 / (3. * d - 2. * (pd + 2. * x)))
a2 = phi * (pd + 2. * x) ** 2 - chi * (gd - 2. * x) ** 2
a2 = (a2 + 2. * sqr3 * d * sqrt((3. * d - pd - 2. * x) * (3. * d
1 - gd + 2. * x))) / 8.
v2 = pi * (gd - 2. * x) ** 2 - 6. * sqr3 * d * d - 4. * pi * (pd
1 + 2. * x) ** 2
v2 = (v2 + 24. * (a1 + 2. * a2)) * (gl - 2. * x) / 4.
s2 = 2. * v2 / (gl - 2. * x)
s2 = s2 + (gl - 2. * x) * ((pi - 6. * chi) * (gd - 2. * x) + 2. *
1 (2. * pi - 3. * phi - 3. * theta) * (pd + 2. * x))
vol = v1 + v2
surf = s1 + s2
frac = 1. - vol / v0
dsdx = - surf / (x2 - x)
dsdx = coef(kpr) * dsdxsl(kpr) + (1. - coef(kpr)) * dsdx
dsdxsl(kpr) = dsdx
frac = coef(kpr) * fracsl(kpr) + (1. - coef(kpr)) * frac
fracsl(kpr) = frac
surf = coef(kpr) * surfsl(kpr) + (1. - coef(kpr)) * surf

```

```

    surfsl(kpr) = surf
    return
60 vol = 0.
    surf = 0.
    frac = fracsl(kpr) * coef(kpr) + 1. - coef(kpr)
    fracsl(kpr) = frac
    if(frac.gt..9999) frac = 1.
    if(frac.gt..9999) return
    dsdx = 0.
    dsdx = dsdxsl(kpr) * coef(kpr)
    dsdxsl(kpr) = dsdx
    if(abs(dsdx).lt.1.) dsdx = 0.
    surf = surfsl(kpr) * coef(kpr)
    surfsl(kpr) = surf
    return
c
c zero perf calculations start here.
c
70 if(gd - 2. * x.le.0.0) go to 80
    v0 = pi * gd * gd / 4. * gl
    vol = pi * (gd - 2. * x) ** 2 / 4. * (gl - 2. * x)
    frac = 1. - vol / v0
    surf = pi / 2. * (gd - 2. * x) ** 2 + pi * (gd - 2. * x) * (gl -
1 2. * x)
    dsdx = - 2. * pi * (gd + gl - 6. * x)
    return
c
c grain completely burned
c
80 surf = 0.
    frac = 1.0
    vol = 0.
    dsdx = 0.
    nsl = 1
    return
c
c one perf calculation starts here
c
90 if(gd - pd - 4. * x.le.0.0) goto 80
    v0 = pi / 4. * (gd * gd - pd * pd) * gl
    vol = pi / 4. * ((gd - 2. * x) ** 2 - (pd + 2. * x) ** 2) * (gl -
1 2. * x)
    frac = 1. - vol / v0
    surf = pi / 2. * ((gd - 2. * x) ** 2 - (pd + 2. * x) ** 2)
    surf = surf + pi * (gd - 2. * x) * (gl - 2. * x)
    surf = surf + pi * (pd + 2. * x) * (gl - 2. * x)
    dsdx = - 4. * pi * (gd + pd)
    return
c
c below is the calculation for the cylindrical 19 perf grain.
c programmer:fred robbins
c input
c
c p = perf diameter
c d = grain diameter
c gl = grain length
c x = distance burnt
c

```

```

c  output
c
c  vol = the volume of one grain at x.
c  surf = the surface area of one grain at x.
c  frac = the fraction of grain burnt at x.
c
c  w=web
c
110 p = pd
    d = gd
    w = (d - 5. * p) / 6.
    pi = 3.141592654
    sqrt3 = sqrt(3.)
    sqrt5 = sqrt(5.)
    sqrt6 = sqrt(6.)
    sqrt10 = sqrt(10.)
c
c  initial volume and surface area
c
    v0 = pi / 4. * gl * (d * d - 19. * p * p)
    s0 = 2. * v0 / gl + pi * gl * (d + 19. * p)
c
c  x1 = distance to inner sliver burnout
c  x2 = distance to outer sliver burnout
c  dbc = distance between perforation centers
c  assumes burnout does not occur in longitudinal direction
c  w1 = secondary web
c
    dbc = w + p
    w1 = 0.5 * (d - p - 2. * sqrt3 * dbc)
    x1 = dbc / sqrt3 - p / 2.
    x2 = 0.25 * (dbc * (6. - sqrt10) - 2. * p)
    if(x.gt.w / 2.)go to 120
c
c  not slivered yet
c
    vol = pi / 4. * (gl - 2. * x) * ((d - 2. * x) ** 2 - 19. * (p + 2.
1 * x) ** 2)
    surf = 2. * vol / (gl - 2. * x) + pi * (gl - 2. * x) * (d - 2. *
1 x + 19. * (p + 2. * x))
    dsdx = pi * (- 4 * d + 36 * gl - 76 * p - 216 * x)
    frac = 1. - vol / v0
    dsdxsl(kpr) = dsdx
    fracsl(kpr) = frac
    surfsl(kpr) = surf
    return
c
c  v1=total volume of inner sliver, v2=total volume of outer sliver
c  s1=total surface area of inner slivers, s2=total surface area of
c  outer slivers
c
120 v1 = 0.
    v2 = 0.
    s1 = 0.
    s2 = 0.
    delta = 0.
    chi = 0.

```

