

AD-A260 481



ARMY RESEARCH LABORATORY



BRLCB:
A Closed-Chamber Data
Analysis Program
Part II—Theory and User's Manual
(Appendices D–M)

William F. Oberle
Douglas E. Kooker

ARL-TR-36
(Part II)

January 1993

DTIC
ELECTE
FEB 09 1993
S E D

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

93-02362



25728

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 January 1993	3. REPORT TYPE AND DATES COVERED Final, October 1990–October 1992		
4. TITLE AND SUBTITLE BRLCB: A Closed-Chamber Data Analysis Program Part II—Theory and User's Manual (Appendices D–M)		5. FUNDING NUMBERS DA311880 1FZZ9W 9XDGS3		
6. AUTHOR(S) William Oberle and Douglas Kooker				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		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-36 (Part II)		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) BRLCB is a PC-based analysis program designed to perform all data analysis associated with closed chamber experiments. Included in the program are provisions for deterred and layered propellants and electrothermal-chemical augmented firings. The basic features of the program and user's guide are presented and validated, and future plans and additions to the program are outlined.				
14. SUBJECT TERMS closed bomb; closed chamber; deterred propellant; burning rate; propellants		15. NUMBER OF PAGES 248		
		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 SAR	

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

	<u>Page</u>
APPENDIX D: LISTING - BATCH FILE BRLCB.BAT	137
APPENDIX E: LISTING - PROGRAM MKCHCE.FOR	141
APPENDIX F: LISTING - PROGRAM MKMASTER.FOR	145
APPENDIX G: LISTING - PROGRAM MKGAGE.FOR	191
APPENDIX H: LISTING - PROGRAM MKPTDATA.FOR	201
APPENDIX I: LISTING - PROGRAM MKINF.FOR	231
APPENDIX J: LISTING - PROGRAM MKSMOOTH.FOR	243
APPENDIX K: LISTING - PROGRAM MKCAL.FOR	269
APPENDIX L: LISTING - PROGRAM MKOUT.FOR	335
APPENDIX M: SMOOTHING AND DIFFERENTIATION COEFFICIENTS	377
DISTRIBUTION LIST	395

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

U.S. GOVERNMENT PRINTING OFFICE: 1970-3

INTENTIONALLY LEFT BLANK.

PREFACE

The Appendices (D-M) contained in this volume contain the entire FORTRAN 77 listing for the program BRLCB. Details concerning the program itself are provided in the first half of the report "BRLCB: A Closed-Chamber Data Analysis Program, Part I—Theory and User's Manual," which is published as a separate volume.

INTENTIONALLY LEFT BLANK.

APPENDIX D:
LISTING - BATCH FILE BRLCB.BAT

INTENTIONALLY LEFT BLANK.

REM BATCH FILE BRLCB.BAT
ECHO OFF
graphics
mkchce
opt

INTENTIONALLY LEFT BLANK.

APPENDIX E:
LISTING - PROGRAM MKCHCE.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKCHCE

```
C*****
C VERSION: 3.0; January 1992
C     Final Cleanup, 4 April 1992
C
C Written By: William Oberle; U.S. Army Research Laboratory
C
C Purpose: The purpose of this program is to provide
C     initial choices for BRLCB and create a
C     IBM-DOS batch file named OPT.BAT to the
C     drive the program.
C*****
C     OPEN (UNIT = 8, FILE = 'OPT.BAT')
C     REWIND (8)
C     WRITE (8, *) 'ECHO OFF'
C     CALL CLEAR
C The main menu of selections is written.
C     WRITE (*, 6000)
C 6000 FORMAT(20X,'BRLCB ** Version 3.0 ** JANUARY 1992')
C     WRITE (*, *)
C     WRITE (*, *)
C     WRITE (*, 6010)
C 6010 FORMAT(35X,'Main Menu')
C     WRITE (*, *)
C     WRITE (*, 6020)
C 6020 FORMAT(20x,'1. Create Master Information File')
C     WRITE (*, 6030)
C 6030 FORMAT(20x,'2. Update gage information')
C     WRITE (*, 6040)
C 6040 FORMAT(20x,'3. Prepare pressure/time data')
C     WRITE (*, 6050)
C 6050 FORMAT(20x,'4. Prepare firing information file')
C     WRITE (*, 6060)
C 6060 FORMAT(20x,'5. Smooth pressure/time data')
C     WRITE (*, 6070)
C 6070 FORMAT(20x,'6. Perform data analysis')
C     WRITE (*, 6080)
C 6080 FORMAT(20x,'7. Prepare output')
C     WRITE (*, 6090)
C 6090 FORMAT(20x,'8. Exit program')
C     WRITE (*, *)
C 1000 CONTINUE
C     WRITE (*, *) 'Please Enter Your Choice (1 - 8): '
C     READ (*, *) ICHOICE
```

```

IF (ICHOICE .GT. 8) THEN
  WRITE (*, *)
  WRITE (*, *) 'The choice you have made is not between'
  WRITE (*, *) '1 and 8. Please select again.'
  GO TO 1000
END IF
C The batch file OPT.BAT is created for the proper choice.
IF (ICHOICE .EQ. 1) WRITE (8, *) 'MKMASTER'
IF (ICHOICE .EQ. 2) WRITE (8, *) 'MKGAGE'
IF (ICHOICE .EQ. 3) WRITE (8, *) 'MKPTDATA'
IF (ICHOICE .EQ. 4) WRITE (8, *) 'MKINF'
IF (ICHOICE .EQ. 5) WRITE (8, *) 'MKSMOOTH'
IF (ICHOICE .EQ. 6) WRITE (8, *) 'MKCAL'
IF (ICHOICE .EQ. 7) WRITE (8, *) 'MKOUT'
IF (ICHOICE .EQ. 8) GO TO 1010
WRITE (8, *) 'BRL'
1010 CONTINUE
CLOSE (8)
END
C*****
C***** SUBROUTINE CLEAR*****
C*****
SUBROUTINE CLEAR
CHARACTER ST*4
DATA ST/' [2J'/
WRITE (*, 6000) ST
6000 FORMAT (1X,A4)
RETURN
END

```

APPENDIX F:
LISTING - PROGRAM MKMASTER.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKMASTER

```

C*****
C   Version: 3.0, January 1992
C   Written by: William Oberle, U.S. Army Research Laboratory
C               Last Modified: 12/30/91 ** Depth of Each layer added
C                       to P(1,J,2)
C               1/19/92 ** Modified for homogeneous grain
C               2/8/92: Cleanup, hex grain fixed
C
C   Purpose : This program is the calling program to generate
C               a master information file for a series of closed
C               bomb chamber firings.
C*****
C*****
C               CHARACTER*20 FNAME, FNAME1, A1 (20)*20, A2 (6)*80, MARK (4)*1, ZZ1
C               COMMON A3 (100), P (11, 15, 5)
C*****
C***** INITIALIZATION OF ALL ARRAYS *****
C*****
      DO 1000 I = 1, 20
        A1 (I) = ' '
1000  CONTINUE
      DO 1010 I = 1, 6
        A2 (I) = ' '
1010  CONTINUE
      DO 1020 I = 1, 100
        A3 (I) = 0.0
1020  CONTINUE
      DO 1030 I = 1, 7
        DO 1040 J = 1, 15
          DO 1050 K = 1, 5
            P (I, J, K) = 0.0
1050    CONTINUE
1040  CONTINUE
1030  CONTINUE
      DO 1060 J = 1, 15
        P (8, J, 1) = 1.
        P (8, J, 2) = 1.
1060  CONTINUE
      MARK (1) = ' '
      MARK (2) = ' '
      MARK (3) = ' '
      MARK (4) = ' '
C*****

```

```

C***** WRITE THE INITIAL HEADER INFORMATION *****
C*****
1070 CONTINUE
    CALL CLEAR
    WRITE (*, 6000)
6000 FORMAT(' ',20X,'BRLCB ** VERSION 3.0 ** JANUARY 1992')
    WRITE (*, 6010)
    WRITE (*, 6020)
6010 FORMAT(////,20X,'Creating a Master Information File'/,
1      10X,58(' '))
6020 FORMAT(10X,'This program will create a master information'
1      /,10X,'file which will contain information that does',
2      /,10X,'not change from one firing to the next.')
    WRITE (*, 6030)
6030 FORMAT(///)
C*****
C***** CHECKING TO BUILD FROM EXISTING .MAS FILE *****
C*****
    WRITE (*, *) '      Create .MAS file from existing file? [Y/N]'
    WRITE (*, *) '      Default: Y'
    READ (*, 5000) ZZ
5000 FORMAT(A1)
    IF ((ZZ.EQ. ' ') .OR. (ZZ.EQ. 'Y') .OR. (ZZ.EQ. 'y')) THEN
        WRITE (*, *) 'Enter the file name of the master file from'
        WRITE (*, *) 'which the new master file will be created.'
        WRITE (*, *) 'Include drive and file extension if necessary.'
        READ (*, 5010) FNAME1
5010 FORMAT(A20)
        OPEN (UNIT = 9, FILE = FNAME1, STATUS = 'OLD', ERR = 1080)
        REWIND (UNIT = 9)
        GO TO 1090
1080 CONTINUE
        WRITE (*, 6040)
6040 FORMAT(20X,'The indicated master file does not exist, try again.')
        PAUSE
        GO TO 1070
C*****
C***** READING OLD MASTER FILE TO BUILD FROM *****
C*****
1090 CONTINUE
    DO 1100 I = 1, 6
        READ (9, 5020) A2 (I)
1100 CONTINUE
5020 FORMAT(A80)
    DO 1110 I = 1, 20
        READ (9, 5010) A1 (I)
1110 CONTINUE

```

```

        DO 1120 I = 1, 100
          READ (9, *) A3 (I)
1120   CONTINUE
        DO 1130 I = 1, 11
          DO 1140 J = 1, 15
            DO 1150 K = 1, 5
              READ (9, *) P (I, J, K)
1150      CONTINUE
1140      CONTINUE
1130      CONTINUE
        CLOSE (UNIT = 9)
      END IF
C*****
C***** IDENTIFICATION INFO MUST BE OBTAINED *****
C*****
      CALL IDINFO (FNAME, A1)
      MARK (1) = '*'
C*****
C***** NEW INFORMATION IS OBTAINED *****
C*****
1160 CONTINUE
      CALL CLEAR
      WRITE (*, *) 'Creation of Master File: Main Menu'
      WRITE (*, 6050) MARK (1), MARK (2), MARK (3), MARK (4)
6050  FORMAT(15X,////,
1      /,10X,'1. Identification information ',A1
2      /,10X,'2. Propellant information ',A1
3      /,10X,'3. Igniter information ',A1
4      /,10X,'4. Write the master file ',A1
5      /,10X,'5. Exit          '/)
      WRITE (*, *) ' '
      WRITE (*, 6060)
6060  FORMAT(10X,'An * indicates that the information'/
1,10X,'has been provided.',/10X,'Please Enter Your Choice (1-5):')
      READ (*, *) ISELEC
      IF (ISELEC .EQ. 10) THEN
        CALL PRINTP (A1)
C*****
C***** UNDOCUMENTED FEATURE TO PRINT ARRAY P *****
C*****
        GO TO 1160
      END IF
      IF (ISELEC .GT. 5) THEN
        WRITE (*, *) 'Your choice is not between 1 and 5.'
        PAUSE
        GO TO 1160
      END IF
C*****

```

```

C***** ENDING THE PROGRAM *****
C*****
IF (ISELEC.EQ. 5) GO TO 1170
C*****
C***** THE IDENTIFICATION INFORMATION IS OBTAINED *****
C*****
IF (ISELEC.EQ. 1) THEN
  CALL IDINFO (FNAME, A1)
  MARK (1) = '*'
  GO TO 1160
END IF
C*****
C THE PROPELLANT INFORMATION IS OBTAINED
C*****
IF (ISELEC.EQ. 2) THEN
  CALL CLEAR
  WRITE (*, *) 'Creation of Master File: Propellant Information'
  WRITE (*, 6070)
6070 FORMAT(///,10X,'BRLCB treats all propellants as deterred.',/,
  110X,'To describe a deterred grain up to 15 distinct regions',/,
  210X,'may be specified. For each region all properties are',/,
  310X,'specified at the beginning and end of the region.',/,
  410X,'Intermediate values are obtained by linear interpolation.',/,
  510X,'To describe layered grains the properties at the beginning',
  6/,10X,'and end of a region are the same. A homogeneous grain',/,
  710X,'is a layered grain with one region.'//)
  PAUSE
  CALL CLEAR
  WRITE (*, 6080)
716 FORMAT(///,10X,'If the .MAS file is being built from an existing
  1',/,10X,'.MAS file, it is assumed that either the propellant',/,
  210X,'geometry or thermochemistry is the same as in the .MAS',/,
  310X,'from which the current file is being created. If this is',
  4/,10X,'the case, then the appropriate options on the Propellant'
  5',/,10X,'Information Sub-menu do not have to be run.'//)
  PAUSE
  CALL PROPEL (A1)
  MARK (2) = '*'
  GO TO 1160
END IF
C*****
C***** IGNITER INFORMATION IS OBTAINED *****
C*****
IF (ISELEC.EQ. 3) THEN
  CALL IGNIT (A1)
  MARK (3) = '*'
  GO TO 1160
END IF

```

```

C*****
C***** THE MASTER INFORMATION FILE IS CREATED *****
C*****
      IF (ISELEC .EQ. 4) THEN
C*****
C***** DEPTHS OF THE LAYERS ARE DETERMINED *****
C*****
      NUMPTS = INT (A3 (4) + .5)
      IF (NUMPTS .EQ. 1) THEN
        P (1, 1, 2) = A3 (1)
      ELSE
        P (1, NUMPTS, 2) = A3 (1) - P (1, NUMPTS, 1)
        DO 1180 J = 1, NUMPTS - 1
          P (1, J, 2) = P (1, J + 1, 1) - P (1, J, 1)
1180    CONTINUE
      END IF
      CALL CLEAR
1190  CONTINUE
      OPEN (UNIT = 7, FILE = FNAME, ERR = 1200)
      GO TO 1210
1200  CONTINUE
      WRITE (*, 6090)
6090  FORMAT(10X,'An Error has occurred on opening the master file.'/,
1 10X,'Most likely the file already exists. Your options are:'/,
2 15x,'1. Overwrite existing file.'/,
3 15x,'2. Enter new name.'/,
4 15x,'Enter your choice.')
      READ (*, *) MCHOICE
      IF (MCHOICE .EQ. 1) THEN
        OPEN (UNIT = 7, FILE = FNAME, STATUS = 'OLD', ERR = 1200)
        GO TO 1210
      ELSE
        WRITE (*, 6100)
6100  FORMAT(10X,'Enter the new file name, include drive'/,
1 10X,'and extension.')
        READ (*, 5010) FNAME
        A1 (3) = FNAME
        GO TO 1190
      END IF
1210  CONTINUE
      DO 1220 I = 1, 6
        WRITE (7, 6110) A2 (I)
6110  FORMAT(A80)
1220  CONTINUE
      DO 1230 I = 1, 20
        WRITE (7, 6120) A1 (I)
6120  FORMAT(A20)
1230  CONTINUE

```

```

        DO 1240 I = 1, 100
          WRITE (7, *) A3 (I)
1240    CONTINUE
        DO 1250 I = 1, 11
          DO 1260 J = 1, 15
            DO 1270 K = 1, 5
              WRITE (7, *) P (I, J, K)
1270          CONTINUE
1260        CONTINUE
1250      CONTINUE
        MARK (4) = '*'
        CLOSE (UNIT = 7)
        CALL CLEAR
        WRITE (*, 6130) A1 (3)
6130    FORMAT(////,10X,'*****',
1/,10X,'*',
2/,10X,'*   The Master File Has Been Written   *',
3/,10X,'*   The name is: ',A20,'*',
4/,10X,'*****')
        WRITE (*, 6140)
6140    FORMAT(////)
        PAUSE
        GO TO 1160
      END IF
C*****
C***** EXIT FROM PROGRAM *****
C*****
1170 CONTINUE
      IEXIT = 1
      CALL CLEAR
      IF (MARK (1) .NE. '*') THEN
        WRITE (*, *)
1        'The identification information has not been entered.'
        WRITE (*, *) 'Proceed with exit (1) or return (2)? '
        READ (*, *) IEXIT
      END IF
      IF (MARK (2) .NE. '*') THEN
        WRITE (*, *) 'The propellant information has not been entered.'
        WRITE (*, *) 'Proceed with exit (1) or return (2)? '
        READ (*, *) IEXIT
      END IF
      IF (MARK (3) .NE. '*') THEN
        WRITE (*, *) 'The igniter information has not been entered.'
        WRITE (*, *) 'Proceed with exit (1) or return (2)? '
        READ (*, *) IEXIT
      END IF
      IF (MARK (4) .NE. '*') THEN
        WRITE (*, *) 'The master file has not been written.'