```

    nsl = 1
    coef(kpr) = 0.
    if(igrad.eq.1.or.igrad.eq.2)go to 140
    if(nslp(kpr).eq.1)goto 130
    tsl(kpr) = yar(3)
    ts(kpr) = w / 2. * ( - 1. + (pbrch / pmean) ** abr(kpr)) /
    1 (bbr(kpr) * (pbase * 1.e - 6) ** abr(kpr))
130 continue
    coef(kpr) = (ts(kpr) + tsl(kpr) - (deltat + yar(3))) / ts(kpr)
    if(coef(kpr).gt.1.)coef(kpr) = 1.
    if(coef(kpr).lt.0.)coef(kpr) = 0.
140 a3 = 0.
    if(x.ge.x2)go to 170
    theta = acos(dbc / (p + 2. * x))
    a1 = theta / 4. * (p + 2. * x) ** 2 - dbc / 4. * sqrt((p + 2. * x)
    1 ** 2 - dbc * dbc)
    if(x.gt.x1)go to 150
    v1 = 3. * (g1 - 2. * x) * (2. * sqrt3 * dbc * dbc - pi * (p + 2.
    1 * x) ** 2 + 24 * a1)
    s1 = 2. * v1 / (g1 - 2. * x) + 12. * (g1 - 2. * x) * (pi - 6. *
    1 theta) * (p + 2. * x)
150 phi = acos((5. * d - 13. * p - 36. * x) / (12. * (p + 2. * x)))
    xi = acos((13. * d - 5. * p - 36. * x) / (12. * (d - 2. * x)))
    if(x.le.w1 / 2.)go to 160
    delta = acos((2. * d - p - 6. * x) / (sqrt3 * (d - 2. * x)))
    chi = acos((d - 2. * p - 6. * x) / (sqrt3 * (p + 2. * x)))
    a3 = .125 * (chi * (p + 2. * x) ** 2 - delta * (d - 2. * x) ** 2
    1 + 2. * sqrt6 * dbc * sqrt(6. * dbc * (p + 2. * x - dbc) - (p + 2.
    1 * x) ** 2))
160 a2 = .125 * (phi * (p + 2. * x) ** 2 - xi * (d - 2. * x) ** 2 + 2.
    1 * sqrt5 * dbc * sqrt((5. * dbc - p - 2. * x) * (5. * dbc - d + 2.
    1 * x)))
    v2 = .25 * (g1 - 2. * x) * (pi * (d - 2. * x) ** 2 - 7. * pi *
    1 (p+2. * x) ** 2 - 24. * sqrt3 * dbc * dbc + 48. * (a1 + a2 + a3))
    s2 = 2. * v2 / (g1 - 2. * x) + (g1 - 2. * x) * ((d - 2. * x) * (pi
    1 - 6. * (xi + delta)) + (p + 2. * x) * (7. * pi - 6. * (2. * theta
    1 + chi + phi)))
    vol = v1 + v2
    surf = s1 + s2
    dsdx = - surf / (x2 - x)
    frac = 1. - vol / v0
    dsdx = coef(kpr) * dsdxsl(kpr) + (1. - coef(kpr)) * dsdx
    dsdxsl(kpr) = dsdx
    frac = coef(kpr) * fracsl(kpr) + (1. - coef(kpr)) * frac
    fracsl(kpr) = frac
    surf = coef(kpr) * surfsl(kpr) + (1. - coef(kpr)) * surf
    surfsl(kpr) = surf
    return
170 vol = 0.
    surf = 0.
    frac = fracsl(kpr) * coef(kpr) + 1. - coef(kpr)
    fracsl(kpr) = frac
    if(frac.gt..9999) frac = 1.
    if(frac.gt..9999)return
    dsdx = 0.

```

```

    dsdx = dsdxsl(kpr) * coef(kpr)
    dsdxsl(kpr) = dsdx
    if(abs(dsdx).lt.1.)dsdx = 0.
    surf = surfsl(kpr) * coef(kpr)
    surfsl(kpr) = surf
    return
c
c below is the calculation for the 19 perf hex grain.
c   programmer:karen a. cieslewicz<std.cont.>
c
c translation of the input values.
c   p= perf diameter
c   d= grain diameter
c   gl= grain length
c   x= distance burnt
c
c translation of the output values.
c   vol= volume of one grain at x.
c   surf= surface area of one grain at x.
c   frac= mass fraction of the grain burnt at x.
c
c assignment statement for pi.
180 pi = 3.141592654
    sqrt3 = sqrt(3.)
    p = pd
    d = gd
c
c   d=6w + 5p is the statement for the grain diameter which will be
c used to calculate the web.
c
c to calculate the web.
    w = (d - 5. * p) / 6.
c
c below is the equation to calculate the distance between the perf cen-
c ters.
    dpc = p + w
c to calculate the grain diameter between the flats.
    f = 2. * (sqrt3 * dpc + p / 2. + w)
c
c to calculate the distance burnt.
    x1 = dpc / sqrt3 - p / 2.
    x2 = 0.125 * (5. * dpc - 4. * p)
c
c to calculate the area.
    a = sqrt3/3. * ((w + p / 2.) ** 2) - pi / 6. * ((w + p / 2.) **2)
c
c to calculate the initial volume of the sharp corner grain.
    vs = gl / 4. * (2. * sqrt3 * f ** 2 - 19. * pi * p ** 2)
c
c to calculate the volume that will be removed from the grain.
    vr = 6. * a * gl
c
c to calculate the initial volume for the 19hex grain with rounded
c corners.
    vo = vs - vr
c
c to calculate the initial surface area of the sharp corner grain.
    ss = 2. * vs / gl + gl * (2. * sqrt3 * f + 19. * pi * p)

```