```

```

        WRITE (*, *) 'Proceed with exit (1) or return (2)? '
        READ (*, *) IEXIT
    END IF
    IF (IEXIT .NE. 1) GO TO 1160
    STOP
    END
C*****
C***** SUBROUTINE CLEAR *****
C*****
    SUBROUTINE CLEAR
    CHARACTER ST*4
    DATA ST/' [2J'/
    WRITE (*, 6000) ST
    6000 FORMAT (1X,A4)
    RETURN
    END
C*****
C***** SUBROUTINE IDINFO *****
C*****
    SUBROUTINE IDINFO (FNAME, A1)
    CHARACTER*20 DRIVE, TNAME, FNAME, A, A1 (20), ZZ1
C*****
C  Version: 3.0, January 1992
C
C  Written by : William Oberle, Ballistic Research Laboratory
C
C  This subroutine will ask for identification information and determine
C  the file name under which the information is to be stored.
C*****
    CALL CLEAR
    WRITE (*, 6000)
    6000 FORMAT(' ', 'Creation of Master File: Identification Information',
    1//////)
    WRITE (*, 6010)
    6010 FORMAT(' ', 10X, 'Enter the project name:')
    READ (*, 5000) ZZ1
    5000 FORMAT(A20)
    A1 (1) = ZZ1
    CALL CLEAR
    WRITE (*, 6000)
    WRITE (*, 6020)
    6020 FORMAT(10X, 'Enter the name of the person requesting the work:')
    READ (*, 5000) ZZ1
    A1 (2) = ZZ1
    CALL CLEAR
    WRITE (*, 6000)
    1000 CONTINUE
    WRITE (*, 6030)

```

```

6030  FORMAT(/,10X,'Enter the name for the Master Information',
      1/,10X,'File. All DOS path conventions can be utilized. It is',
      2/,10X,'suggested that the extension of the file be .MAS.')
      READ (*, 5010) FNAME
5010  FORMAT(A20)
      OPEN (UNIT = 12, FILE = FNAME, STATUS = 'NEW', ERR = 1010)
      A1 (3) = FNAME
      CLOSE (UNIT = 12)
      GO TO 1020
1010  CONTINUE
      WRITE (*, 6040)
6040  FORMAT(10X,'WARNING: The indicated file already exists.'
      1/,15x,'1. Overwrite'/,15x,'2. Enter new name.'/,
      215x,'Enter your choice.')
      READ (*, *) IH
      IF (IH .EQ. 2) THEN
        GO TO 1000
      ELSE
        A1 (3) = FNAME
      END IF
1020  CONTINUE
      RETURN
      END

```

```

C*****
C***** SUBROUTINE IGNIT *****
C*****

```

```

      SUBROUTINE IGNIT (A1)
      COMMON A3 (100), P (11, 15, 5)
      CHARACTER*20 MARK (2)*1, A1 (20), A5 (10, 3),
1    CTEMP, CTEMP1*15, ZZ1
      DIMENSION B (10, 6)
      MARK (1) = ' '
      MARK (2) = ' '

```

```

C*****

```

C Version: 3.0, January 1992

C Written by: William Oberle, Ballistic Research Laboratory

C an extension of a routine written by

C Kristopher Koehnen

C Applied Combustion Technology, Inc.

C

C The purpose of this program is to determine the values for the

C thermochemistry and pertinent data for the igniter used in the

C closed chamber firing.

```

C*****

```

```

1000 CONTINUE

```

```

      CALL CLEAR

```

```

C*****

```

```

C***** DETERMINATION OF THE TYPE OF IGNITER BEING USED *****

```

```

C*****
WRITE (*, 6000)
6000 FORMAT(' ', 'Creation of Master File: Igniter Information', '/')
WRITE (*, 6010)
6010 FORMAT(10X, 'Information concerning the type of igniter',
1/, 10X, 'being used is needed. The available options are:', '/',
215X, ' 1. Black Powder ', '/',
315X, ' 2. Enter New Information', '/',
415X, ' 3. Use Igniter Library ', '/',
515X, ' 4. Exit', '/')
1010 CONTINUE
WRITE (*, *) 'Please Enter Your Choice: '
READ (*, *) IGC
IF (IGC .GT. 4) THEN
WRITE (*, *) 'Your choice is not between 1 and 4.'
GO TO 1010
END IF
C*****
C***** SELECTION OF OPTION IS MADE *****
C*****
GO TO ( 1020, 1030, 1040, 1050) IGC
C*****
C***** USING BLACK POWDER *****
C*****
1020 CONTINUE
A1 (11) = 'Black Powder'
A1 (12) = 'FFFG '
A1 (13) = 'Pellets, Milan Ord.'
A3 (13) = 290.
A3 (14) = 2188.
A3 (15) = 1.75
A3 (16) = 66.37
A3 (17) = .785
A3 (18) = 1.2184
C*****
C***** EXITING THE ROUTINE *****
C*****
1060 CONTINUE
CALL CLEAR
WRITE (*, *) ' '
WRITE (*, *) ' You have selected black powder as the '
WRITE (*, *) ' type igniter which is being used. Your'
WRITE (*, *) ' options are:'
WRITE (*, *) ' '
WRITE (*, *) ' 1. Accept Black Powder and Exit'
WRITE (*, *) ' 2. Review Black Powder Information'
WRITE (*, *) ' 3. Enter Different Igniter Information'
WRITE (*, *) ' '

```

1070 CONTINUE

WRITE (*, *) ' Please Enter Your Choice: '

READ (*, *) IANSWER

IF (IANSWER .GT. 3) THEN

WRITE (*, *) 'Your choice is not between 1 and 3.'

GO TO 1070

END IF

IF (IANSWER .EQ. 1) RETURN

IF (IANSWER .EQ. 3) GO TO 1000

C*****

C***** BLACK POWDER REVIEWING INFORMATION *****

C*****

CALL CLEAR

WRITE (*, *)

WRITE (*, *) ' Information for black powder.'

WRITE (*, *) ' '

WRITE (*, 6020)

WRITE (*, 6030)

WRITE (*, 6040)

WRITE (*, 6050) A1 (11)

WRITE (*, 6060) A1 (12)

WRITE (*, 6070) A1 (13)

PAUSE

CALL CLEAR

WRITE (*, 6020)

WRITE (*, *) ' '

WRITE (*, *) 'This is the thermochemical data'

WRITE (*, *) 'for black powder.'

WRITE (*, *) ' '

WRITE (*, 6020)

WRITE (*, 6030)

WRITE (*, 6040)

WRITE (*, 6080) A3 (13)

WRITE (*, 6090) A3 (14)

WRITE (*, 6100) A3 (15)

WRITE (*, 6110) A3 (16)

WRITE (*, 6120) A3 (17)

WRITE (*, 6130) A3 (18)

WRITE (*, 6020)

WRITE (*, *) ' '

PAUSE

GO TO 1060

C*****

C***** ENTERING NEW INFORMATION *****

C*****

1030 CONTINUE

CALL CLEAR

WRITE (*, *) ' '

```

WRITE (*, *) 'There are two sets of data which must be'
WRITE (*, *) 'entered for the igniter.'
WRITE (*, *) 'An * indicates that the information has entered.'
WRITE (*, *) 'They are:'
WRITE (*, 6140) MARK (1)
WRITE (*, 6150) MARK (2)
6140 FORMAT('    1) Identification ',A1)
6150 FORMAT('    2) Thermochemical ',A1)
WRITE (*, *) '    3) Exit'
WRITE (*, *) ' '
1080 CONTINUE
WRITE (*, *) 'Please Enter Your Choice: '
READ (*, *) ICHOICE
IF (ICHOICE .GT. 3) THEN
WRITE (*, *) 'Your choice is not between 1 and 3.'
GO TO 1080
END IF
CALL CLEAR
IF (ICHOICE .EQ. 3) THEN
WRITE (*, *) 'Add the new information to the igniter library?'
WRITE (*, *) 'Enter 1 to add to library else a 2. '
READ (*, *) IGADD
IF (IGADD .EQ. 1) CALL IGNADD (A1)
GO TO 1000
END IF
C*****
C***** The program will branch to the choice that was made. *****
C*****
GO TO ( 1090, 1100) ICHOICE
C*****
C***** The identification data is read in interactively. *****
C*****
1090 CONTINUE
CALL CLEAR
MARK (1) = '*'
WRITE (*, *) ' '
WRITE (*, *) 'For each question, enter the'
WRITE (*, *) 'information for the igniter.'
WRITE (*, *) ' '
WRITE (*, 6160)
6160 FORMAT(' ', 'Enter Igniter Name:')
READ (*, 5000) ZZ1
A1 (11) = ZZ1
WRITE (*, *) ' '
WRITE (*, 6170)
6170 FORMAT(' ', 'Enter Igniter Lot: ')
READ (*, 5000) ZZ1
A1 (12) = ZZ1

```

```

WRITE (*, *) ' '
WRITE (*, 6180)
6180 FORMAT(' ', 'Enter Igniter Source:')
READ (*, 5000) ZZ1
A1 (13) = ZZ1
CALL CLEAR
WRITE (*, *) ' '
WRITE (*, *) 'This is the identification information',
1 ' that you have entered.'
WRITE (*, 6020)
WRITE (*, 6030)
WRITE (*, 6040)
WRITE (*, 6050) A1 (11)
WRITE (*, 6060) A1 (12)
WRITE (*, 6070) A1 (13)
WRITE (*, 6020)
PAUSE
GO TO 1030
C*****
C***** The thermochemical data is entered interactively. *****
C*****
1100 CONTINUE
CALL CLEAR
MARK (2) = '*'
WRITE (*, *) 'For each question, enter the'
WRITE (*, *) 'information for the igniter.'
WRITE (*, *) ' '
WRITE (*, *) 'Theoretical Impetus (J/gram): '
READ (*, *) A3 (13)
WRITE (*, *) 'Flame Temperature (K): '
READ (*, *) A3 (14)
WRITE (*, *) 'Density (gram/cc): '
READ (*, *) A3 (15)
A3 (16) = 8.314*A3 (14)/A3 (13)
WRITE (*, 6190) A3 (16)
6190 FORMAT(/, ' ', 'The molecular weight consistent with the Flame',
1/, ' ', 'Temperature and Impetus is:', F8.4,
2/, ' 1. Accept Value', '/', ' 2. Enter different value')
READ (*, *) IMAYBE
IF (IMAYBE .EQ. 2) THEN
WRITE (*, *) 'Enter value for Average Molecular Weight: '
READ (*, *) A3 (16)
END IF
WRITE (*, *) 'Co-Volume (cc/gram): '
READ (*, *) A3 (17)
WRITE (*, *) 'Ratio of Specific Heats (gamma): '
READ (*, *) A3 (18)
CALL CLEAR

```

```

WRITE (*, *) ' '
WRITE (*, *) 'This is the thermochemical data you have entered.'
WRITE (*, 6020)
WRITE (*, 6030)
WRITE (*, 6040)
WRITE (*, 6080) A3 (13)
WRITE (*, 6090) A3 (14)
WRITE (*, 6100) A3 (15)
WRITE (*, 6110) A3 (16)
WRITE (*, 6120) A3 (17)
WRITE (*, 6130) A3 (18)
WRITE (*, 6020)
WRITE (*, *) ' '
PAUSE
GO TO 1030
C*****
C***** FORMAT STATEMENTS *****
C*****
6030 FORMAT(38X,'Igniter')
6040 FORMAT(38X,'-----')
6050 FORMAT(10X,'Type:',23X,A20)
6060 FORMAT(10X,'Lot:',24X,A20)
6070 FORMAT(10X,'Source:',21X,A20)
8000 FORMAT(10X,'Weight (grams):')
8010 FORMAT(10X,'Initial Temp. (K):')
8020 FORMAT(' ')
6080 FORMAT(10X,'Theo. Impetus - (J/gram):',F15.5)
6090 FORMAT(10X,'Flame Temp. ----- (K):',F15.5)
6100 FORMAT(10X,'Density ----- (gram/cc):',F15.5)
6110 FORMAT(10X,'Avg. Mol. Wt. ----- :',F15.5)
6120 FORMAT(10X,'Co-Volume ---- (cc/gram):',F15.5)
6130 FORMAT(10X,'Gamma ----- :',F15.5)
8030 FORMAT(' ')
5000 FORMAT(A20)
6020 FORMAT(80(' '))
C*****
C***** USING THE IGNITER LIBRARY *****
C*****
1040 CONTINUE
CALL CLEAR
OPEN (UNIT = 3, FILE = 'IGNITER.INF')
READ (3, *) NUM
WRITE (*, *) ' Igniter Library'
WRITE (*, *) ' '
WRITE (*, 6200)
479 FORMAT(' ', # Type Impetus Flame Temp Dcnsity Mol
1Wt. Covolume Gamma')
WRITE (*, 6210)

```

```

DO 1110 I = 1, NUM
  READ (3, 5000) A5 (I, 1)
  CTEMP = A5 (I, 1)
  CTEMP1 = CTEMP (1:15)
  READ (3, 5000) A5 (I, 2)
  READ (3, 5000) A5 (I, 3)
  READ (3, *) (B (I, J), J = 1, 6)
  WRITE (*, 6220) I, CTEMP1, (B (I, J), J = 1, 6)
6220 FORMAT(' ', I2, 1X, A15, 6F9.3)
1110 CONTINUE
  CLOSE (UNIT = 3)
6210 FORMAT(75('-'))
1120 CONTINUE
  WRITE (*, *) ' '
  WRITE (*, *) 'Enter the number of the igniter you want to use.'
  WRITE (*, *) 'To delete an entry, enter 11.'
  READ (*, *) IGNU
  IF (IGNUM .GT. 11) THEN
    WRITE (*, *) 'Your choice is not between 1 and 11.'
    GO TO 1120
  END IF
C*****
C***** Any data from the igniter library will be deleted. *****
C*****
  IF (IGNUM .EQ. 11) THEN
    WRITE (*, *) 'Enter number to delete.'
    READ (*, *) IDEL
    OPEN (3, FILE = 'IGNITER.INF')
    WRITE (3, *) NUM - 1
    DO 1130 , I = 1, NUM
      IF (I .EQ. IDEL) GO TO 1130
      WRITE (3, 5000) A5 (I, 1)
      WRITE (3, 5000) A5 (I, 2)
      WRITE (3, 5000) A5 (I, 3)
      WRITE (3, *) (B (I, J), J = 1, 6)
1130 CONTINUE
    CLOSE (UNIT = 3)
    GO TO 1000
  END IF
  A1 (11) = A5 (IGNUM, 1)
  A1 (12) = A5 (IGNUM, 2)
  A1 (13) = A5 (IGNUM, 3)
  A3 (13) = B (IGNUM, 1)
  A3 (14) = B (IGNUM, 2)
  A3 (15) = B (IGNUM, 3)
  A3 (16) = B (IGNUM, 4)
  A3 (17) = B (IGNUM, 5)
  A3 (18) = B (IGNUM, 6)

```

```

      GO TO 1000
1050 CONTINUE
      RETURN
      END
C*****
C***** SUBROUTINE IGNADD *****
C*****
      SUBROUTINE IGNADD (A1)
      COMMON A3 (100), P (11, 15, 5)
      CHARACTER*20 A1 (20), A5 (10, 3)
      DIMENSION B (10, 6)
      CALL CLEAR
      OPEN (UNIT = 3, FILE = 'IGNITER.INF')
      READ (3, *) NUM
      IF (NUM .EQ. 10) THEN
        WRITE (*, *) 'The igniter library is full,'
        WRITE (*, *) 'delete an entry.'
        CLOSE (UNIT = 3)
        RETURN
      END IF
      DO 1000 I = 1, NUM
        READ (3, 5000) A5 (I, 1)
        READ (3, 5000) A5 (I, 2)
        READ (3, 5000) A5 (I, 3)
5000  FORMAT(A20)
        READ (3, *) (B (I, J), J = 1, 6)
1000 CONTINUE
        CLOSE (UNIT = 3)
        NUM = NUM + 1
        A5 (NUM, 1) = A1 (11)
        A5 (NUM, 2) = A1 (12)
        A5 (NUM, 3) = A1 (13)
        B (NUM, 1) = A3 (13)
        B (NUM, 2) = A3 (14)
        B (NUM, 3) = A3 (15)
        B (NUM, 4) = A3 (16)
        B (NUM, 5) = A3 (17)
        B (NUM, 6) = A3 (18)
        OPEN (UNIT = 3, FILE = 'IGNITER.INF')
        WRITE (3, *) NUM
        DO 1010 I = 1, NUM
          WRITE (3, 5000) A5 (I, 1)
          WRITE (3, 5000) A5 (I, 2)
          WRITE (3, 5000) A5 (I, 3)
          WRITE (3, *) (B (I, J), J = 1, 6)
1010 CONTINUE
          CLOSE (UNIT = 3)
          RETURN

```

```

END
C*****
C***** SUBROUTINE PROPEL *****
C*****
SUBROUTINE PROPEL (A1)
COMMON A3 (100), P (11, 15, 5)
CHARACTER*20 A1 (20), SMARK (3)*1, GTYPE*14
C*****
C Version: 3.0, January 1992
C
C   This program requests information concerning the propellant
C   and its geometry. The grain geometry is checked for consistency
C   and changes are computed if the information is not accurate.
C*****
SMARK (1) = ' '
SMARK (2) = ' '
SMARK (3) = ' '
1000 CONTINUE
CALL CLEAR
WRITE (*, 6000)
6000 FORMAT(///,10X,' Propellant Information' /,
110X,' Sub Menu' /,
210X,'Three sets of information must be entered' /,
310X,'for the propellant.' //,
410X,'An * indicates that the information has been entered.' //)
WRITE (*, 6010) SMARK (1)
WRITE (*, 6020) SMARK (2)
WRITE (*, 6030) SMARK (3)
6010 FORMAT(10X,' 1) Identification ',A1)
6020 FORMAT(10X,' 2) Thermochemical ',A1)
6030 FORMAT(10X,' 3) Geometry ',A1,/
110X,' 4) Exit' //,
210X,'Please Enter Your Choice (1-4): ')
READ (*, *) ICHOICE
IF (ICHOICE .GT. 4) THEN
WRITE (*, 6040)
6040 FORMAT(///,'YOUR CHOICE IS NOT BETWEEN 1 AND 4.')
GO TO 1000
END IF
C*****
C***** CHECKING TO RETURN*****
C*****
IF (ICHOICE .EQ. 4) THEN
CALL MAXDEPTH
CALL COMMASS
RETURN
END IF
C*****

```

```

C***** The program will branch to the choice that was made.*****
C*****
GO TO ( 1010, 1020, 1030) ICHOICE
C*****
C***** The identification data is read in interactively. *****
C*****
1010 CONTINUE
      SMARK (1) = '*'
      CALL CLEAR
      WRITE (*, 6050)
720  FORMAT(10X,'For each question, enter the information for the prop
      ellant.']/)
      WRITE (*, 6060)
6060  FORMAT(10X,'Propellant Type: ')
      READ (*, 5000) A1 (6)
      WRITE (*, 6070)
6070  FORMAT(/,10X,'Propellant Lot: ')
      READ (*, 5000) A1 (7)
      WRITE (*, 6080)
6080  FORMAT(/,10X,'Propellant Source: ')
      READ (*, 5000) A1 (8)
      CALL CLEAR
      WRITE (*, *) ' '
      WRITE (*, *) 'This is the identification information entered.'
      WRITE (*, 6090)
      WRITE (*, 6100)
      WRITE (*, 6110)
      WRITE (*, 6120) A1 (6)
      WRITE (*, 6130) A1 (7)
      WRITE (*, 6140) A1 (8)
      WRITE (*, 6090)
      WRITE (*, 6150)
6150  FORMAT(///)
      PAUSE
      GO TO 1000
C*****
C***** Thermochemical data is entered *****
C*****
1020 CONTINUE
      SMARK (2) = '*'
      CALL CLEAR
      CALL LAYDET
      GO TO 1000
C*****
C***** The geometry is inputted interactively.
*****
C*****
1030 CONTINUE

```

```

SMARK (3) = '*'
CALL CLEAR
WRITE (*, 6160)
6160 FORMAT(////)
WRITE (*, *) ' 1) Sphere'
WRITE (*, *) ' 2) Cord'
WRITE (*, *) ' 3) Rectangular Strip'
WRITE (*, *) ' 4) 1-Perforated Cylinder'
WRITE (*, *) ' 5) Slotted Tube'
WRITE (*, *) ' 6) 7-Perforation Cylinder'
WRITE (*, *) ' 7) 7-Perforation Hexagonal'
WRITE (*, *) ' 8) 19-Perforation Cylinder'
WRITE (*, *) ' 9) 19-Perforation Hexagonal'
WRITE (*, *) '10) 37-Perforation Hexagonal'
WRITE (*, *) '11) Cord With Inhibited Ends'
WRITE (*, *) '12) Sandwich With Inhibited Sides'
WRITE (*, *) '13) Cigarette'
WRITE (*, *) ' '
WRITE (*, *) 'Enter the number corresponding to the '
WRITE (*, *) 'grain type for the propellant: '
READ (*, *) IGTP
A1 (9) = GTYPE (IGTP)
A3 (45) = IGTP
CALL CLEAR
WRITE (*, 6160)
WRITE (*, *) 'For each question, enter the information'
WRITE (*, *) 'for the propellant grain geometry, all '
WRITE (*, *) 'dimensions in cm.'
WRITE (*, *)
PAUSE
CALL GRAIN (IGTP)
CALL CKGRAIN (IGTP)
CALL CLEAR
WRITE (*, 6170)
6170 FORMAT(////)
WRITE (*, *) 'This is the grain geometry that you have entered.'
WRITE (*, 6090)
WRITE (*, 6100)
WRITE (*, 6110)
WRITE (*, 6180) A1 (9)
IF (IGTP .EQ. 1) THEN
    WRITE (*, 6190) A3 (8)
    GO TO 1040
END IF
IF ((IGTP .EQ. 2) .OR. (IGTP .EQ. 11) .OR. (IGTP .EQ. 13)) THEN
    WRITE (*, 6200) A3 (7)
    WRITE (*, 6190) A3 (8)
    GO TO 1040

```

```

END IF
IF ((IGTP .EQ. 3) .OR. (IGTP .EQ. 12)) THEN
  WRITE (*, 6200) A3 (7)
  WRITE (*, 6210) A3 (8)
6210  FORMAT(10X,'Width      (cm.):',F21.5)
  WRITE (*, 6220) A3 (10)
6220  FORMAT(10X,'Thickness (cm.):',F21.5)
  GO TO 1040
END IF
IF ((IGTP .EQ. 4) .OR. (IGTP .EQ. 5)) THEN
  WRITE (*, 6200) A3 (7)
  WRITE (*, 6190) A3 (8)
  WRITE (*, 6230) A3 (9)
  IF (IGTP .EQ. 4) THEN
    WRITE (*, 6240) A3 (10)
6240    FORMAT(10X,'Web      (cm.):',F21.5)
  END IF
  IF (IGTP .EQ. 5) THEN
    WRITE (*, 6250) A3 (11)
6250    FORMAT(10X,'Slot Width (cm.):',F21.5)
  END IF
  GO TO 1040
END IF
IF ((IGTP .EQ. 6) .OR. (IGTP .EQ. 7)) THEN
  WRITE (*, 6200) A3 (7)
  WRITE (*, 6190) A3 (8)
  WRITE (*, 6230) A3 (9)
  WRITE (*, 6260) A3 (10)
  WRITE (*, 6270) A3 (12)
  GO TO 1040
END IF
IF ((IGTP .GE. 8) .AND. (IGTP .LE. 10)) THEN
  WRITE (*, 6200) A3 (7)
  WRITE (*, 6190) A3 (8)
  WRITE (*, 6230) A3 (9)
  WRITE (*, 6260) A3 (10)
  WRITE (*, 6280) A3 (11)
  WRITE (*, 6270) A3 (12)
  GO TO 1040
END IF
1040 CONTINUE
  WRITE (*, 6090)
  WRITE (*, *) ' '
  PAUSE
  GO TO 1000
8000 FORMAT(10X,'Propellant Type: 'A18)
6100 FORMAT(38X,'Propellant')
6110 FORMAT(38X,'-----')

```

```

6120 FORMAT(10X,'Type:',23X,A19)
6130 FORMAT(10X,'Lot:',24X,A19)
6140 FORMAT(10X,'Source:',21X,A19)
8010 FORMAT(10X,'Weight (grams):')
8020 FORMAT(10X,'Initial Temp. (K):')
8030 FORMAT(' ')
8040 FORMAT(10X,'Theo. Impetus - (J/gram):',F15.5)
8050 FORMAT(10X,'Flame Temp. ----- (K):',F15.5)
8060 FORMAT(10X,'Density ----- (gram/cc):',F15.5)
8070 FORMAT(10X,'Avg. Mol. Wt. ----- :',F15.5)
8080 FORMAT(10X,'Co-Volume ---- (cc/gram):',F15.5)
8090 FORMAT(10X,'Gamma ----- :',F15.5)
8100 FORMAT(' ')
6180 FORMAT(10X,'Grain Type:',16X,A14)
6200 FORMAT(10X,'Length --- (cm.):',F21.5)
6190 FORMAT(10X,'Outer Dia. (cm.):',F21.5)
6230 FORMAT(10X,'Perf. Dia. (cm.):',F21.5)
6260 FORMAT(10X,'Inner Web (cm.):',F21.5)
6280 FORMAT(10X,'Middle Web (cm.):',F21.5)
6270 FORMAT(10X,'Outer Web (cm.):',F21.5)
5000 FORMAT(A20)
6090 FORMAT(80('-'))
      RETURN
      END

```

```

C*****
C***** This function will determine the grain type. *****
C*****

```

```

      FUNCTION GTYPE (I)
      CHARACTER GTYPE*14
      IF (I .EQ. 1) GTYPE = 'Sphere'
      IF (I .EQ. 2) GTYPE = 'Cord'
      IF (I .EQ. 3) GTYPE = 'Strip'
      IF (I .EQ. 4) GTYPE = '1-Perf. Cyl.'
      IF (I .EQ. 5) GTYPE = 'Slotted Tube'
      IF (I .EQ. 6) GTYPE = '7-Perf. Cyl.'
      IF (I .EQ. 7) GTYPE = '7-Perf. Hex.'
      IF (I .EQ. 8) GTYPE = '19-Perf. Cyl.'
      IF (I .EQ. 9) GTYPE = '19-Perf. Hex.'
      IF (I .EQ. 10) GTYPE = '37-Perf. Hex.'
      IF (I .EQ. 11) GTYPE = 'Cord Inh. End'
      IF (I .EQ. 12) GTYPE = 'Sandwich'
      IF (I .EQ. 13) GTYPE = 'Cigarette'
      RETURN
      END

```

```

C*****
C***** SUBROUTINE GRAIN *****
C*****
      SUBROUTINE GRAIN (ITYPE)

```

```

COMMON A3 (100), P (11, 15, 5)
C*****
C Version: 3.0, January 1992
C
C This subroutine will interactively ask the user for information
C concerning the geometry of the propellant being used in the
C analysis.
C*****
C***** Setting grain dimensions to zero *****
C*****
DO 1000 I = 7, 12
    A3 (I) = 0.0
1000 CONTINUE
CALL CLEAR
C*****
C***** INPUT FOR A RECTANGULAR STRIP GRAIN *****
C*****
IF ((ITYPE .EQ. 3) .OR. (ITYPE .EQ. 12)) THEN
    IF (ITYPE .EQ. 12) THEN
        WRITE (*, *) 'For the sandwich grain, burning is only'
        WRITE (*, *) 'on the top and bottom surfaces. These'
        WRITE (*, *) 'surfaces are determined by the length'
        WRITE (*, *) 'and thickness. Burning is parallel to'
        WRITE (*, *) 'the width dimension.'
        WRITE (*, *) ' '
    END IF
    WRITE (*, *) 'Enter the Length: '
    READ (*, *) A3 (7)
    WRITE (*, *) 'Enter the Width: '
    READ (*, *) A3 (8)
    WRITE (*, *) 'Enter the Thickness: '
    READ (*, *) A3 (10)
    RETURN
END IF
C*****
C** INPUT OF THE GRAIN DIAMETER -- REQUIRED OF ALL REMAINING GRAINS ****
C*****
WRITE (*, *) 'Enter the Grain Diameter: '
READ (*, *) A3 (8)
C*****
C***** TEST FOR SPHERE GEOMETRY -- ONLY NEED DIAMETER *****
C*****
IF (ITYPE .EQ. 1) RETURN
C*****
C***** INPUT OF GRAIN LENGTH *****
C*****
WRITE (*, *) 'Enter the Grain Length: '
READ (*, *) A3 (7)

```

```

C*****
C***** TEST FOR CORD GEOMETRY -- ONLY NEED DIAMETER AND LENGTH *****
C*****
      IF ((ITYPE .EQ. 2) .OR. (ITYPE .EQ. 11) .OR. (ITYPE .EQ. 13))
        1 RETURN
C*****
C***** INPUT OF PERF DIAMETER *****
C*****
      WRITE (*, *) 'Enter the Perf Diameter: '
      READ (*, *) A3 (9)
C*****
C***** SPECIAL INPUT FOR SLOTTED TUBE
*****
C*****
      IF (ITYPE .EQ. 5) THEN
        WRITE (*, *) 'Enter the Slot Width: '
        READ (*, *) A3 (11)
        RETURN
      END IF
C*****
C***** INPUT OF INNER WEB *****
C*****
      WRITE (*, *) 'Enter the Inner Web: '
      READ (*, *) A3 (10)
C*****
C*** TEST FOR SINGLE PERF GEOMETRY -- NO ADDITIONAL INPUT REQUIRED ***
C*****
      IF (ITYPE .EQ. 4) RETURN
C*****
C***** TEST FOR SEVEN PERF GEOMETRY -- MIDDLE WEB NOT NEEDED *****
C*****
      IF ((ITYPE .EQ. 6) .OR. (ITYPE .EQ. 7)) GO TO 1010
C*****
C***** INPUT OF MIDDLE WEB *****
C*****
      WRITE (*, *) 'Enter the Middle Web: '
      READ (*, *) A3 (11)
C*****
C***** INPUT OF OUTER WEB *****
C*****
1010 CONTINUE
      WRITE (*, *) 'Enter the Outer Web: '
      READ (*, *) A3 (12)
      RETURN
      END
C*****
C***** SUBROUTINE CKGRAIN *****
C*****

```

SUBROUTINE CKGRAIN (ITYPE)

COMMON A3 (100), P (11, 15, 5)

C*****

C Version: 3.0, January 1992

C

C This program will check to determine if the entered values for the
C grain geometry are correct. The error tolerance is 1%. If the
C geometry is not within 1% then the webs are adjusted to make the
C geometry consisted. The user is prompted as to accept the new grain
C geometry or keep the original values.

C*****

C***** NO CHECKING IS DONE FOR CORD, RECTANGULAR STRIP, SPHERE, OR ****

C***** SLOTTED TUBE. *****

C*****

XL = A3 (7)

XD = A3 (8)

XPD = A3 (9)

XIW = A3 (10)

XMW = A3 (11)

XOW = A3 (12)

IF ((ITYPE .EQ. 1) .OR. (ITYPE .EQ. 2)) RETURN

IF ((ITYPE .EQ. 3) .OR. (ITYPE .EQ. 5)) RETURN

IF (ITYPE .GE. 11) RETURN

C*****

C***** CHECKING A SEVEN PERF GRAIN *****

C*****

IF ((ITYPE .EQ. 6) .OR. (ITYPE .EQ. 7)) THEN

XDT = 3.*XPD + 2.*(XIW + XOW)

DELTA = XDT - XD

IF (ABS (DELTA/XD) .GE. 0.0001) THEN

XIWT = XIW - DELTA/4.

XOWT = XOW - DELTA/4.

XMWT = XMW

GO TO 1000

END IF

RETURN

END IF

C*****

C***** CHECKING A SINGLE PERF GRAIN *****

C*****

IF (ITYPE .EQ. 4) THEN

XDT = XPD + 2.*XIW

DELTA = XDT - XD

IF (ABS (DELTA/XD) .GE. 0.0001) THEN

XIWT = XIW - DELTA/2.