```

c
c to calculate the surface area that will be removed from the grain.
    sr = 12. * a + gl * (w + p / 2.) * (4. * sqrt3 - 2. * pi)
c
c to calculate the initial surface area for the 19hex grain with rounded
c corners.
    so = ss - sr
c
c to calculate the unknowns of the grain under the condition x.le.5*w.
    if(0.le.x.and.x.le.w / 2.) then
        a = sqrt3 / 3. * (w - 2. * x + (p + 2. * x) / 2.) ** 2 - pi /
1        6. * (w - 2. * x + (p + 2. * x) / 2.) ** 2
c to calculate the volume that will be removed from the sharp corner gra
    vr = 6. * a * (gl - 2. * x)
c to calculate the volume for the sharp corner grain at some distance bu
    vn = .25 * (gl - 2. * x) * (2. * sqrt3 * (f - 2. * x) ** 2. -
1    19. * pi * (p + 2. * x) ** 2.)
c
c to calculate the volume for the 19hex grain with rounded corners.
    v = vn - vr
c
c to calculate the surface area that will be removed from the sharp
c corner grain.
    sr = 12. * a + (gl - 2. * x) * (w - 2 * x + (p + 2. * x) / 2.)
1    * (4. * sqrt3 - 2. * pi)
c to calculate the surface area for the sharp corner grain.
    sn = 2. * v / (gl - 2. * x) + (gl - 2. * x) * (2. * sqrt3 * (f
1    - 2. * x) + 19. * pi * (p + 2. * x))
c
c to calculate the surface area for 19hex grain with rounded corners.
    s = sn - sr
c to calculate the mass fraction.
    frac = 1 - v / vo
    dsdx = - 8. * sqrt3 * (f - 2. * x) - 76. * pi * (p + 2. * x)
1    + (gl - 2. * x) * (- 4. * sqrt3 + 38. * pi) + 16 * sqrt3 *
1    (w + p / 2. - x) - 8. * pi * (w + p / 2. - x) + (gl - 2. * x)
1    * (4. * sqrt3 - 2. * pi)
    surf = s
    vol = v
    dsdxsl(kpr) = dsdx
    fracsl(kpr) = frac
    surfsl(kpr) = surf
    return
endif

c
c due to the cross section at the sliver point x=.5*w there will be 24
c identical inner slivers,12 identical side slivers. after slivering th
c surface area and the volume function become more complex. each type
c sliver will be treated separately and later the volumes will be combin
c to complete the function.
c
c to calculate the 12 identical side slivers for the grain x=.5/w.
    nsl = 1
    coef(kpr) = 0.
    if(igrad.eq.1.or.igrad.eq.2)go to 200
    if(nslp(kpr).eq.1)goto 190

```

```

    tsl(kpr) = yar(3)
    ts(kpr) = w / 2. * ( - 1. + (pbrch / pmean) ** abr(kpr)) /
1 (bbr(kpr) * (pbase * 1.e - 6) ** abr(kpr))
190 continue
    coef(kpr) = (ts(kpr) + tsl(kpr) - (deltat + yar(3))) / ts(kpr)
    if(coef(kpr).gt.1.)coef(kpr) = 1.
    if(coef(kpr).lt.0.)coef(kpr) = 0.
200 if(w / 2.lt.x.and.x.lt.x1.and.x.lt.x2) then
c
c to calculate the areas of the grain.
    a = sqrt3 / 3. * (w - 2. * x + (p + 2. * x) / 2.) ** 2 - pi /
1 6. * (w - 2. * x + (p + 2. * x) / 2.) ** 2
    theta = acos(dpc / (p + 2. * x))
    a1 = theta / 4. * (p + 2. * x) ** 2 - dpc / 4. * sqrt((p + 2.
1 * x) ** 2 - dpc ** 2)
    omega = acos(2. * dpc / (p + 2. * x) - 1.)
    a2 = 0.125 * (p + 2. * x) * ((p + 2. * x) * (omega + sin(omega
1 )) - 2. * dpc * sin(omega))
c to calculate the volumes of the grain.
    v1 = 3. * (g1 - 2. * x) * (2. * sqrt3 * dpc ** 2 - pi * (p +
1 2. * x) ** 2 + 24. * a1)
    v2 = 6. * (g1 - 2. * x) * (2. * dpc ** 2 - dpc * (p + 2. * x)
1 - pi / 4. * (p + 2. * x) ** 2 + 2. * a1 + 4. * a2)
c to calculate the surface areas of the grain.
    s1 = 2. * v1 / (g1 - 2. * x) + 12. * (g1 - 2. * x) * (pi - 6.
1 * theta) * (p + 2. * x)
    s2 = 2. * v2 / (g1 - 2. * x) + 12. * (g1 - 2. * x) * (dpc + (p
1 + 2. * x) * (pi / 2. - omega - theta - sin(omega)))
c to calculate the total volume and total surface area.
    vf = v1 + v2
    sf = s1 + s2
c to calculate the mass fraction.
    frac = 1. - vf / vo
    surf = sf
    dsdx = - surf / (x2 - x)
    vol = vf
    dsdx = coef(kpr) * dsdxsl(kpr) + (1. - coef(kpr)) * dsdx
    dsdxsl(kpr) = dsdx
    frac = coef(kpr) * fracsl(kpr) + (1. - coef(kpr)) * frac
    fracsl(kpr) = frac
    surf = coef(kpr) * surfsl(kpr) + (1. - coef(kpr)) * surf
    surfsl(kpr) = surf
    return
endif
    if(x.gt.x1.and.x.lt.x2)then
c to calculate the area of the grain.
    a = sqrt3 / 3. * (w - 2. * x + (p + 2. * x) / 2.) ** 2 - pi /
1 6. * (w - 2. * x + (p + 2. * x) / 2.) ** 2
    theta = acos(dpc / (p + 2. * x))
    a1 = theta / 4. * (p + 2. * x) ** 2 - dpc / 4. * sqrt((p + 2.
1 * x) ** 2 - dpc ** 2)
    omega = acos(2. * dpc / (p + 2. * x) - 1.)
    a2 = 0.125 * (p + 2. * x) * ((p + 2. * x) * (omega +
1 sin(omega)) - 2. * dpc * sin(omega))
c to calculate the volume of the grain.
    v2 = 6. * (g1 - 2. * x) * (2. * dpc ** 2 - dpc * (p + 2. * x)
1 - pi / 4. * (p + 2. * x) ** 2 + 2. * a1 + 4. * a2)
c to calculate the surface area of the grain.

```