XMWT = XMW

XOWT = XOW

GO TO 1000

```

      END IF
      RETURN
    END IF
C*****
C***** CHECKING A 37-PEFR HEX GRAIN *****
C*****
    IF (ITYPE .EQ. 10) THEN
      XDT = 7.*XPD + 6.*XIW + 2.*XOW
      DELTA = XDT - XD
      IF (ABS (DELTA/XD) .GE. 0.0001) THEN
        XIWT = XIW - DELTA/8.
        XOWT = XOW - DELTA/8.
        XMWT = XMW
        GO TO 1000
      END IF
      RETURN
    END IF
C*****
C***** CHECKING 19-PERF HEX *****
C*****
    IF (ITYPE .EQ. 9) THEN
      XDT = 5.*XPD + 4.*XIW + 2.*XOW
      DELTA = XDT - XD
      IF (ABS (DELTA/XD) .GE. 0.0001) THEN
        XIWT = XIW - DELTA/6.
        XOWT = XOW - DELTA/6.
        XMWT = XMW
        GO TO 1000
      END IF
      RETURN
    END IF
C*****
C***** CHECKING 19 PERF *****
C*****
    IF (ITYPE .EQ. 8) THEN
      XDT = 5*XPD + 2.*(XIW + XMW + XOW)
      DELTA = XDT - XD
      IF (ABS (DELTA/XD) .GE. 0.0001) THEN
        DELTA = DELTA/6.
        XIWT = XIW - DELTA
        XMWT = XMW - DELTA
        XOWT = XOW - DELTA
        GO TO 1000
      END IF
      RETURN
    END IF
C*****
C***** USER IS ALLOWED TO ACCEPT OR REJECT NEW DIMENSIONS *****

```

```

C*****
  CALL CLEAR
1000 CONTINUE
  WRITE (*, *) ' '
  WRITE (*, *) ' '
  WRITE (*, *) 'WARNING!! The grain geometry entered is not'
  WRITE (*, *) 'consistent. The following values have been'
  WRITE (*, *) 'recomputed for the grain geometry.'
  WRITE (*, *) ' '
  WRITE (*, 6000) XL, XD, XPD, XIWT, XMWT, XOWT
6000 FORMAT('      Grain Length : ',F10.5/,
1      '      Diameter   : ',F10.5/,
2      '      Perf Diameter: ',F10.5/,
3      '      Inner Web   : ',F10.5/,
4      '      Middle Web  : ',F10.5/,
5      '      Outer Web   : ',F10.5)
  WRITE (*, *) ' '
  WRITE (*, *) ' 1. Accept New Geometry'
  WRITE (*, *) ' 2. Retain Values as Entered'
  WRITE (*, *) ' '
  WRITE (*, *) ' Please Enter Your Choice: '
  READ (*, *) ICHOICE
  IF (ICHOICE.EQ. 2) RETURN
C*****
C***** CHANGING VALUES TO NEW GEOMETRY *****
C*****
  A3 (10) = XIWT
  A3 (11) = XMWT
  A3 (12) = XOWT
  RETURN
  END
C*****
C*****SUBROUTINE LAYDET *****
C*****
  SUBROUTINE LAYDET
C*****
C Version: 3.0, January 1992
C
C Purpose: This program provides the necessary utilities to prepare the
C          files for use with layered/deterred propellant.
C*****
  COMMON A3 (100), P (11, 15, 5)
  CALL CLEAR
  WRITE (*, 6000)
6000 FORMAT(////)
  WRITE (*, *) 'Enter the number of layers (1 - 15)'
  READ (*, *) NUMPT
  A3 (4) = NUMPT

```

```

P (1, 1, 1) = 0.0
IF (NUMPT .EQ. 1) GO TO 1000
WRITE (*, *) ' '
WRITE (*, *) 'Enter the starting depth for each layer.'
WRITE (*, *) 'The first layer starts at a depth of 0 cm.'
WRITE (*, *) 'and will be automatically entered. The '
WRITE (*, *) 'last layer must start at a depth no deeper'
WRITE (*, *) 'than one-half the length of the smallest'
WRITE (*, *) 'web, i.e. no slivering may occur except in'
WRITE (*, *) 'the inner layer of the grain.'
WRITE (*, *) ' '
DO 1010 J = 2, NUMPT
  WRITE (*, 6010) J
6010 FORMAT(5X,'Enter beginning depth for layer ',I2,')
  READ (*, *) P (1, J, 1)
1010 CONTINUE
1000 CONTINUE
C*****
C***** Thermochemical information entered *****
C*****
CALL CLEAR
WRITE (*, *) '    Thermochemical Properties'
WRITE (*, *)
WRITE (*, *) ' 1. Constant properties in each layer'
WRITE (*, *) '    (Use for standard and layered grains)'
WRITE (*, *)
WRITE (*, *) ' 2. Varying properties in at least one layer'
WRITE (*, *)
WRITE (*, *) ' Enter your choice'
READ (*, *) IGHR
IF ((IGHR .EQ. 1) .AND. (NUMPT .EQ. 1)) THEN
  A3 (31) = 1.
ELSE
  A3 (31) = 2.
END IF
IF (IGHR .EQ. 2) THEN
  A3 (31) = 0.0
END IF
CALL CLEAR
IF (IGHR .EQ. 1) THEN
  WRITE (*, *) 'Values will be entered for each layer. '
  WRITE (*, *) 'All input values are to be in metric units.'
  WRITE (*, *) ' '
  WRITE (*, *) ' '
  PAUSE
ELSE
  WRITE (*, 6000)
  WRITE (*, *) 'Values will be entered for the beginning and'

```

```

WRITE (*, *) 'end of each layer. All input in metric units.'
WRITE (*, *) 'For any layer there should be no more than a'
WRITE (*, *) '20% variation in any property.'
WRITE (*, *) ''
WRITE (*, *) ''
PAUSE
END IF
DO 1020 KK = 1, NUMPT
  CALL CLEAR
  IF (IGHR .EQ. 1) THEN
    WRITE (*, 6020) KK
6020  FORMAT(' ', 'Flame Temperature (K) for layer ', I2, ' :')
      READ (*, *) P (3, KK, 1)
      P (3, KK, 2) = P (3, KK, 1)
      WRITE (*, 6030) KK
6030  FORMAT(/, ' ', 'Impetus (J/g) for layer ', I2, ' :')
      READ (*, *) P (2, KK, 1)
      P (2, KK, 2) = P (2, KK, 1)
      P (5, KK, 1) = 8.314*P (3, KK, 1)/P (2, KK, 1)
      WRITE (*, 6040) KK, P (5, KK, 1)
6040  FORMAT(/, ' ', 'A consistent Molecular Weight for layer ', I2,
1/ ' given the Flame Temperature and Impetus is: ', F8.4,
2/ ' (Accept = 1, Enter new = 2) Enter choice', /)
      READ (*, *) IVHE
      IF (IVHE .EQ. 1) THEN
        P (5, KK, 2) = P (5, KK, 1)
      ELSE
        WRITE (*, *) 'Enter the Average Molecular Weight:'
        READ (*, *) P (5, KK, 1)
        P (5, KK, 2) = P (5, KK, 1)
      END IF
      WRITE (*, 6050) KK
6050  FORMAT(' ', 'Covolume (cc/g) for layer ', I2, ' :')
      READ (*, *) P (6, KK, 1)
      P (6, KK, 2) = P (6, KK, 1)
      WRITE (*, 6060) KK
6060  FORMAT(' ', 'Gamma for layer ', I2, ' :')
      READ (*, *) P (7, KK, 1)
      P (7, KK, 2) = P (7, KK, 1)
      WRITE (*, 6070) KK
6070  FORMAT(' ', 'Density (g/cc) for layer ', I2, ' :')
      READ (*, *) P (4, KK, 1)
      P (4, KK, 2) = P (4, KK, 1)
      ELSE
        WRITE (*, 6080) KK
6080  FORMAT(' ', 'Flame Temperature for layer ', I2, ' beginning (K):')
      READ (*, *) P (3, KK, 1)
      WRITE (*, 6090) KK

```

```

6090 FORMAT(' ','Flame Temperature for layer ',I2,' at end (K):')
      READ (*, *) P (3, KK, 2)
      WRITE (*, *) ' '
      WRITE (*, 6100) KK
6100 FORMAT(' ','Impetus for layer ',I2,' at beginning (J/g):')
      READ (*, *) P (2, KK, 1)
      WRITE (*, 6110) KK
6110 FORMAT(' ','Impetus for layer ',I2,' at end (J/g):')
      READ (*, *) P (2, KK, 2)
      WRITE (*, *) ' '
      WRITE (*, 6120) KK
6120 FORMAT(' ','Molecular Weight for layer ',I2,' at beginning :')
      READ (*, *) P (5, KK, 1)
      WRITE (*, 6130) KK
6130 FORMAT(' ','Molecular Weight for layer ',I2,' at end:')
      READ (*, *) P (5, KK, 2)
      WRITE (*, *) ' '
      WRITE (*, 6140) KK
6140 FORMAT(' ','Covolume for layer ',I2,' at beginning (cc/g):')
      READ (*, *) P (6, KK, 1)
      WRITE (*, 6150) KK
6150 FORMAT(' ','Covolume for layer ',I2,' at end (cc/g):')
      READ (*, *) P (6, KK, 2)
      WRITE (*, *) ' '
      WRITE (*, 6160) KK
6160 FORMAT(' ','Gamma for layer ',I2,' at beginning:')
      READ (*, *) P (7, KK, 1)
      WRITE (*, 6170) KK
6170 FORMAT(' ','Gamma for layer ',I2,' at end:')
      READ (*, *) P (7, KK, 2)
      WRITE (*, *) ' '
      WRITE (*, 6180) KK
6180 FORMAT(' ','Density for layer ',I2,' at beginning (g/cc):')
      READ (*, *) P (4, KK, 1)
      WRITE (*, 6190) KK
6190 FORMAT(' ','Density for layer ',I2,' at end (g/cc):')
      READ (*, *) P (4, KK, 2)
      END IF
      P (9, KK, 1) = 1.98717/P (5, KK, 1)
      P (9, KK, 2) = 1.98717/P (5, KK, 2)
      P (10, KK, 1) = P (9, KK, 1)/(P (7, KK, 1) - 1.)
      P (10, KK, 2) = P (9, KK, 2)/(P (7, KK, 2) - 1.)
      P (11, KK, 1) = P (10, KK, 1)*P (3, KK, 1)
      P (11, KK, 2) = P (10, KK, 2)*P (3, KK, 2)
1020 CONTINUE
      RETURN
      END
C*****

```

```

C***** SUBROUTINE MAXDEPTH *****
C*****
SUBROUTINE MAXDEPTH
COMMON A3 (100), P (11, 15, 5)
ITYPE = A3 (45)
C*****
C***** GRAINS FOR WHICH THE DEPTH CAN BE SOLVED *****
C*****
IF (ITYPE .EQ. 1) THEN
  A3 (1) = A3 (8)/2.
  RETURN
END IF
IF (ITYPE .EQ. 2) THEN
  X = A3 (8)/2.
  Y = A3 (7)/2.
  A3 (1) = MIN (X, Y)
  RETURN
END IF
IF (ITYPE .EQ. 3) THEN
  X = A3 (8)/2.
  Y = A3 (7)/2.
  Z = A3 (10)/2.
  A3 (1) = MIN (X, Y, Z)
  RETURN
END IF
IF (ITYPE .EQ. 4) THEN
  X = (A3 (8) - A3 (9))/2.
  Y = A3 (7)/2.
  A3 (1) = MIN (X, Y)
  RETURN
END IF
IF (ITYPE .EQ. 11) THEN
  A3 (1) = A3 (8)/2.
  RETURN
END IF
IF (ITYPE .EQ. 12) THEN
  A3 (1) = A3 (8)/2.
  RETURN
END IF
IF (ITYPE .EQ. 13) THEN
  A3 (1) = A3 (7)
  RETURN
END IF
C*****
C***** REMAINING GRAINS ARE HANDLED *****
C*****
R = MIN (A3 (10)/2., A3 (11)/2., A3 (12)/2., A3 (7)/2.)
B = MAX (A3 (7)/2., A3 (8)/2.)

```

```

      B = B/10000.
      DO 1000 I = 1, 10000
        R = R + B
        CALL FORMT (ITYPE, SFAREA, VOLUNB, R)
        IF (ABS (VOLUNB) .LE. 0.000001) THEN
          A3 (1) = R
          GO TO 1010
        END IF
      1000 CONTINUE
      WRITE (*, *) 'THE MAXIMUM DEPTH BURNED CANNOT BE DETERMINED.'
      WRITE (*, *) 'CHECK GRAIN GEOMETRY AND START OVER.'
      PAUSE
      1010 CONTINUE
      RETURN
      END
C*****
C*
C***** SUBROUTINE FORMT *****
C*
C*****
C*
C      ICODE: code for type of grain
C      R : burn depth
C      GL: unburned grain length
C      D : unburned outer diameter
C      PD: unburned perforation diameter
C      WI, WM, WO: inner, middle and outer webs respectively
C
C      Output:
C      SFAREA: surface area
C      FRCSFA: surface area/initial surface area
C      VOLUNB: unburned volume
C      VOLBRN: burned volume
C      FRCBRN: burned volume/initial volume
C      VOLMAO: unburned volume of outer layer
C      VOLMBO: unburned volume of inner layer
C      VOLABR: burned volume of outer layer
C      VOLBBR: burned volume of inner layer
C
C*****
      SUBROUTINE FORMT (ICODE, SFAREA, VOLUNB, R)
      COMMON A3 (100), P (11, 15, 5)
      DIMENSION S7 (4), S19 (3, 4)
      DATA RT/1.732050808/, PI3/1.047197551/, PI/3.141592654/
C*****
C***** SET GRAIN GEOMETRY *****
C*****
      GL = A3 (7)

```

```

D = A3 (8)
PD = A3 (9)
WI = A3 (10)
WM = A3 (11)
WO = A3 (12)
C*****
C***** Set U = 2*(depth burned) and branch to grain type *****
C*****
      U = 2.0*R
C*****
C***** CIGARETTE GRAIN *****
C*****
      IF (ICODE .EQ. 13) THEN
        SFAREA = PI*D*D/4
        VOLUNB = (GL - R)*SFAREA
        RETURN
      END IF
C*****
C***** ALL OTHER GRAINS ARE HANDLED *****
C*****
      GO TO ( 1000, 1010, 1020, 1030, 1040, 1050, 1060, 1070,
1      1080, 1090, 1100, 1110), ICODE
C*****
C***** CODE 1: 7-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns.
C*****
1050 CONTINUE
      D = 3.0*PD + 2.0*(WI + WO)
      E0 = PI*(D**2 - 7.0*PD**2)/4.0
      S0 = PI*(D + 7.0*PD)*GL + 2.0*E0
      V0 = E0*GL
      WW = WI + PD
      DO 1120 K = 1, 3
        S7 (K) = WW
1120 CONTINUE
      WEBC = AMIN1 (WO, WI, GL)
C*****
C*** This part does the calculations for the burning grain.
C*****
      GRL = AMAX1 (GL - U, 0.0)
      OD = D - U
      PRFD = PD + U
      IF (U .GT. WEBC) GO TO 1130
      E = PI*(OD**2 - 7.0*PRFD**2)/4.0
      SFAREA = PI*(OD + 7.0*PRFD)*GRL + 2.0*E
      FRCSFA = SFAREA/S0
      VOLUNB = E*GRL

```

```

VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C*** This part does the calculations for when the grain slivers. *****
C*****
1130 CONTINUE
CALL GENIS (S7, PRFD, GRL, SF1, GV1)
CALL GENOS (S7, PRFD, GRL, 0.5*OD, SF2, GV2)
SFAREA = 6.0*(SF1 + SF2)
FRCSFA = SFAREA/S0
VOLUNB = 6.0*(GV1 + GV2)
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 2: 1-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns.
C*****
1030 CONTINUE
D = PD + 2.0*WI
E0 = PI*(D**2 - PD**2)/4.0
S0 = PI*(D + PD)*GL + 2.0*E0
V0 = E0*GL
WEBC = AMIN1 (GL, WI)
C*****
C***** This part does the calculations for the burning grain. *****
C*****
IF (U .GE. WEBC) THEN
  GRL = 0.0
  E = 0.0
  GO TO 1140
END IF
GRL = GL - U
OD = D - U
PRFD = PD + U
E = PI*(OD**2 - PRFD**2)/4.0
1140 CONTINUE
SFAREA = PI*(OD + PRFD)*GRL + 2.0*E
FRCSFA = SFAREA/S0
VOLUNB = E*GRL
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 3: CORD GRAIN *****
C*****

```

```

C*** This part calculates the conditions before the grain burns. *****
C*****
1010 CONTINUE
  S0 = GL*PI*D + PI*D**2/2.0
  V0 = GL*PI*D**2/4.0
C*** This part does the calculations for the burning grain.
  GRL = AMAX1 (GL - U, 0.0)
  OD = AMAX1 (D - U, 0.0)
  E = PI*OD**2/4.0
  SFAREA = PI*OD*GRL + 2.0*E
  FRCSFA = SFAREA/S0
  VOLUNB = GRL*PI*OD**2/4.0
  VOLBRN = V0 - VOLUNB
  FRCBRN = VOLBRN/V0
  RETURN
C*****
C***** CODE 4: RECTANGULAR STRIP GRAIN *****
C*****
1020 CONTINUE
  S0 = 2.0*(GL*D + D*WI + GL*WI)
  V0 = GL*D*WI
  GRL = AMAX1 (GL - U, 0.0)
  DS = AMAX1 (D - U, 0.0)
  WIS = AMAX1 (WI - U, 0.0)
  SFAREA = 2.0*(GRL*DS + DS*WIS + WIS*GRL)
  FRCSFA = SFAREA/S0
  VOLUNB = GRL*DS*WIS
  VOLBRN = V0 - VOLUNB
  FRCBRN = VOLBRN/V0
  RETURN
C*****
C***** CODE 5: SPHERICAL GRAIN *****
C*****
1000 CONTINUE
  S0 = PI*D**2
  V0 = PI*D**3/6.0
  OD = AMAX1 (D - U, 0.0)
  SFAREA = PI*OD**2
  FRCSFA = SFAREA/S0
  VOLUNB = PI*OD**3/6.0
  VOLBRN = V0 - VOLUNB
  FRCBRN = VOLBRN/V0
  RETURN
C*****
C***** CODE 6: SLOTTED-TUBE GRAIN *****
C*****
C*** This part does the calculations before the grain burns. *****
C*****