```

        s2 = 2. * v2 / (g1 - 2. * x) + 12. * (g1 - 2. * x) * (dpc + (p
1      + 2. * x) * (pi / 2. - omega - theta - sin(omega)))
c to calculate the volume and the surface area.
        vf = v2
        sf = s2
c to calculate the the mass fraction.
        frac = 1 - vf / vo
        surf = sf
        dsdx = - surf / (x2 - x)
        vol = vf
        dsdx = coef(kpr) * dsdxsl(kpr) + (1. - coef(kpr)) * dsdx
        dsdxsl(kpr) = dsdx
        frac = coef(kpr) * fracsl(kpr) + (1. - coef(kpr)) * frac
        fracsl(kpr) = frac
        surf = coef(kpr) * surfsl(kpr) + (1. - coef(kpr)) * surf
        surfsl(kpr) = surf
        return
    endif
    if(x.gt.x2)then
        dsdx = 0.
        surf = 0.
        vol = 0.
        frac = fracsl(kpr) * coef(kpr) + 1. - coef(kpr)
        fracsl(kpr) = frac
        if(frac.gt..9999) frac = 1.
        if(frac.gt..9999)return
        dsdx = 0.
        dsdx = dsdxsl(kpr) * coef(kpr)
        dsdxsl(kpr) = dsdx
        if(abs(dsdx).lt.1.)dsdx = 0.
        surf = surfsl(kpr) * coef(kpr)
        surfsl(kpr) = surf
        return
    endif
    stop
c
c    spherical grain calculations start here
c
210 if (gd .le. 2.*x) goto 80
    vol = pi / 6. * (gd - 2. * x) ** 3
    surf = pi * (gd - 2. * x) ** 2
    frac = ((gd - 2. * x) / gd) ** 3
    dsdx = pi * 4. * (2. * x - gd)
    return
c
220 format (1x,'unacceptable granulation')
    end
    subroutine jint(btdia,btlen,x,y,nchpts,chdist,chkdiam,bint,bvol)
    dimension bint(10),chdist(6),chkdiam(6)
    pi=3.141593
    areaa=pi*btdia*btdia/4.
    distbp=x-btlen
    points=100.
    step=y/points
    zz=0.
    bvoll=0.
    ichg=0
    do 1 j=1,10

```

```

    bint(j)=0.
1    continue
    if(step.lt.1.e-10)then
    bint(7)=1./(pi*(chdiam(1)**2/4.))**2
    return
    endif
    intsw=0
    do 2 j=2,nchpts
    if(y.gt.chdist(j)) go to 2
    nchp=j
    ichg=1
    holddt=chdist(j)
    holddm=chdiam(j)
    diam=(y-chdist(j-1))/(chdist(j)-chdist(j-1))
    chdiam(j)=chdiam(j-1)+diam*(chdiam(j)-chdiam(j-1))
    chdist(j)=y
    go to 3
2    continue
    nchp=nchpts+1
    chdist(nchp)=y
    chdiam(nchp)=chdiam(nchpts)
c    write(6,*)chdist(nchp-1),chdist(nchp)
c 910    format(1x,2e11.4)
3    continue
    areal=chdiam(1)**2 * pi / 4.
    bint5o=0.0
    do 58 il=1,nchp-1
    npt=int((chdist(il+1)-chdist(il))/step)
    if(npt.le.0)npt=1
c    write(6,*)npt,chdist(il),chdist(il+1),step
    step1=(chdist(il+1)-chdist(il))/npt
c    write(6,912)chdiam(nchp-1),chdiam(nchp)
c 912    format(1x,2e11.4)
    r1=chdiam(il)*.5
c    areal=pi*r1*r1
    do 57 i=1,npt
    zz=zz+step1
    if(zz.gt.chdist(nchp))zz=chdist(nchp)
    diam=(zz-chdist(il))/(chdist(il+1)-chdist(il))
    diam=chdiam(il)+diam*(chdiam(il+1)-chdiam(il))
    r2=0.5*diam
    area2=pi*r2*r2
    bvol2=bvol1+step1*pi/3.*(r1*r1+r1*r2+r2*r2)
    if(zz.gt.distbp)then
    if(intsw.eq.0)then
    diam=(distbp-chdist(il))/(chdist(il+1)-chdist(il))
    diam=chdiam(il)+diam*(chdiam(il+1)-chdiam(il))
    r2a=0.5*diam
    area2=pi*r2a*r2a
    step1a=step1-(zz-distbp)
    bvol2=bvol1+step1a*pi/3.*(r1*r1+r1*r2a+r2a*r2a)
    bint1=bint(1)+0.5*step1a*(bvol1/areal+bvol2/area2)
    bint(2)=bvol2*bvol2/area2/area2
    bint(3)=bint(3)+0.5*step1a*(bint(1)*areal+bint1*area2)
    bint(4)=bint(4)+.5*step1a*(bvol1*bvol1/areal+bvol2*bvol2/area2)
c    bint(5)=bint(5)+.5*step1a*(1./areal+1./area2)
    bint(6)=bvol2/area2/area2
    bint(7)=1./area2/area2

```