```

```

1040 CONTINUE
  SLOT = 0.5*WM
  SO = 0.5*D
  SI = 0.5*PD
  THETA = ASIN (SLOT/SO)
  ALPHA = ASIN (SLOT/SI)
  E0 = (PI - ALPHA)*(SO**2 - SI**2) + (SO - SI)**2*ALPHA
  S0 = 2.0*((PI - ALPHA)*SI + (PI - THETA)*SO + (SO*COS (THETA)
1 - SI*COS (ALPHA)))*GL + 2.0*E0
  V0 = GL*E0
  WI = SO - SI
  WEBC = AMIN1 (GL, WI)

```

C*****

C*** This part does the calculations for the burning grain. *****

C*****

```

  IF (U .GE. WEBC) THEN

```

```

    GRL = 0.0

```

```

    E = 0.0

```

```

    GO TO 1150

```

```

  END IF

```

```

  SLOT = 0.5*(WM + U)

```

```

  SO = 0.5*(D - U)

```

```

  SI = 0.5*(PD + U)

```

```

  GRL = GL - U

```

```

  THETA = ASIN (SLOT/SO)

```

```

  ALPHA = ASIN (SLOT/SI)

```

```

  E = (PI - ALPHA)*(SO**2 - SI**2) + (SO - SI)**2*ALPHA

```

```

1150 CONTINUE

```

```

  SFAREA = 2.0*((PI - ALPHA)*SI + (PI - THETA)*SO + (SO*COS (THETA)
1 - SI*COS (ALPHA)))*GRL + 2.0*E

```

```

  FRCSFA = SFAREA/S0

```

```

  VOLUNB = E*GRL

```

```

  VOLBRN = V0 - VOLUNB

```

```

  FRCBRN = VOLBRN/V0

```

```

  RETURN

```

C*****

C***** CODE 7: ROUND-HEX 37-PERF GRAIN *****

C*****

```

1090 CONTINUE

```

```

  SO = 18.

```

```

  SI = 54.

```

```

  NPERF = 37

```

```

  D = 7.0*PD + 6.0*WI + 2.0*WO

```

```

  GO TO 1160

```

C*****

C***** CODE 8: ROUND-HEX 19-PERF GRAIN *****

C*****

```

1080 CONTINUE

```

```

SO = 12.0
SI = 24.0
NPERF = 19
D = 5.0*PD + 4.0*WI + 2.0*WO
C*****
C***** CALCULATIONS FOR CODES 7,8,10 *****
C*****
1160 CONTINUE
  WW = WI + PD
  WW2 = WW**2
  PRFD = PD + U
  PRFD2 = PRFD**2
  GRL = AMAX1 (GL - U, 0.0)
  E = 0.0
  THETA = 2.0*ACOS (AMIN1 (WW/PRFD, 1.0))
  ALPHA = ACOS (AMIN1 ((2.0*WO + PD - U)/PRFD, 1.0))
  IF (U .LT. WO) E = 0.25*PI*((2.0*WO + PD - U)**2 - PRFD2)
  IF (THETA .GE. PI3) GO TO 1170
  E = E + SI*0.25*(WW2*RT - 1.5*PRFD2*(SIN (THETA) + PI3 - THETA))
1170 CONTINUE
  IF (ALPHA .GE. 0.5*(PI - THETA)) GO TO 1180
  E = E + SO*0.125*(2.0*(2.0*WO + PD - U)*(2.0*
1  WW - PRFD*SIN (ALPHA))
2  - PRFD2*(SIN (THETA) + PI - 2.0*ALPHA - THETA))
1180 CONTINUE
  IF (2.0*WO + PD .LT. WI) THEN
    WRITE (*, *) '*FORMT* BAD HEX PROP'
    PAUSE
    GO TO 1190
  END IF
  VOLUNB = E*GRL
C***** TEST TO SEE IF GRAIN CONSUMED *****
  IF (VOLUNB .LE. 0.0) THEN
    SFAREA = 0.0
    VOLUNB = 0.0
    GO TO 1190
  END IF
C*****NOW THE SURFACE AREA*****
  PH = D/2. - WO - PD/2.
  IF (U .EQ. 0.0) THEN
    SFAREA = 2.*E + GRL*PH*6. + NPERF*PI*PD*
1  GRL + PI*GRL*(2*WO + PD)
    GO TO 1190
  END IF
C*****NO SLIVERING YET*****
  IF ((WO .GT. U) .AND. (WI .GT. U)) THEN
    SFAREA = 2.*E + NPERF*(PD + U)*GRL*PI + 6.*PH*
1  GRL + PI*(2.*WO + PD - U)*GRL

```

```

        GO TO 1190
    END IF
C***** NOW SLIVERING *****
    SFAREA = 2.*E
C***** FIRST THE INNER SLIVERS *****
    IF (THETA .GE. PI3) THEN
        GO TO 1200
    ELSE
        SFAREA = SFAREA + 1.5*PRFD*GRL*(PI3 - THETA)*SI
    END IF
    1200 CONTINUE
C***** NOW THE OUTER SLIVERS & CORNERS *****
C***** CORNERS NOT CONSUMED *****
    IF (WO .GT. U) THEN
        SFAREA = SFAREA + PI*(PD + 2.*WO - U + PRFD)*GRL
    END IF
C*****NOW OUTER SLIVERS*****
    IF (ALPHA .LT. .5*(PI - THETA)) THEN
        SFAREA = SFAREA + (WW - PRFD*SIN (ALPHA))*GRL*SO +
        1    PRFD*GRL*(PI/2. - ALPHA - THETA/2.)*SO
    END IF
    1190 CONTINUE
    RETURN
C*****
C***** CODE 9: 19-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns *****
C*****
    1070 CONTINUE
        D = 5.0*PD + 2.0*(WI + WM + WO)
        E0 = PI*(D**2 - 19.0*PD**2)/4.0
        S0 = PI*(D + 19.0*PD)*GL + 2.0*E0
        V0 = E0*GL
        S19 (1, 1) = WI + PD
        S19 (2, 1) = S19 (1, 1)
        S19 (3, 1) = S19 (1, 1)
        S19 (1, 2) = 0.5*SQRT (3.0*(WM + PD)**2 + (WI + PD)**2)
        S19 (2, 2) = S19 (1, 2)
        S19 (3, 2) = S19 (1, 1)
        S19 (1, 3) = PD + 0.5*(WI + WM)
        S19 (2, 3) = S19 (1, 2)
        S19 (3, 3) = WM + PD
        S19 (1, 4) = S19 (1, 3)
        S19 (2, 4) = 2.0*S19 (1, 3)
        S19 (3, 4) = S19 (1, 3)*RT
        WEBC = AMIN1 (WO, WM, WI, S19 (1, 3) - PD, S19 (1, 2) - PD, GL)
C*****
C*** This part does the calculations for the burning grain. *****

```

```

C*****
  GRL = AMAX1 (GL - U, 0.0)
C*****
  OD = D - U
  PRFD = PD + U
  IF (U .GE. WEBC) GO TO 1210
  E = 0.25*PI*(OD**2 - 19.0*PRFD**2)
  SFAREA = PI*(OD + 19.0*PRFD)*GRL + 2.0*E
  FRCSFA = SFAREA/S0
  VOLUNB = E*GRL
  VOLBRN = V0 - VOLUNB
  FRCBRN = VOLBRN/V0
  RETURN
1210 CONTINUE
  SUMSA = 0.0
  SUMGV = 0.0
  DO 1220 K = 1, 2
    CALL GENIS (S19 (1, K), PRFD, GRL, SA, GV)
    SUMSA = SUMSA + 6.0*SA
    SUMGV = SUMGV + 6.0*GV
1220 CONTINUE
  CALL GENIS (S19 (1, 3), PRFD, GRL, SA, GV)
  SUMSA = SUMSA + 12.0*SA
  SUMGV = SUMGV + 12.0*GV
  CALL GENOS (S19 (1, 4), PRFD, GRL, .5*OD, SA, GV)
  SUMSA = SUMSA + 12.0*SA
  SUMGV = SUMGV + 12.0*GV
  SFAREA = SUMSA
  FRCSFA = SFAREA/S0
  VOLUNB = SUMGV
  VOLBRN = V0 - VOLUNB
  FRCBRN = VOLBRN/V0
  RETURN
C*****
C***** CODE 10: ROUND-HEX 7-PERF GRAIN *****
C*****
1060 CONTINUE
  SO = 6.0
  SI = 6.0
  NPERF = 7
  GO TO 1160
C*****
C***** CODE 11: CORD WITH INHIBITED ENDS *****
C*****
C This routine will only calculate the surface area of the lateral
C surfaces.
C It will not calculate the surface area of the inhibited ends. *****
C*****

```

```

1100 CONTINUE
    SO = GL*PI*D
    VO = GL*PI*D**2/4.0
C **** This part does the calculations for the burning grain
    OD = AMAX1 (D - U, 0.0)
    SFAREA = PI*OD*GL
    FRCSFA = SFAREA/SO
    VOLUNB = GL*PI*OD**2/4.0
    VOLBRN = VO - VOLUNB
    FRCBRN = VOLBRN/VO
    RETURN
C*****
C***** CODE 12: RECTANGULAR STRIP GRAIN WITH INHIBITED SIDES *****
C*****
C This routine will only calculate the surface area of the two burning
C sides.
C It will not calculate the surface area of the inhibited sides.
C*****
1110 CONTINUE
    SO = 2.0*GL*WI
    VO = GL*D*WI
    VOLMAO = (D - WI)*GL*WO
    VOLMBO = WI*GL*WO
C*****
C *** This part does the calculations for the burning grain
    DS = AMAX1 (D - U, 0.0)
    SFAREA = 2.0*GL*WI
    FRCSFA = SFAREA/SO
    VOLUNB = GL*DS*WI
    VOLBRN = VO - VOLUNB
    FRCBRN = VOLBRN/VO
    IF (DS .GE. WI) THEN
        VOLABR = VOLBRN
        VOLBBR = 0.
    ELSE
        VOLABR = VOLMAO
        VOLBBR = VOLBRN - VOLABR
    END IF
    RETURN
    END
C*****
C*****
C
C SUBROUTINE *GENIS*: calculate surface area and volume for a
C     general inner sliver of a burning grain
C     with length = GRL & perforation dia. = PRFD.
C
    SUBROUTINE GENIS (S, PRFD, GRL, SURF, VOL)

```

```

      DIMENSION S (3), A (4)
      DATA PI2/1.5707963/
C
C
C ***** : Store angles A1,A2,A3 and area of triangle
C           with sides S(1),S(2),S(3) into A(1)...A(4)
C
      A (1) = ACOS ((S (2)**2 + S (3)**2 - S (1)**2)/(2.0*S (2)*S (3)))
      A (2) = ACOS ((S (1)**2 + S (3)**2 - S (2)**2)/(2.0*S (1)*S (3)))
      A (3) = ACOS ((S (1)**2 + S (2)**2 - S (3)**2)/(2.0*S (1)*S (2)))
      A (4) = 0.5*S (1)*S (3)*SIN (A (2))
C
C ...check for error condition: find if triangle acceptable...
C
      J = 0
      DO 1000 I = 1, 3
        IF (A (I) .LT. 0.5*PI2) J = J + 1
      1000 CONTINUE
      IF (J .GT. 1) STOP ' GENIS ERROR'
C
C
C succeeding passes until burnout: find auxiliary angles
C
      TAU12 = ACOS (AMIN1 (1.0, S (3)/PRFD))
      TAU13 = ACOS (AMIN1 (1.0, S (2)/PRFD))
      TAU23 = ACOS (AMIN1 (1.0, S (1)/PRFD))
C
C ...and branch to 25 if sliver fails burnout criteria. If not
C then sliver is burned and go to 30.
C
      IF (TAU12 + TAU13 + TAU23 .LT. PI2 .AND. GRL .GT. 0.0) THEN
        GO TO 1010
      ELSE
        GO TO 1020
      END IF
C
C
C sliver not burned out: determine end area, volume and surface area
C
      1010 CONTINUE
      E = A (4) - 0.25*PRFD*(S (1)*SIN (TAU23) + S (2)*SIN (TAU13)
      1 + S (3)*SIN (TAU12) + PRFD*(PI2 - TAU12 - TAU13 - TAU23))
C
      VOL = E*GRL
C
      SURF = 2.0*E + GRL*PRFD*(PI2 - TAU12 - TAU13 - TAU23)
C
C ...and RETURN

```

```

C
C   RETURN
C
C   sliver is burned out: return with zero volume and surface area.
C
C   1020 CONTINUE
C       VOL = 0.0
C       SURF = 0.0
C       RETURN
C       END
C*****
C SUBROUTINE "GENOS" : Calculates surface area and volume for a
C                       general outer sliver of a burning grain
C                       with length = GRL, radius = RAD, and
C                       perforation diameter = PRFD
C
C   SUBROUTINE GENOS (S, PRFD, GRL, RAD, SURF, VOL)
C   DIMENSION S (3), A (4)
C
C   ***** Store angles A1,A2,A3 and area of triangle
C   with sides S(1),S(2),S(3) into A(1) ...A(4)
C
C   A (1) = ACOS ((S (2)**2 + S (3)**2 - S (1)**2)/(2.0*S (2)*S (3)))
C   A (2) = ACOS ((S (1)**2 + S (3)**2 - S (2)**2)/(2.0*S (1)*S (3)))
C   A (3) = ACOS ((S (1)**2 + S (2)**2 - S (3)**2)/(2.0*S (1)*S (2)))
C   A (4) = 0.5*S (1)*S (3)*SIN (A (2))
C
C
C   succeeding passes until burnout: determine auxiliary angles
C
C   TAU1 = ACOS (AMIN1 (1., (S (2)**2 + RAD**2 - 0.25*
1   PRFD**2)/(2.*S (2)*RAD)))
C   TAU2 = ACOS (AMIN1 (1., (S (3)**2 + RAD**2 - 0.25*
1   PRFD**2)/(2.*S (3)*RAD)))
C   TAU3 = ACOS (AMAX1 (- 1.0, (S (2)**2 - RAD**2 + 0.25*
1   PRFD**2)/(S (2)*PRFD)))
C   TAU4 = ACOS (AMAX1 (- 1.0, (S (3)**2 - RAD**2 + 0.25*
1   PRFD**2)/(S (3)*PRFD)))
C
C   SIG = ACOS (AMIN1 (1.0, S (1)/PRFD))
C
C   ...then check error conditions...
C
C   IF (TAU3 .LT. A (3) .OR. TAU4 .LT. A (2)) STOP ' *GENOS* ERROR'
C
C   ...IF ok, check if sliver burned out. If not burned out go to 25.
C   If burned cut go to 30
C

```

```

      IF (TAU1 + TAU2 .LT. A (1) .AND. GRL .GT. 0.0) THEN
        GO TO 1000
      ELSE
        GO TO 1010
      END IF
C
C
C sliver not burned out: determine end area, volume and surface area.
C
1000 CONTINUE
      E = 0.5*RAD*(S (2)*SIN (TAU1) + RAD*(A (1) - TAU1 - TAU2)
1      + S (3)*SIN (TAU2)) - A (4) - 0.25*PRFD*(S (1)*SIN (SIG)
2      + 0.5*PRFD*(TAU3 + TAU4 - 2.0*SIG - A (2) - A (3)))
C
      VOL = E*GRL
C
      SURF = 2.0*E + GRL*(RAD*(A (1) - TAU1 - TAU2) + 0.5*PRFD*(TAU3
1      + TAU4 - 2.0*SIG - A (2) - A (3)))
C
C ...and RETURN.
C
      RETURN
C
C sliver is burned out: return with zero volume and surface area.
C
1010 CONTINUE
      VOL = 0.0
      SURF = 0.0
      RETURN
      END
C*****
C*****SUBROUTINE COMMASS*****
C*****
      SUBROUTINE COMMASS
C*****
C Version 3.0, January 1992
C
C THIS SUBROUTINE WILL COMPUTE THE INITIAL MASS PER LAYER AS WELL AS
C THE INTEGRALS OF EACH PROPERTY
C
C*****
      COMMON A3 (100), P (11, 15, 5)
      IGTYP = INT (A3 (45) + .5)
      NL = INT (A3 (4) + .5)
      IGQR = INT (A3 (31) + .5)
C*****
C***** STORAGE LOCATION FOR INTEGRAL ARE ZEROED *****
C*****

```

```

DO 1000 I = 2, 11
  DO 1000 J = 1, 15
    P (I, J, 3) = 0.0
1000 CONTINUE
  IF (IGQR .EQ. 1) THEN
    XI = 0.0
    CALL FORMT (IGTYPE, ASURF, VOLUNB, XI)
    P (8, 1, 3) = VOLUNB
    DO 1010 I = 2, 11
      P (I, 1, 3) = P (I, 1, 1)*P (8, 1, 3)
1010 CONTINUE
    A3 (3) = P (4, 1, 3)
    RETURN
  END IF
  DO 1020 J = 1, NL
C*****
C***** DETERMINE X VALUES FOR INTEGRATION *****
C*****
    IF (J .EQ. NL) THEN
      XS = P (1, J, 1)
      XL = A3 (1)
    ELSE
      XS = P (1, J, 1)
      XL = P (1, J + 1, 1)
    END IF
C*****
C***** ALWAYS 300 SUBDIVISIONS *****
C*****
    XDEL = XL - XS
    XSTEP = XDEL/300.
    DO 1030 I = 1, 301
      XI = XS + (I - 1)*XSTEP
C*****
      CALL FORMT (IGTYPE, ASURF, VOLUNB, XI)
C*****
    DO 1040 K = 2, 11
      PDEL = P (K, J, 2) - P (K, J, 1)
C*****
C***** INTEGRATION IS PERFORMED *****
C*****
      FCN = ((PDEL/XDEL)*(XI - XS) + P (K, J, 1))*ASURF
      IF ((I .EQ. 1) .OR. (I .EQ. 301)) THEN
        P (K, J, 3) = P (K, J, 3) + FCN
      ELSE
        P (K, J, 3) = P (K, J, 3) + 2.*FCN
      END IF
1040 CONTINUE
1030 CONTINUE

```

```

      DO 1050 K = 2, 11
        P (K, J, 3) = P (K, J, 3)*(XDEL/600.)
1050  CONTINUE
1020 CONTINUE
C*****
C***** MASS OF SINGLE GRAIN*****
C*****
      DO 1060 I = 1, NL
        A3 (3) = A3 (3) + P (4, I, 3)
1060 CONTINUE
      RETURN
      END
C*****
C***** SUBROUTINE PRINTP*****
C*****
      SUBROUTINE PRINTP (A1)
      CHARACTER*20 A1 (20)
      COMMON A3 (100), P (11, 15, 5)
C*****
      NL = INT (A3 (4) + .5)
      OPEN (UNIT = 7, FILE = 'LPT1')
      WRITE (7, 6000) A1 (3)
6000 FORMAT(10X,'MASTER FILE: ',A20//)
      WRITE (7, 6010)
6010 FORMAT(/,30X,'BEG LAY      END LAY      INTEGRAL')
      DO 1000 I = 1, 11
        WRITE (7, 6020) I
6020 FORMAT(/,30X,'PROPERTY NUMBER: ',I3)
        DO 1010 J = 1, NL
          WRITE (7, 6030) J, P (I, J, 1), P (I, J, 2), P (I, J, 3)
6030 FORMAT(' ',I3,15X,3F20.10)
1010  CONTINUE
1000 CONTINUE
      RETURN
      END

```

INTENTIONALLY LEFT BLANK.

APPENDIX G:
LISTING - PROGRAM MKGAGE.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKGAGE

C*****

C Version: 3.0, January 1992

C

C Written by: William Oberle, U.S. Army Research Laboratory

C

C Purpose: This program maintains a data base for gages used

C in the BRLCB data reduction program. The data

C file is GAGEFILE and should be on the same drive

C as the program.

C*****

CHARACTER AG (100)*6, BG (100)*9, XGAGE*6, YGAGE*6, XTEMP*1

DIMENSION CG (100, 3)

C*****

C***** CLEARING ARRAYS *****

C*****

DO 1000 I = 1, 100

AG (I) = ' '

BG (I) = ' '

CG (I, 1) = 0.0

CG (I, 2) = 0.0

CG (I, 3) = 0.0

1000 CONTINUE

C*****

C The gage data file is read.

C*****

CALL READER (ITN, AG, BG, CG)

C*****

C The number of pages is determined.

C*****

1010 CONTINUE

IF ((ITN .GE. 1) .AND. (ITN .LE. 20)) NPAGE = 1

IF ((ITN .GE. 21) .AND. (ITN .LE. 40)) NPAGE = 2

IF ((ITN .GE. 41) .AND. (ITN .LE. 60)) NPAGE = 3

IF ((ITN .GE. 61) .AND. (ITN .LE. 80)) NPAGE = 4

IF ((ITN .GE. 81) .AND. (ITN .LE. 100)) NPAGE = 5

C*****

C The gage sub-menu is presented.

C*****

1020 CONTINUE

CALL CLEAR

WRITE (*, 6000)

6000 FORMAT(' ', T30, 'Gage Maintenance Program')

WRITE (*, 6010)

```

6010 FORMAT(' ',T38,'Sub Menu')
      WRITE (*, *)
      WRITE (*, *)
      WRITE (*, *) '          1. View gage information.'
      WRITE (*, *) '          2. Add a gage to data base.'
      WRITE (*, *) '          3. Delete a gage from data base.'
      WRITE (*, *) '          4. Locate specific gage in data base.'
      WRITE (*, *) '          5. Exit.'
      WRITE (*, *) ' '
      WRITE (*, *) '    Please Enter Your Choice (1-5): '
      READ (*, *) ICHOICE
      IF (ICHOICE .GT. 5) THEN
        WRITE (*, *)
        WRITE (*, *) 'The choice you have made is not between'
        WRITE (*, *) '1 and 5. Please select again.'
        GO TO 1020
      END IF
C*****
C The proper action is taken depending on the selection.
C*****
      IF (ICHOICE .EQ. 5) GO TO 1030
      IF (ICHOICE .EQ. 1) THEN
C*****
        IF (ITN .EQ. 0) THEN
          CALL CLEAR
          WRITE (*, *) 'No gages are in the gage file.'
          WRITE (*, *) 'Enter another option.'
          WRITE (*, *)
          PAUSE
          GO TO 1020
        END IF
        CALL CLEAR
        WRITE (*, *)
1040    CONTINUE
        WRITE (*, 6020) ITN
6020    FORMAT(' ', 'There are ',I3,' entries in the gage data file.')
        WRITE (*, 6030) NPAGE
6030    FORMAT(' ', 'This information is displayed in ',I3,' screen pages.')
        WRITE (*, 6040)
6040    FORMAT(' ', 'Which page would you like to view? ')
        READ (*, *) IPAGE
        IF (IPAGE .GT. NPAGE) THEN
          WRITE (*, *)
          WRITE (*, *) 'The choice you have made is not in'
          WRITE (*, *) 'the proper range. Please select again.'
          GO TO 1040
        END IF
        CALL HEADER

```

```

CALL PAGER (ITN, IPAGE, AG, BG, CG)
GO TO 1010
C*****
C The end of option 1.
C*****
END IF
C*****
C Option 2
C*****
IF (ICHOICE .EQ. 2) THEN
CALL GAGEADD (ITN, AG, BG, CG)
GO TO 1010
END IF
C*****
C Option 3
C*****
IF (ICHOICE .EQ. 3) THEN
CALL DROPPER (ITN, AG, BG, CG)
GO TO 1010
END IF
C*****
C Option 4
C*****
IF (ICHOICE .EQ. 4) THEN
C*****
C A screen clearing routine should be inserted at this point
C*****
CALL CLEAR
1050 CONTINUE
WRITE (*, *) 'Enter the I.D. number of the desired gage. '
WRITE (*, *) 'Enter all letters as capital letters.'
READ (*, 5000) XGAGE
5000 FORMAT(A6)
1060 CONTINUE
YGAGE = ' '
IF (XGAGE .EQ. ' ') GO TO 1050
C*****
C Test is determine if xgage starts with a blank
C*****
IF (XGAGE (1:1) .EQ. ' ') THEN
DO 1070 II = 2, 6
YGAGE (II - 1:II - 1) = XGAGE (II)
1070 CONTINUE
XGAGE = YGAGE
GO TO 1060
END IF
C*****
C Now the search begins.

```

```

C*****
DO 1080 I = 1, ITN
  IF (XGAGE .EQ. AG (I)) THEN
    WRITE (*, *) 'A match has been found.'
    WRITE (*, *)
    WRITE (*, *)
    PAUSE
    CALL HEADER
    WRITE (*, 6050) I, AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
6050 FORMAT(' ',I3,2X,A6,2X,A9,3X,E12.5,6X,E12.5,6X,E12.5)
    WRITE (*, *)
    WRITE (*, *) 'Press Enter to Continue. '
    READ (*, 5010) XTEMP
5010 FORMAT(A1)
    GO TO 1010
  END IF
1080 CONTINUE
  WRITE (*, *) 'The gage is not in the table.'
  WRITE (*, *) 'Press Enter to Continue. '
  READ (*, 5010) XTEMP
  GO TO 1010
C*****
C This is the end of option 4.
C*****
  END IF
C*****
C To exit the program the gage file is written.
C*****
1030 CONTINUE
  OPEN (UNIT = 8, FILE = 'GAGEFILE')
  WRITE (8, *) ITN
  DO 1090 I = 1, ITN
    WRITE (8, 6060) AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1090 CONTINUE
6060 FORMAT(A6,A9,3E12.5)
  CLOSE (UNIT = 8)
  STOP
  END
C*****
  SUBROUTINE HEADER
C*****
  CALL CLEAR
  WRITE (*, *) ' The data fit is of the form: A + Bx + Cx^2'
  WRITE (*, 6000)
6000 FORMAT(' ', ' #',4X,'I.D.',3X,'Cal. Date',8X,'A',18X,'B',18X,'C')
  WRITE (*, 6010)
6010 FORMAT(' ',80X('-'))
  RETURN

```

```

      END
C*****
      SUBROUTINE PAGER (ITN, IPAGE, AG, BG, CG)
C*****
      CHARACTER AG (100)*6, BG (100)*9
      DIMENSION CG (100, 3)
C*****
C Test to see if there are entries on the chosen page.
C*****
1000 CONTINUE
      IX = 20*(IPAGE - 1)
      IF (IX .LT. ITN) GO TO 1010
      WRITE (*, *) 'There are no entries on the page selected.'
1020 CONTINUE
      WRITE (*, *) 'Please enter a new desired page number. '
      READ (*, *) IPAGE
      CALL HEADER
      GO TO 1000
1010 CONTINUE
      ISTART = 20*(IPAGE - 1) + 1
      IEND = MIN0 (20*IPAGE, ITN)
      DO 1030 I = ISTART, IEND
          WRITE (*, 6000) I, AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1030 CONTINUE
6000 FORMAT(' ',I3,2X,A6,2X,A9,3X,E12.5,6X,E12.5,6X,E12.5)
1040 CONTINUE
      WRITE (*, *) 'New page (1) or options (2)? '
      READ (*, *) IOPT
      IF (IOPT .GT. 2) THEN
          WRITE (*, *)
          WRITE (*, *) 'The choice you have made is not a'
          WRITE (*, *) '1 or 2. Please select again.'
          GO TO 1040
      END IF
      IF (IOPT .EQ. 1) GO TO 1020
      RETURN
      END
C*****
      SUBROUTINE READER (ITN, AG, BG, CG)
C*****
      CHARACTER AG (100)*6, BG (100)*9
      DIMENSION CG (100, 3)
      OPEN (UNIT = 7, FILE = 'GAGEFILE')
      REWIND 7
C*****
C The gage file is opened and read
C*****
      READ (7, *) ITN

```

```

IF (ITN .EQ. 0) THEN
  WRITE (*, *) 'The gage file is empty.'
  CLOSE (UNIT = 7)
  RETURN
END IF
DO 1000 I = 1, ITN
  READ (7, 5000) AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1000 CONTINUE
5000   FORMAT(A6,A9,3E12.5)
  CLOSE (UNIT = 7)
  RETURN
END
C*****
  SUBROUTINE GAGEADD (ITN, AG, BG, CG)
C*****
  CHARACTER AG (100)*6, BG (100)*9, XGAGE*6, YGAGE*6
  DIMENSION CG (100, 3)
  ITN = ITN + 1
  IF (ITN .GT. 100) THEN
    WRITE (*, *) 'No room left in the gage data file.'
    WRITE (*, *) 'Return to menu and delete unwanted gages.'
    RETURN
  END IF
  CALL CLEAR
1000 CONTINUE
  WRITE (*, *) ' Please enter the gage ID. '
  WRITE (*, *) ' Enter all letters as capital letters.'
  READ (*, 5000) XGAGE
5000 FORMAT(A6)
1010 CONTINUE
  YGAGE = ' '
  IF (XGAGE .EQ. ' ') GO TO 1000
C*****
C Test is determine if xgage starts with a blank
C*****
  IF (XGAGE (1:1) .EQ. ' ') THEN
    DO 1020 II = 2, 6
      YGAGE (II - 1:II - 1) = XGAGE (I:I)
1020 CONTINUE
      XGAGE = YGAGE
      GO TO 1010
    END IF
    AG (ITN) = XGAGE
    WRITE (*, *) 'Enter the date of gage calibration (mm/dd/yy). '
    READ (*, 5010) BG (ITN)
5010 FORMAT(A9)
    WRITE (*, *) 'Calibration constants must produce pressure in MPa'
    WRITE (*, *) 'The form of the data fit is: P(MPa)= A + Bx + Cx^2'

```

```

WRITE (*, *) 'Enter the value for A: '
READ (*, *) CG (ITN, 1)
WRITE (*, *) 'Enter the value for B: '
READ (*, *) CG (ITN, 2)
WRITE (*, *) 'Enter the value for C: '
READ (*, *) CG (ITN, 3)
C*****
C Now a search is performed to determine if the new entry supercedes
C a earlier data set.
C*****
DO 1030 I = 1, ITN - 1
  IF (XGAGE .EQ. AG (I)) THEN
    WRITE (*, *) 'The gage added to the data base was already in'
    WRITE(*,*)'the data base. The earlier entry will be tagged'
    WRITE (*, *) 'with a "*" as the last character of the I.D. field.'
    XGAGE (6:6) = '*'
    AG (I) = XGAGE
    WRITE (*, *)
    PAUSE
  END IF
1030 CONTINUE
RETURN
END
C*****
SUBROUTINE DROPPER (ITN, AG, BG, CG)
C*****
CHARACTER AG (100)*6, BG (100)*9
DIMENSION CG (100, 3)
CALL CLEAR
WRITE (*, *) 'Enter the number for the line in the gage data file'
WRITE (*, *) 'to be deleted. This is not the gage I.D. '
READ (*, *) IDEL
C*****
C Shifting the entries in the table.
C*****
DO 1000 I = IDEL + 1, ITN
  AG (I - 1) = AG (I)
  BG (I - 1) = BG (I)
  CG (I - 1, 1) = CG (I, 1)
  CG (I - 1, 2) = CG (I, 2)
  CG (I - 1, 3) = CG (I, 3)
1000 CONTINUE
ITN = ITN - 1
RETURN

```

```

      END
C*****
      SUBROUTINE CLEAR
C*****
      CHARACTER ST*4
      DATA ST/' [2J'/
      WRITE (*, 6000) ST
6000 FORMAT (1X,A4)
      RETURN
      END

```

APPENDIX H:
LISTING - PROGRAM MKPTDATA.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM PTDATA

```
C*****
C Version: 3.0, JANUARY 1991
C
C       Last Revision: 2/8/92; cleanup
C
C Written by: William Oberle, U.S. Army Research Laboratory
C
C Purpose: The purpose of this program is to prepare the pressure
C          time data for use in BRLCB. This assumes that the P/t
C          or V/t data has been obtained elsewhere.
C*****
```