```

bint(1)=bint1
bvol1=bvol2
areal=area2-areaa
area2=pi*r2*r2
area2=area2-areaa
stepla=zz-distbp
r1=r2a
bvol2=bvol1+stepla*pi/3.*(r1*r1+r1*r2+r2*r2)
bvol2=bvol2-stepla*areaa
bint10=bint(10)+0.5*stepla*(bvol1/areal+bvol2/area2)
bint5=bint(5)+0.5*stepla*(1./areal+1./area2)
bint(2)=bvol2*bvol2/area2/area2
c   bint(3)=bint(3)+0.5*stepla*(bint(1)*areal+bint1*area2)
bint(4)=bint(4)+.5*stepla*(bvol1*bvol1/areal+bvol2*bvol2/area2)
c   bint(5)=bint(5)+.5*stepla*(1./areal+1./area2)
bint(6)=bvol2/area2/area2
bint(7)=1./area2/area2
c   bint5a=bint5+.5*stepla*(1./areal+1./area2)
bint(8)=bint(8)+.5*stepla*(areal*bint(5)+bint5*area2)
bint(9)=bint(9)+.5*stepla*(areal*bint(10)+area2*bint10)
bint(10)=bint10
bint(1)=bint1
bint(5)=bint5
c   bint5o=bint5a
areal=area2
bvol1=bvol2
r1=r2
intsw=1
go to 57
else
area2=area2-areaa
bvol2=bvol2-step1*areaa
bint10=bint(10)+0.5*step1*(bvol1/areal+bvol2/area2)
bint(2)=bvol2*bvol2/area2/area2
c   bint(3)=bint(3)+0.5*step1*(bint(1)*areal+bint1*area2)
bint(4)=bint(4)+0.5*step1*(bvol1*bvol1/areal+bvol2*bvol2/area2)
c   bint(5)=bint(5)+.5*step1*(1./areal+1./area2)
bint(6)=bvol2/area2/area2
bint(7)=1./area2/area2
bint5=bint(5)+.5*step1*(1./areal+1./area2)
bint(8)=bint(8)+.5*step1*(areal*bint(5)+bint5*area2)
bint(9)=bint(9)+.5*step1*(areal*bint(10)+area2*bint10)
bint(10)=bint10
c   bint(1)=bint1
bint(5)=bint5
c   bint5o=bint5a
areal=area2
r1=r2
bvol1=bvol2
go to 57
endif
endif
bint1=bint(1)+0.5*step1*(bvol1/areal+bvol2/area2)
bint(2)=bvol2*bvol2/area2/area2
bint(3)=bint(3)+0.5*step1*(bint(1)*areal+bint1*area2)
bint(4)=bint(4)+0.5*step1*(bvol1*bvol1/areal+bvol2*bvol2/area2)
c   bint(5)=bint(5)+.5*step1*(1./areal+1./area2)
bint(6)=bvol2/area2/area2

```

```

        bint(7)=1./area2/area2
        bint(1)=bint1
        areal=area2
        r1=r2
        bvol1=bvol2
57      continue
58      continue
        bvol=bvol2
        if(ichg.eq.1)then
            chdiam(nchp)=holddm
            chdist(nchp)=holddt
        endif
c        write(6,915)
c 915    format('1',1x,'Leaving Jint')
        return
    end

```

INTENTIONALLY LEFT BLANK.

LIST OF SYMBOLS

ρ	density
$A(z)$	area at position z
$A(z_a)$	area at position z_a
$A(z_a^-)$	area just before the aft end of the projectile
$A(z_a^+)$	area just after the aft end of the projectile
A_B	cross-sectional area of the bore
A_A	cross-sectional area of the boattail
A_{BA}	area at the base of the projectile
C	total charge mass
g_0	a constant to reconcile dimensions
m	mass flux
M_p	mass of projectile
P	pressure
P_B	projectile base pressure
P_{BR}	breech pressure
P_m	mean pressure
P_{res}	resistive pressure
$P(z_a)$	pressure on the aft end of the projectile
t	time
u	velocity
$V(x,t)$	volume as a function of distance and time
$V(z_a,t)$	volume, at z_a , as a function of time
$V(z_p,t)$	volume, at z_p , as a function of time
V_p	projectile velocity
$\cdot$	derivative with respect to time

INTENTIONALLY LEFT BLANK.

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145	1	Commander U.S. Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander U.S. Army Materiel Command ATTN: AMCAM 5001 Eisenhower Ave. Alexandria, VA 22333-0001	1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-JSK (Armor Eng. Br.) Warren, MI 48397-5000
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Tech Publishing 2800 Powder Mill Rd. Adelphi, MD 20783-1145	1	Director U.S. Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Records Management 2800 Powder Mill Rd. Adelphi, MD 20783-1145	(Class. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-CD (Security Mgr.) Fort Benning, GA 31905-5660
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-IMI-I Picatinny Arsenal, NJ 07806-5000	(Unclass. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-WCB-O Fort Benning, GA 31905-5000
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000	1	WL/MNOI Eglin AFB, FL 32542-5000 <u>Aberdeen Proving Ground</u>
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050	2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Director U.S. Army Advanced Systems Research and Analysis Office (ATCOM) ATTN: AMSAT-R-NR, M/S 219-1 Ames Research Center Moffett Field, CA 94035-1000	1	Cdr, USATECOM ATTN: AMSTE-TC
		1	Dir, ERDEC ATTN: SCBRD-RT
		1	Cdr, CBDA ATTN: AMSCB-CII
		1	Dir, USARL ATTN: AMSRL-SL-I
		10	Dir, USARL ATTN: AMSRL-OP-CI-B (Tech Lib)

<u>No. of Copies</u>	<u>Organization</u>
1	Chairman DOD Explosives Safety Board Room 856-C Hoffman Bldg. 1 2461 Eisenhower Avenue Alexandria, VA 22331-0600
1	Headquarters U.S. Army Materiel Command ATTN: AMCICP-AD, M. Fisette 5001 Eisenhower Ave. Alexandria, VA 22333-0001
1	U.S. Army Ballistic Missile Defense Systems Command Advanced Technology Center P.O. Box 1500 Huntsville, AL 35807-3801
1	Department of the Army Office of the Product Manager 155mm Howitzer, M109A6, Paladin ATTN: SFAE-AR-HIP-IP, Mr. R. De Kleine Picatinny Arsenal, NJ 07806-5000
3	Project Manager Advanced Field Artillery System ATTN: SFAE-ASM-AF-E, LTC A. Ellis T. Kuriata J. Shields Picatinny Arsenal, NJ 07801-5000
1	Project Manager Advanced Field Artillery System ATTN: SFAE-ASM-AF-Q-W, Warren Picatinny Arsenal, NJ 07801-5000
2	Commander Production Base Modernization Agency U.S. Army Armament Research, Development, and Engineering Center ATTN: AMSMC-PBM, A. Siklosi AMSMC-PBM-E, L. Laibson Picatinny Arsenal, NJ 07806-5000

<u>No. of Copies</u>	<u>Organization</u>
4	PEO-Armaments Project Manager Tank Main Armament System ATTN: AMCPM-TMA AMCPM-TMA-105 AMCPM-TMA-120 AMCPM-TMA-AS, H. Yuen Picatinny Arsenal, NJ 07806-5000
4	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCH-V, C. Mandala E. Fennell SMCAR-CCH-T, L. Rosendorf SMCAR-CCS Picatinny Arsenal, NJ 07806-5000
19	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-AEE, J. Lannon SMCAR-AEE-B, A. Beardell D. Downs S. Einstein S. Westley S. Bernstein J. Rutkowski B. Brodman P. O'Reilly R. Cirincione A. Grabowsky P. Hui J. O'Reilly SMCAR-AEE-WW, M. Mezger J. Pinto D. Wiegand P. Lu C. Hu SMCAR-AES, S. Kaplowitz Picatinny Arsenal, NJ 07806-5000
1	Commander U.S. Army Armament Research, Development and Engineering Center ATTN: SMCAR-HFM, E. Barrieres Picatinny Arsenal, NJ 07806-5000

<u>No. of Copies</u>	<u>Organization</u>
9	Commander U.S. Army Armament Research, Development and Engineering Center ATTN: SMCAR-FSA-T, M. Salsbury SMCAR-FSA-F, LTC R. Riddle SMCAR-FSC, G. Ferdinand SMCAR-FS, T. Gora SMCAR-FS-DH, J. Feneck SMCAR-FSS-A, R. Kopmann B. Machek L. Pinder SMCAR-FSN-N, K. Chung Picatinny Arsenal, NJ 07806-5000
3	Director Benet Weapons Laboratories ATTN: SMCAR-CCB-RA, G.P. O'Hara G.A. Pilegl SMCAR-CCB-S, F. Heiser Watervliet, NY 12189-4050
2	Commander U.S. Army Research Office ATTN: Technical Library D. Mann P.O. Box 12211 Research Triangle Park, NC 27709-2211
1	Commander, USACECOM R&D Technical Library ATTN: ASQNC-ELC-IS-L-R, Myer Center Fort Monmouth, NJ 07703-5301
1	Commandant U.S. Army Aviation School ATTN: Aviation Agency Fort Rucker, AL 36360
1	Program Manager U.S. Tank-Automotive Command ATTN: AMCPM-ABMS, T. Dean Warren, MI 48092-2498

<u>No. of Copies</u>	<u>Organization</u>
1	Project Manager U.S. Tank-Automotive Command Fighting Vehicle Systems ATTN: SFAE-ASM-BV Warren, MI 48397-5000