CHARACTER CH*1

```
1000 CONTINUE
      CALL CLEAR
      WRITE (*, 6000)
6000 FORMAT(////)
      WRITE (*, 6010)
6010 FORMAT(10X,'BRLCB supports five options for preparing',
1/,10X,'the pressure-time data for the computation',
2//,15X,'1. ASCII file of time & pressure, 2 col, time/pressure',
3/,15X,'2. ASCII file of pressure, 1 col, pressure',
4/,15X,'3. ASCII file of time & voltage, 2 col, time/voltage',
5/,15X,'4. ASCII file of voltage, 1 col, voltage',
6/,15X,'5. A voltage-time file from VuPoint/BRL Procedure.',
7/,15X,'6. Exit Option.'//,
810X,'How will the pressure-time data be entered?',
9/,10X,'Enter your choice. (1-5 or 6 to EXIT)'//)
      READ (*, 5000) ICHO
5000 FORMAT(I2)
      IF (ICHO .EQ. 1) THEN
        CALL ASCII
      ELSE IF (ICHO .EQ. 2) THEN
        CALL BSCII
      ELSE IF (ICHO .EQ. 3) THEN
        CALL CSCII
      ELSE IF (ICHO .EQ. 4) THEN
        CALL DSCII
      ELSE IF (ICHO .EQ. 5) THEN
        CALL VU
      ELSE IF (ICHO .EQ. 6) THEN
        GO TO 1010
      ELSE
        WRITE (*, *) 'WARNING!! Your selection is not between 1 & 6.'
```

```

        WRITE (*, *) 'Try Again.'
        PAUSE
        GO TO 1000
    END IF
C Provisions to convert an additional file included
    CH = 'Y'
    CALL CLEAR
    WRITE (*, 6020)
6020  FORMAT(///,10X,'Do you wish to prepare an additional pressure',
1/,10X,'time file for BRLCB? [Y/N]')
    WRITE (*, 6030) CH
6030  FORMAT(' '///,10X,'Enter your choice. [',A1,']')
    READ (*, 5010) CH
5010  FORMAT(A1)
    IF ((CH.EQ. 'Y') .OR. (CH.EQ. 'y') .OR. (CH.EQ. ' ')) THEN
        GO TO 1000
    END IF
1010  CONTINUE
    END
C*****
C***** SUBROUTINE ASCII *****
C*****
    SUBROUTINE ASCII
    DIMENSION VT (2050)
    CHARACTER*20 NAME, NAME1, GAGE*6
    CALL CLEAR
    WRITE (*, 6000)
6000  FORMAT(/,10X,'Enter the file name for the pressure and',
1/,10X,'time data, include drive and extension. ',
2/,10X,'It is expected that the data will be in two',
3/,10X,'columns, time & pressure. The time step is',
4/,10X,'not important but the pressure must be in',
5/,10X,'metric units of MPa. If the pressure is not',
6/,10X,'in MPa, it must first be converted. A maximum',
7/,10X,'of 2048 time/pressure pairs is allowed.'//)
    READ (*, 5000) NAME
    CALL CLEAR
    WRITE (*, 6010)
6010  FORMAT(///,10X,'Enter the file for the output file which will',
1/,10X,'contain the pressure-time data ready for BRLCB. ',
2/,10X,'Enter file name, all DOS path conventions apply.',
3/,10X,'but there can be no extension for the file.',
4/,10X,'Note: This file name will be the name used for all',
5/,10X,'other files produced by the BRLCB analysis.',
6/,10X,'This is the file name to use when asked in subsequent',
7/,10X,'options for a file name.'//)
    READ (*, 5000) NAME1
5000  FORMAT(A20)

```

```

      CALL CKNAME (NAME1)
      CALL CLEAR
C*****
C***** GAGE INFORMATION IS STILL REQUIRED FOR THE FILE *****
C*****
      GAGE = 'None'
      AX = 0.0
      BX = 0.0
      CX = 0.0
      VIN = 0.0
C*****
C***** OUTPUT FILE IS CREATED *****
C*****
C***** Pressure-time data is read *****
C*****
      CALL CLEAR
      NV = 1
      OPEN (UNIT = 7, FILE = NAME)
      REWIND (7)
      READ (7, *) T1, VT (1)
      T1 = 0.0
      PMAX = VT (1)
      PMIN = PMAX
      WRITE (*, 6020)
6020 FORMAT(///,10X,'Enter the time step for the data in millisecs.')
```

WRITE (*, 6030)

```

6030 FORMAT(///)
      READ (*, *) DTT
      DTT = DTT/1000.
      TMIN = 0.0
1000 CONTINUE
      READ (7, *, END = 1010) T1, VT (NV + 1)
      NV = NV + 1
      IF (NV .GT. 2048) THEN
          CALL CLEAR
          WRITE (*, 6030)
          WRITE (*, *) 'There are more than 2048 points in the file.'
          WRITE (*, *) 'Delete points and start over.'
          PAUSE
          RETURN
      END IF
      IF (VT (NV) .LT. PMIN) PMIN = VT (NV)
      IF (VT (NV) .GT. PMAX) PMAX = VT (NV)
      GO TO 1000
1010 CONTINUE
      CLOSE (UNIT = 7)
      TMAX = (NV - 1) * DTT
C*****

```

C***** OUTPUT FILE IS CREATED *****
C*****

```

      OPEN (UNIT = 7, FILE = NAME1)
      REWIND (7)
      WRITE (7, 6040) NV
      WRITE (7, 6050) PMAX
      WRITE (7, 6050) PMIN
      WRITE (7, 6050) TMAX
      WRITE (7, 6050) TMIN
6040  FORMAT(' ',I5)
6050  FORMAT(' ',F15.6)
8000  FORMAT(' ',2F15.6)
      DO 1020 I = 1, NV
          T1 = (I - 1)*DTT
          WRITE (7, 8000) T1, VT (I)
1020  CONTINUE
      WRITE (7, 6060) GAGE
6060  FORMAT(' ',A6)
      WRITE (7, *) VIN
      WRITE (7, *) CX
      WRITE (7, *) BX
      WRITE (7, *) AX
      CLOSE (UNIT = 7)
      RETURN
      END

```

C*****
C***** SUBROUTINE VU *****
C*****

C Version: 3.0, January 1992

C

C Written by: William Oberle, Ballistic Research Laboratory

C

C Purpose: The purpose of this program is to convert data obtained

C from a closed chamber firing and read by VuPoint

C to the proper form in use by the smoothing and C
differentiating

C routines of BRLCB.

C

C*****

```

      SUBROUTINE VU
      DIMENSION VT (2050)
      CHARACTER*20 DRIVE*1, NAME, DRIVE1*1, NAME1, FILEC, FILEV, FPVT,
1  A*1, LINE*40, GAGE*6, G*6, H*9
      CALL CLEAR
      WRITE (*, 6000)
6000  FORMAT(///,10X,'Enter the drive where the calibration',
1/,10X,'and voltage data is stored. '//)
      READ (*, 5000) DRIVE

```

```

5000 FORMAT(A1)
      WRITE (*, 6010)
6010 FORMAT(/,10X,'Enter the file name for the calibration',
      1/,10X,'and voltage data. ' //)
      READ (*, 5010) NAME
5010 FORMAT(A20)
      WRITE (*, 6020) DRIVE
6020 FORMAT(/,10X,'Enter the drive for the output file',
      1/,10X,'which will contain pressure-time data. [' ,A1,'] ' //)
      READ (*, 5000) DRIVE1
      IF (DRIVE1 .EQ. ' ') DRIVE1 = DRIVE
      WRITE (*, 6030)
6030 FORMAT(///,10X,'Enter the file for the output file which will',
      1/,10X,'contain the pressure-time data ready for BRLCB. ',
      2/,10X,'Enter file name, there can be no extension for the file.',
      3/,10X,'Note: This file name will be the name used for all',
      4/,10X,'other files produced by the BRLCB analysis.',
      5/,10X,'This is the file name to use when asked in subsequent',
      6/,10X,'options for a file name. ' //)
      READ (*, 5010) NAME1
C*****
C***** FILE NAMES ARE DETERMINED *****
C*****
      FILEC = DRIVE
      FILEV = DRIVE
      FPVT = DRIVE1
      FILEC(2:2) = ':'
      FILEV(2:2) = ':'
      FPVT(2:2) = ':'
      FILEC(3:20) = NAME
      FILEV(3:20) = NAME
      FPVT(3:20) = NAME1
      DO 1000 I = 1, 20
        A = FILEC(I:I)
        IF (A .EQ. '.' .OR. A .EQ. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
      FILEC(I:I + 3) = 'C.AD'
      FILEV(I:I + 3) = 'V.AD'
      DO 1020 I = 1, 20
        A = FPVT(I:I)
        IF (A .EQ. '.' .OR. A .EQ. ' ') GO TO 1030
1020 CONTINUE
1030 CONTINUE
      FPVT(I:I + 3) = 'PVT'
C*****
C***** GAGE INFORMATION IS ENTERED *****
C*****

```

```

CALL CLEAR
WRITE (*, *) 'Enter the gage calibration input voltage. '
READ (*, *) VIN
1040 CONTINUE
CALL CLEAR
WRITE (*, *) ' '
1050 CONTINUE
WRITE (*, *) 'Gage information is required.'
WRITE (*, *) '1. Use gage information file.'
WRITE (*, *) '2. Enter gage information interactively.'
WRITE (*, *) 'Please Enter Your Choice (1-2):'
READ (*, *) ICHOICE
IF (ICHOICE .GT. 2) THEN
    WRITE (*, *) 'Your choice is not a 1 or 2.'
    GO TO 1040
END IF
IF (ICHOICE .EQ. 1) THEN
    CALL CLEAR
    WRITE (*, 6040)
6040 FORMAT(////,10X,'First the gage information file will be',
1/,10X,'provided to allow the proper gage ID to be determined',
2/,10X,'the gage ID will be used to select the required',
3/,10X,'information.'//)
    PAUSE
    CALL MKGAGE
    CALL CLEAR
    WRITE (*, *) 'Please enter the gage I.D. as in the gage file.'
    WRITE (*, *) 'All capital letters.'
    READ (*, 5020) GAGE
5020 FORMAT(A6)
    OPEN (UNIT = 3, FILE = 'GAGEFILE', STATUS = 'OLD')
    READ (3, *) ITN
    DO 1060 I = 1, ITN
        READ (3, 5030) G, H, CX, BX, AX
5030 FORMAT(A6,A9,3E12.5)
        IF (GAGE .EQ. G) THEN
            CLOSE (UNIT = 3)
            GO TO 1070
        END IF
1060 CONTINUE
        CLOSE (UNIT = 3)
        WRITE (*, *) 'The gage is not in the gage file, enter'
        WRITE (*, *) 'the information interactively or add to the'
        WRITE (*, *) 'the gage information to the gage file.'
        PAUSE
        CALL CLEAR
        GO TO 1050
1070 CONTINUE

```

```

        CLOSE (UNIT = 3)
    END IF
    IF (ICHOICE .EQ. 2) THEN
        CALL CLEAR
        WRITE (*, *) 'Enter the gage I.D.'
        WRITE (*, *) 'All capital letters.'
6045  FORMAT (A6)
        READ (*, 6045) GAGE
        WRITE (*, *) 'Enter the coefficients of the second order fit'
        WRITE (*, *) 'for the conversion from voltage to MPa. Start'
        WRITE (*, *) 'with the coefficient of the second power. '
        WRITE (*, *)
        WRITE (*, *) 'Enter the coefficient of the second power. '
        READ (*, *) AX
        WRITE (*, *) 'Enter the coefficient of the linear term. '
        READ (*, *) BX
        WRITE (*, *) 'Enter the constant term. '
        READ (*, *) CX
    END IF
C*****
C***** DATA CONVERSION *****
C*****
    CALL CLEAR
    WRITE (*, 6050)
6050  FORMAT('////////;', DATA CONVERSION IN PROGRESS')
    CALL POINT (FILEC, NC)
    IF (NC .GT. 2048) THEN
        CALL CLEAR
        WRITE(*,*) 'There are more than 2048 points in the input file.'
        WRITE(*,*) 'Delete sufficient points from the calibration file'
        WRITE (*, *) ' and start over.'
        PAUSE
        RETURN
    END IF
    OPEN (UNIT = 7, FILE = FILEC)
    REWIND (UNIT = 7)
    DO 1080 I = 1, 13
        READ (7, 5040) LINE
5040  FORMAT(A40)
1080  CONTINUE
        DO 1090 I = 1, NC
            READ (7, *) T1, VT (I)
1090  CONTINUE
        CLOSE (UNIT = 7)
1100  CONTINUE
        DO 1110 KK = 1, NC
            IF (VT (KK) .GT. 0.0) THEN
                N2 = KK - 3
                N3 = KK + 3

```

```

        GO TO 1120
    END IF
1110 CONTINUE
1120 CONTINUE
    N4 = NC - 3
    N1 = 3
1130 CONTINUE
    VMIN = 0.0
    VMAX = 0.0
    DO 1140 I = N1, N2
        VMIN = VMIN + VT (I)
1140 CONTINUE
    VMIN = VMIN/(N2 - N1 + 1)
    DO 1150 I = N3, N4
        VMAX = VMAX + VT (I)
1150 CONTINUE
    VMAX = VMAX/(N4 - N3 + 1)
    CALL CLEAR
    WRITE (*, *) ' '
    WRITE (*, 6060) VMIN, VMAX
6060 FORMAT(' ', 'Minimum calibration voltage = ', F10.5/,
1 ' Maximum calibration voltage = ', F10.5)
    PAUSE
C*****
C***** CONVERSION FACTOR IS COMPUTED *****
C*****
    Q = 1000.0*VIN
    F = Q/(VMAX - VMIN)
C*****
C***** VOLTAGE DATA IS READ AND CONVERTED TO MPa *****
C*****
    WRITE (*, 6070)
6070 FORMAT(///, 10X, 'Enter the time step for the data in millisecs.')
    WRITE (*, 6080)
6080 FORMAT(///)
    READ (*, *) DTT
    DTT = DTT/1000.
    TMIN = 0.0
    CALL POINT (FILEV, NV)
    IF (NV .GT. 2048) THEN
        CALL CLEAR
        WRITE(*,*) ' There are more than 2048 points in the input file.'
        WRITE(*,*) ' Delete sufficient points from the calibration file'
        WRITE (*, *) ' and start over.'
        PAUSE
        RETURN
    END IF
    OPEN (UNIT = 7, FILE = FILEV)

```

```

REWIND (7)
DO 1160 I = 1, 13
  READ (7, 5040) LINE
1160 CONTINUE
  READ (7, *) T1, VT (1)
  TEMP = VT (1) - VMIN
  TEMP = TEMP*F
  VT (1) = AX*TEMP*TEMP + BX*TEMP + CX
  PMAX = VT (1)
  PMIN = PMAX
  DO 1170 I = 2, NV
    READ (7, *) T1, VT (I)
    TEMP = VT (I) - VMIN
    TEMP = TEMP*F
    VT (I) = AX*TEMP*TEMP + BX*TEMP + CX
    IF (VT (I) .LT. PMIN) PMIN = VT (I)
    IF (VT (I) .GT. PMAX) PMAX = VT (I)
1170 CONTINUE
  CLOSE (7)
  TMAX = (NV - 1)*DTT
C*****
C***** WRITING OUTPUT FILE *****
C*****
  OPEN (UNIT = 7, FILE = FPVT)
  REWIND (7)
  WRITE (7, 6090) NV
  WRITE (7, 6100) PMAX
  WRITE (7, 6100) PMIN
  WRITE (7, 6100) TMAX
  WRITE (7, 6100) TMIN
6090 FORMAT(' ',I5)
6100 FORMAT(' ',F15.6)
8000 FORMAT(' ',2F15.6)
  DO 1180 I = 1, NV
    T1 = (I - 1)*DTT
    WRITE (7, 8000) T1, VT (I)
1180 CONTINUE
  WRITE (7, 6110) GAGE
6110 FORMAT(' ',A6)
  WRITE (7, *) VIN
  WRITE (7, *) CX
  WRITE (7, *) BX
  WRITE (7, *) AX
  CLOSE (7)
  RETURN
  END
C*****
C***** SUBROUTINE POINT *****

```

```

C*****
SUBROUTINE POINT (F, N)
CHARACTER A*40, F*20, B*21
OPEN (UNIT = 8, FILE = F)
REWIND (8)
DO 1000 I = 1, 9
  READ (8, 5000) A
5000 FORMAT(A40)
1000 CONTINUE
  READ (8, 5000) A
  BACKSPACE (8)
  IF (A (23:23) .EQ. '') THEN
    READ (8, 5010) B, N
  ELSE
    IF (A (24:24) .EQ. '') THEN
      READ (8, 5020) B, N
    ELSE
      IF (A (25:25) .EQ. '') THEN
        READ (8, 5030) B, N
      ELSE
        READ (8, 5040) B, N
      END IF
    END IF
  END IF
5010 FORMAT(A21,I1)
5020 FORMAT(A21,I2)
5030 FORMAT(A21,I3)
5040 FORMAT(A21,I4)
  CLOSE (UNIT = 8)
  RETURN
END