1	Project Manager, Abrams Tank System ATTN: SFAE-ASM-AB Warren, MI 48397-5000
1	Director HQ, TRAC RPD ATTN: ATCD-MA Fort Monroe, VA 23651-5143
1	Commander U.S. Army Belvoir Research and Development Center ATTN: STRBE-WC Fort Belvoir, VA 22060-5006
1	Director U.S. Army TRAC-Ft. Lee ATTN: ATRC-L, Mr. Cameron Fort Lee, VA 23801-6140
1	Commandant U.S. Army Command and General Staff College Fort Leavenworth, KS 66027
1	Commandant U.S. Army Special Warfare School ATTN: Rev and Trng Lit Div Fort Bragg, NC 28307
1	Commander Radford Army Ammunition Plant ATTN: SMCAR-QA/HI LIB Radford, VA 24141-0298
1	Commander U.S. Army Foreign Science and Technology Center ATTN: AMXST-MC-3 220 Seventh Street, NE Charlottesville, VA 22901-5396

No. of Copies	Organization
2	Commandant U.S. Army Field Artillery Center and School ATTN: ATSF-CO-MW, E. Dublisky ATSF-CN, P. Gross Ft. Sill, OK 73503-5600
1	Commandant U.S. Army Armor School ATTN: ATZK-CD-MS, M. Falkovitch Armor Agency Fort Knox, KY 40121-5215
2	Commander Naval Sea Systems Command ATTN: SEA 62R SEA 64 Washington, DC 20362-5101
1	Commander Naval Air Systems Command ATTN: AIR-954-Tech Library Washington, DC 20360
4	Commander Naval Research Laboratory ATTN: Technical Library Cale 4410, K. Kailasanate J. Boris E. Oran Washington, DC 20375-5000
1	Office of Naval Research ATTN: Code 473, R.S. Miller 800 N. Quincy Street Arlington, VA 22217-9999
1	Office of Naval Technology ATTN: ONT-213, D. Siegel 800 N. Quincy St. Arlington, VA 22217-5000

No. of Copies	Organization
3	Commander Naval Surface Warfare Center ATTN: Code 730 Code R-13, R. Bernecker H. Sandusky Silver Spring, MD 20903-5000
7	Commander Naval Surface Warfare Center ATTN: T.C. Smith K. Rice S. Mitchell S. Peters J. Consaga C. Gotzmer Technical Library Indian Head, MD 20640-5000
4	Commander Naval Surface Warfare Center ATTN: Code G30, Guns & Munitions Div Code G32, Guns Systems Div Code G33, T. Doran Code E23 Technical Library Dahlgren, VA 22448-5000
5	Commander Naval Air Warfare Center ATTN: Code 388, C.F. Price T. Boggs Code 3895, T. Parr R. Derr Information Science Division China Lake, CA 93555-6001
1	Commanding Officer Naval Underwater Systems Center ATTN: Code 5B331, Technical Library Newport, RI 02840
1	AFOSR/NA ATTN: J. Tishkoff Bolling AFB, D.C. 20332-6448
1	OLAC PL/TSTL ATTN: D. Shiplett Edwards AFB, CA 93523-5000

<u>No. of</u> <u>Copies</u>	<u>Organization</u>	<u>No. of</u> <u>Copies</u>	<u>Organization</u>
3	AL/LSCF ATTN: J. Levine L. Quinn T. Edwards Edwards AFB, CA 93523-5000	1	Director Sandia National Laboratories Energetic Materials & Fluid Mechanics Department, 1512 ATTN: M. Baer P.O. Box 5800 Albuquerque, NM 87185
1	WL/MNAA ATTN: B. Simpson Eglin AFB, FL 32542-5434	1	Director Sandia National Laboratories Combustion Research Facility ATTN: R. Carling Livermore, CA 94551-0469
1	WL/MNME Energetic Materials Branch 2306 Perimeter Rd. STE 9 Eglin AFB, FL 32542-5910	1	Director Sandia National Laboratories ATTN: 8741, G. A. Beneditti P.O. Box 969 Livermore, CA 94551-0969
1	WL/MNSH ATTN: R. Drabczuk Eglin AFB, FL 32542-5434	2	Director Lawrence Livermore National Laboratory ATTN: L-355, A. Buckingham M. Finger P.O. Box 808 Livermore, CA 94550-0622
2	NASA Langley Research Center ATTN: M.S. 408, W. Scallion D. Witcofski Hampton, VA 23605	2	Director Los Alamos Scientific Lab ATTN: T3/D. Butler M. Division/B. Craig P.O. Box 1663 Los Alamos, NM 87544
1	Central Intelligence Agency Office of the Central References Dissemination Branch Room GE-47, HQS Washington, DC 20502	2	Battelle ATTN: TACTEC Library, J.N. Huggins V. Levin 505 King Avenue Columbus, OH 43201-2693
1	Central Intelligence Agency ATTN: J. Backofen NHB, Room 5N01 Washington, DC 20505	1	Battelle PNL ATTN: Mr. Mark Garnich P.O. Box 999 Richland, WA 99352
1	SDIO/TNI ATTN: L.H. Caveny Pentagon Washington, DC 20301-7100	1	Institute of Gas Technology ATTN: D. Gidaspow 3424 S. State Street Chicago, IL 60616-3896
1	SDIO/DA ATTN: E. Gerry Pentagon Washington, DC 21301-7100		
2	HQ DNA ATTN: D. Lewis A. Fahey 6801 Telegraph Rd. Alexandria, VA 22310-3398		

<u>No. of Copies</u>	<u>Organization</u>
1	Institute for Advanced Technology ATTN: T.M. Kiehne The University of Texas at Austin 4030-2 W. Braker Lane Austin, TX 78759-5329
2	CPIA - JHU ATTN: H. J. Hoffman T. Christian 10630 Little Patuxent Parkway Suite 202 Columbia, MD 21044-3200
1	Brigham Young University Department of Chemical Engineering ATTN: M. Beckstead Provo, UT 84601
1	Jet Propulsion Laboratory California Institute of Technology ATTN: L.D. Strand, MS 125/224 4800 Oak Grove Drive Pasadena, CA 91109
1	California Institute of Technology 204 Karman Lab Main Stop 301-46 ATTN: F.E.C. Culick 1201 E. California Street Pasadena, CA 91109
3	Georgia Institute of Technology School of Aerospace Engineering ATTN: B.T. Zim E. Price W.C. Strahle Atlanta, GA 30332
1	Massachusetts Institute of Technology Department of Mechanical Engineering ATTN: T. Toong 77 Massachusetts Avenue Cambridge, MA 02139-4307
2	University of Illinois Department of Mechanical/Industry Engineering ATTN: H. Krier R. Beddini 144 MEB: 1206 N. Green St. Urbana, IL 61801-2978

<u>No. of Copies</u>	<u>Organization</u>
1	University of Maryland ATTN: Dr. J.D. Anderson College Park, MD 20740
1	University of Massachusetts Department of Mechanical Engineering ATTN: K. Jakus Amherst, MA 01002-0014
1	University of Minnesota Department of Mechanical Engineering ATTN: E. Fletcher Minneapolis, MN 55414-3368
3	Pennsylvania State University Department of Mechanical Engineering ATTN: V. Yang K. Kuo C. Merkle University Park, PA 16802-7501
1	Rensselaer Polytechnic Institute Department of Mathematics Troy, NY 12181
1	Stevens Institute of Technology Davidson Laboratory ATTN: R. McAlevy III Castle Point Station Hoboken, NJ 07030-5907
1	Rutgers University Department of Mechanical and Aerospace Engineering ATTN: S. Temkin University Heights Campus New Brunswick, NJ 08903
1	University of Southern California Mechanical Engineering Department ATTN: OHE200, M. Gerstein Los Angeles, CA 90089-5199
1	University of Utah Department of Chemical Engineering ATTN: A. Baer Salt Lake City, UT 84112-1194
1	Washington State University Department of Mechanical Engineering ATTN: C.T. Crowe Pullman, WA 99163-5201

<u>No. of Copies</u>	<u>Organization</u>
1	AFELM, The Rand Corporation ATTN: Library D 1700 Main Street Santa Monica, CA 90401-3297
1	Arrow Technology Associates, Inc. ATTN: W. Hathaway P.O. Box 4218 South Burlington, VT 05401-0042
3	AAI Corporation ATTN: J. Hebert J. Frankle D. Cleveland P.O. Box 126 Hunt Valley, MD 21030-0126
2	Alliant Techsystems, Inc. ATTN: R.E. Tompkins J. Kennedy 7225 Northland Dr. Brooklyn Park, MN 55428
1	AVCO Everett Research Laboratory ATTN: D. Stickler 2385 Revere Beach Parkway Everett, MA 02149-5936
1	General Applied Sciences Lab ATTN: J. Erdos 77 Raynor Ave. Ronkonkoma, NY 11779-6649
1	General Electric Company Tactical System Department ATTN: J. Mandzy 100 Plastics Ave. Pittsfield, MA 01201-3698
1	ITRI ATTN: M.J. Klein 10 W. 35th Street Chicago, IL 60616-3799
4	Hercules, Inc. Radford Army Ammunition Plant ATTN: L. Gizzi D.A. Worrell W.J. Worrell C. Chandler Radford, VA 24141-0299

<u>No. of Copies</u>	<u>Organization</u>
2	Hercules, Inc. Allegheny Ballistics Laboratory ATTN: William B. Walkup Thomas F. Farabaugh P.O. Box 210 Rocket Center, WV 26726
1	Hercules, Inc. Aerospace ATTN: R. Cartwright 100 Howard Blvd. Kenville, NJ 07847
1	Hercules, Inc. Hercules Plaza ATTN: B.M. Riggleman Wilmington, DE 19894
1	MBR Research Inc. ATTN: Dr. Moshe Ben-Reuven 601 Ewing St., Suite C-22 Princeton, NJ 08540
1	Olin Corporation Badger Army Ammunition Plant ATTN: F.E. Wolf Baraboo, WI 53913
3	Olin Ordnance ATTN: E.J. Kirschke A.F. Gonzalez D.W. Worthington P.O. Box 222 St. Marks, FL 32355-0222
1	Olin Ordnance ATTN: H.A. McElroy 10101 9th Street, North St. Petersburg, FL 33716
1	Paul Gough Associates, Inc. ATTN: P.S. Gough 1048 South St. Portsmouth, NH 03801-5423
1	Physics International Library ATTN: H. Wayne Wampler P.O. Box 5010 San Leandro, CA 94577-0599

No. of
Copies Organization

- 2 Princeton Combustion Research
 Laboratories, Inc.
 ATTN: N. Mer
 N.A. Messina
 Princeton Corporate Plaza
 11 Deerpark Dr., Bldg IV, Suite 119
 Monmouth Junction, NJ 08852
- 3 Rockwell International
 Rocketdyne Division
 ATTN: BA08,
 J. Flanagan
 J. Gray
 R.B. Edelman
 6633 Canoga Avenue
 Canoga Park, CA 91303-2703
- 2 Rockwell International Science Center
 ATTN: Dr. S. Chakravarthy
 Dr. S. Palaniswamy
 1049 Camino Dos Rios
 P.O. Box 1085
 Thousand Oaks, CA 91360
- 1 Science Applications International Corp.
 ATTN: M. Palmer
 2109 Air Park Rd.
 Albuquerque, NM 87106
- 1 Southwest Research Institute
 ATTN: J.P. Riegel
 6220 Culebra Road
 P.O. Drawer 28510
 San Antonio, TX 78228-0510
- 1 Sverdrup Technology, Inc.
 ATTN: Dr. John Deur
 2001 Aerospace Parkway
 Brook Park, OH 44142
- 3 Thiokol Corporation
 Elkton Division
 ATTN: R. Willer
 R. Biddle
 Tech Library
 P.O. Box 241
 Elkton, MD 21921-0241
- 1 Veritay Technology, Inc.
 ATTN: E. Fisher
 4845 Millersport Hwy.
 East Amherst, NY 14501-0305

No. of
Copies Organization

- 1 Universal Propulsion Company
 ATTN: H.J. McSpadden
 25401 North Central Ave.
 Phoenix, AZ 85027-7837
- 1 SRI International
 Propulsion Sciences Division
 ATTN: Tech Library
 333 Ravenwood Avenue
 Menlo Park, CA 94025-3493
- Aberdeen Proving Ground
- 1 Cdr. USACSTA
 ATTN: STECS-PO/R. Hendricksen

<u>No. of Copies</u>	<u>Organization</u>
1	Ernst-Mach-Institut ATTN: Dr. R. Heiser Hauptstrasse 18 Weil am Rhein Germany
1	Defence Research Agency, Military Division ATTN: C. Woodley RARDE Fort Halstead Sevenoaks, Kent, TN14 7BP England
1	School of Mechanical, Materials, and Civil Engineering ATTN: Dr. Bryan Lawton Royal Military College of Science Shrivenham, Swindon, Wiltshire, SN6 8LA England

<u>No. of Copies</u>	<u>Organization</u>
2	Institut Saint Louis ATTN: Dr. Marc Giraud Dr. Gunther Sheets Postfach 1260 7858 Weil am Rhein 1 Germany
1	Explosive Ordnance Division ATTN: A. Wildegger-Gaissmaier Defence Science and Technology Organisation P.O. Box 1750 Salisbury, South Australia 5108
1	Armaments Division ATTN: Dr. J. Lavigne Defence Research Establishment Valcartier 2459, Pie XI Blvd., North P.O. Box 8800 Courcellette, Quebec G0A 1R0 Canada

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

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

1. ARL Report Number ARL-TR-181 Date of Report August 1993
2. Date Report Received _____
3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

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

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

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

CURRENT
ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

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

OLD
ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL

FIRST CLASS PERMIT No 0001, APG, MD

Postage will be paid by addressee

Director
U.S. Army Research Laboratory
ATTN: AMSRL-OP-CI-B (Tech Lib)
Aberdeen Proving Ground, MD 21005-5066



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