```

```

C*****
C***** SUBROUTINE CLEAR *****
C*****

```

```

SUBROUTINE CLEAR
CHARACTER ST*4
DATA ST/' [2J'/
WRITE (*, 6000) ST
6000 FORMAT (1X,A4)
RETURN
END

```

```

C*****
C***** SUBROUTINE MKGAGE *****
C*****

```

```

SUBROUTINE MKGAGE
C*****

```

C Version: 1.0, January 11, 1989

C

```

C Written by: William Oberle, U.S. Army Research Laboratory
C
C Purpose: This program maintains a data base for gages used
C          in the BRLCB data reduction program. The data
C          file is GAGEFILE and should be on the same drive
C          as the program.
C*****
CHARACTER AG (100)*6, BG (100)*9, XGAGE*6, YGAGE*6, XTEMP*1
DIMENSION CG (100, 3)
DO 1000 I = 1, 100
  AG (I) = ' '
  BG (I) = ' '
  CG (I, 1) = 0.0
  CG (I, 2) = 0.0
  CG (I, 3) = 0.0
1000 CONTINUE
C*****
C***** The gage data file is read. *****
C*****
CALL READER (ITN, AG, BG, CG)
C*****
C***** The number of pages is determined. *****
C*****
1010 CONTINUE
  IF ((ITN .GE. 1) .AND. (ITN .LE. 20)) NPAGE = 1
  IF ((ITN .GE. 21) .AND. (ITN .LE. 40)) NPAGE = 2
  IF ((ITN .GE. 41) .AND. (ITN .LE. 60)) NPAGE = 3
  IF ((ITN .GE. 61) .AND. (ITN .LE. 80)) NPAGE = 4
  IF ((ITN .GE. 81) .AND. (ITN .LE. 100)) NPAGE = 5
C*****
C***** The gage sub-menu is presented. *****
C*****
1020 CONTINUE
  CALL CLEAR
  WRITE (*, 6000)
6000 FORMAT(' ', T30, 'Gage Maintenance Program')
  WRITE (*, 6010)
6010 FORMAT(' ', T38, 'Sub Menu')
  WRITE (*, *)
  WRITE (*, *)
  WRITE (*, *) 1. View gage information.'
  WRITE (*, *) 2. Add a gage to data base.'
  WRITE (*, *) 3. Delete a gage from data base.'
  WRITE (*, *) 4. Locate specific gage in data base.'
  WRITE (*, *) 5. Exit.'
  WRITE (*, *)
  WRITE (*, *) Please Enter Your Choice (1-5): '
  READ (*, *) ICHOICE

```

```

IF (ICHOICE .GT. 5) THEN
  WRITE (*, *)
  WRITE (*, *) 'The choice you have made is not between'
  WRITE (*, *) '1 and 5. Please select again.'
  GO TO 1020
END IF
C*****
C***** The proper action is taken depending on the selection. *****
C*****
IF (ICHOICE .EQ. 5) GO TO 1030
IF (ICHOICE .EQ. 1) THEN
  IF (ITN .EQ. 0) THEN
    CALL CLEAR
    WRITE(*,*)'There are no gages in the gage information file.'
    WRITE (*, *) 'Enter another option.'
    PAUSE
    GO TO 1020
  END IF
  CALL CLEAR
  WRITE (*, *)
1040  CONTINUE
  WRITE (*, 6020) ITN
6020  FORMAT(' ', 'There are ', I3, ' entries in the gage data file.')
  WRITE (*, 6030) NPAGE
6030  FORMAT(' ', 'This information is displayed in ', I3, ' screen pages.')
  WRITE (*, 6040)
6040  FORMAT(' ', 'Which page would you like to view? ')
  READ (*, *) IPAGE
  IF (IPAGE .GT. NPAGE) THEN
    WRITE (*, *)
    WRITE (*, *) 'The choice you have made is not in'
    WRITE (*, *) 'the proper range. Please select again.'
    GO TO 1040
  END IF
  CALL HEADER
  CALL PAGER (ITN, IPAGE, AG, BG, CG)
  GO TO 1010
C*****
C***** The end of option 1. *****
C*****
END IF
C*****
C***** Option 2 *****
C*****
IF (ICHOICE .EQ. 2) THEN
  CALL GAGEADD (ITN, AG, BG, CG)
  GO TO 1010
END IF

```

```

C*****
C***** Option 3 *****
C*****
  IF (ICHOICE .EQ. 3) THEN
    CALL DROPPER (ITN, AG, BG, CG)
    GO TO 1010
  END IF
C*****
C***** Option 4 *****
C*****
  IF (ICHOICE .EQ. 4) THEN
    CALL CLEAR
1050  CONTINUE
    WRITE (*, *) 'Enter the I.D. number of the desired gage. '
    WRITE (*, *) 'Enter all letters as capital letters.'
    READ (*, 5000) XGAGE
5000  FORMAT(A6)
1060  CONTINUE
    YGAGE = ' '
    IF (XGAGE .EQ. ' ') GO TO 1050
C*****
C***** Test is determine if xgage starts with a blank *****
C*****
    IF (XGAGE (1:1) .EQ. ' ') THEN
      DO 1070 II = 2, 6
        YGAGE (II - 1:II - 1) = XGAGE (I:I)
1070  CONTINUE
        XGAGE = YGAGE
        GO TO 1060
      END IF
C*****
C***** Now the search begins. *****
C*****
    DO 1080 I = 1, ITN
      IF (XGAGE .EQ. AG (I)) THEN
        WRITE (*, *) 'A match has been found.'
        WRITE (*, *)
        WRITE (*, *)
        PAUSE
        CALL HEADER
        WRITE (*, 6050) I, AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
6050  FORMAT(' ', I3, 2X, A6, 2X, A9, 3X, E12.5, 6X, E12.5, 6X, E12.5)
        WRITE (*, *)
        WRITE (*, *) 'Press Enter to Continuc. '
        READ (*, 5010) XTEMP
5010  FORMAT(A1)
        GO TO 1010
      END IF
    END DO
  END IF

```

```

        END IF
1080  CONTINUE
        WRITE (*, *) 'The gage is not in the table.'
        WRITE (*, *) 'Press Enter to Continue. '
        READ (*, 5010) XTEMP
        GO TO 1010
C*****
C***** This is the end of option 4. *****
C*****
        END IF
C*****
C***** To exit the program the gage file is written. *****
C*****
1030 CONTINUE
        OPEN (UNIT = 8, FILE = 'GAGEFILE')
        WRITE (8, *) ITN
        DO 1090 I = 1, ITN
            WRITE (8, 6060) AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1090 CONTINUE
6060  FORMAT(A6,A9,3E12.5)
        CLOSE (UNIT = 8)
        RETURN
        END
C*****
        SUBROUTINE HEADER
C*****
        CALL CLEAR
        WRITE (*, *) ' The data fit is of the form: A + Bx + Cx^2'
        WRITE (*, 6000)
6000  FORMAT(' ', ' #', 4X, 'I.D.', 3X, 'Cal. Date', 8X, 'A', 18X, 'B', 18X, 'C')
        WRITE (*, 6010)
6010  FORMAT(' ', 80(' - '))
        RETURN
        END
C*****
        SUBROUTINE PAGER (ITN, IPAGE, AG, BG, CG)
C*****
        CHARACTER AG (100)*6, BG (100)*9
        DIMENSION CG (100, 3)
C*****
C***** Test to see if there are entries on the chosen page. *****
C*****
1000 CONTINUE
        IX = 20*(IPAGE - 1)
        IF (IX .LT. ITN) GO TO 1010
        WRITE (*, *) 'There are no entries on the page selected.'
1020 CONTINUE
        WRITE (*, *) 'Please enter a new desired page number. '

```

```

      READ (*, *) IPAGE
      CALL HEADER
      GO TO 1000
1010 CONTINUE
      ISTART = 20*(IPAGE - 1) + 1
      IEND = MIN0 (20*IPAGE, ITN)
      DO 1030 I = ISTART, IEND
          WRITE (*, 6000) I, AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1030 CONTINUE
6000 FORMAT(' ',I3,2X,A6,2X,A9,3X,E12.5,6X,E12.5,6X,E12.5)
1040 CONTINUE
      WRITE (*, *) 'New page (1) or options (2)? '
      READ (*, *) IOPT
      IF (IOPT .GT. 2) THEN
          WRITE (*, *)
          WRITE (*, *) 'The choice you have made is not a'
          WRITE (*, *) '1 or 2. Please select again.'
          GO TO 1040
      END IF
      IF (IOPT .EQ. 1) GO TO 1020
      RETURN
      END
C*****
      SUBROUTINE READER (ITN, AG, BG, CG)
C*****
      CHARACTER AG (100)*6, BG (100)*9
      DIMENSION CG (100, 3)
      OPEN (UNIT = 7, FILE = 'GAGEFILE')
      REWIND 7
C*****
C***** The gage file is opened and read *****
C*****
      READ (7, *) ITN
      IF (ITN .EQ. 0) THEN
          WRITE (*, *) 'The gage file is empty.'
          CLOSE (UNIT = 7)
          RETURN
      END IF
      DO 1000 I = 1, ITN
          READ (7, 5000) AG (I), BG (I), CG (I, 1), CG (I, 2), CG (I, 3)
1000 CONTINUE
5000 FORMAT(A6,A9,3E12.5)
      CLOSE (UNIT = 7)
      RETURN
      END
C*****
      SUBROUTINE GAGEADD (ITN, AG, BG, CG)

```

```

C*****
CHARACTER AG (100)*6, BG (100)*9, XGAGE*6, YGAGE*6
DIMENSION CG (100, 3)
ITN = ITN + 1
IF (ITN .GT. 100) THEN
  WRITE (*, *) 'No room left in the gage data file.'
  WRITE (*, *) 'Return to menu and delete unwanted gages.'
  PAUSE
  RETURN
END IF
CALL CLEAR
1000 CONTINUE
  WRITE (*, *) ' Please enter the gage ID. '
  WRITE (*, *) ' Enter all letters as capital letters.'
  READ (*, 5000) XGAGE
5000 FORMAT(A6)
1010 CONTINUE
  YGAGE = ' '
  IF (XGAGE .EQ. ' ') GO TO 1000
C*****
C***** Test is determine if xgage starts with a blank *****
C*****
  IF (XGAGE (1:1) .EQ. ' ') THEN
    DO 1020 II = 2, 6
      YGAGE (II - 1:II - 1) = XGAGE (I:I)
1020 CONTINUE
      XGAGE = YGAGE
      GO TO 1010
    END IF
    AG (ITN) = XGAGE
    WRITE (*, *) 'Enter the date of gage calibration (mm/dd/yy). '
    READ (*, 5010) BG (ITN)
5010 FORMAT(A9)
    WRITE (*, *) 'Calibration constants must produce pressure in MPa'
    WRITE (*, *) 'The form of the data fit is:  $P(\text{MPa}) = A + Bx + Cx^2$ '
    WRITE (*, *) 'Enter the value for A: '
    READ (*, *) CG (ITN, 1)
    WRITE (*, *) 'Enter the value for B: '
    READ (*, *) CG (ITN, 2)
    WRITE (*, *) 'Enter the value for C: '
    READ (*, *) CG (ITN, 3)
C*****
C Now a search is performed to determine if the new entry supercedes
C a earlier data set.
C*****
  DO 1030 I = 1, ITN - 1
    IF (XGAGE .EQ. AG (I)) THEN
      WRITE (*, *) 'The gage added to the data base was already in'

```

```

        WRITE(*,*)'the data base. The earlier entry will be tagged'
        WRITE (*, *)'with a "*" as the last character of the I.D. field.'
        XGAGE (6:6) = '*'
        AG (I) = XGAGE
        WRITE (*, *)
        PAUSE
    END IF
1030 CONTINUE
    RETURN
    END
C*****
    SUBROUTINE DROPPER (ITN, AG, BG, CG)
C*****
        CHARACTER AG (100)*6, BG (100)*9
        DIMENSION CG (100, 3)
        CALL CLEAR
        WRITE (*, *) 'Enter the number for the line in the gage data file'
        WRITE (*, *) 'to be deleted. This is not the gage I.D. '
        READ (*, *) IDEL.
C*****
C***** Shifting the entries in the table. *****
C*****
        DO 1000 I = IDEL + 1, ITN
            AG (I - 1) = AG (I)
            BG (I - 1) = BG (I)
            CG (I - 1, 1) = CG (I, 1)
            CG (I - 1, 2) = CG (I, 2)
            CG (I - 1, 3) = CG (I, 3)
        1000 CONTINUE
        ITN = ITN - 1
        RETURN
    END
C*****
C***** SUBROUTINE BSCII *****
C*****
    SUBROUTINE BSCII
        DIMENSION VT (2050)
        CHARACTER*20 NAME, NAME1, GAGE*6
        CALL CLEAR
        WRITE (*, 6000)
6000 FORMAT(/,10X,'Enter the file name for the pressure data,',
            1/,10X,'include drive and extension. ',
            2/,10X,'It is expected that the data will be in one',
            3/,10X,'column, pressure. The pressure must be in',
            4/,10X,'metric units of MPa. If the pressure is not',
            5/,10X,'in MPa, it must first be converted. A maximum',

```

```

6/,10X,'of 2048 pressure points is allowed.'//)
READ (*, 5000) NAME
CALL CLEAR
WRITE (*, 6010)
6010 FORMAT(///,10X,'Enter the file for the output file which will',
1/,10X,'contain the pressure-time data ready for BRLCB. ',
2/,10X,'Enter file name, all DOS path conventions apply.',
3/,10X,'but there can be no extension for the file.',
4/,10X,'Note: This file name will be the name used for all',
5/,10X,'other files produced by the BRLCB analysis.',
6/,10X,'This is the file name to use when asked in subsequent',
7/,10X,'options for a file name.'//)
READ (*, 5000) NAME1
5000 FORMAT(A20)
CALL CKNAME (NAME1)
CALL CLEAR
C*****
C***** GAGE INFORMATION IS STILL REQUIRED FOR THE FILE *****
C*****
GAGE = 'None'
AX = 0.0
BX = 0.0
CX = 0.0
VIN = 0.0
C*****
C***** OUTPUT FILE IS CREATED *****
C*****
C***** Pressure-time data is read *****
C*****
CALL CLEAR
NV = 1
OPEN (UNIT = 7, FILE = NAME)
REWIND (7)
READ (7, *) VT (1)
T1 = 0.0
PMAX = VT (1)
PMIN = PMAX
WRITE (*, 6020)
6020 FORMAT(///,10X,'Enter the time step for the data in millisecs.')
WRITE (*, 6030)
6030 FORMAT(///)
READ (*, *) DTT
DTT = DTT/1000.
TMIN = 0.0
1000 CONTINUE
READ (7, *, END = 1010) VT (NV + 1)
NV = NV + 1
IF (NV .GT. 2048) THEN

```

```

CALL CLEAR
WRITE(*,*) ' There are more than 2048 points in the input file.'
WRITE (*, *) ' Delete sufficient points and start over.'
PAUSE
RETURN
END IF
IF (VT (NV) .LT. PMIN) PMIN = VT (NV)
IF (VT (NV) .GT. PMAX) PMAX = VT (NV)
GO TO 1000
1010 CONTINUE
CLOSE (UNIT = 7)
TMAX = (NV - 1)*DTT
C*****
C***** OUTPUT FILE IS CREATED *****
C*****
OPEN (UNIT = 7, FILE = NAME1)
REWIND (7)
WRITE (7, 6040) NV
WRITE (7, 6050) PMAX
WRITE (7, 6050) PMIN
WRITE (7, 6050) TMAX
WRITE (7, 6050) TMIN
6040 FORMAT(' ',I5)
6050 FORMAT(' ',F15.6)
8000 FORMAT(' ',2F15.6)
DO 1020 I = 1, NV
T1 = (I - 1)*DTT
WRITE (7, 8000) T1, VT (I)
1020 CONTINUE
WRITE (7, 6060) GAGE
6060 FORMAT(' ',A6)
WRITE (7, *) VIN
WRITE (7, *) CX
WRITE (7, *) BX
WRITE (7, *) AX
CLOSE (UNIT = 7)
RETURN
END
C*****
C***** SUBROUTINE CSCII *****
C*****
SUBROUTINE CSCII
DIMENSION VT (2050)
CHARACTER*20 NAME, NAME1, GAGE*6, G*6
CALL CLEAR
WRITE (*, 6000)
6000 FORMAT(/,10X,'Enter the file name for the voltage and',
1/,10X,'time data, include drive and extension. ',

```

```

2/,10X,'It is expected that the data will be in two',
3/,10X,'columns, time & voltage. A maximum',
4/,10X,'of 2048 time/pressure pairs is allowed.'//)
READ (*, 5000) NAME
CALL CLEAR
WRITE (*, 6010)
6010 FORMAT(///,10X,'Enter the file for the output file which will',
1/,10X,'contain the pressure-time data ready for BRLCB. ',
2/,10X,'Enter file name, all DOS path conventions apply,',
3/,10X,'but there can be no extension for the file.',
4/,10X,'Note: This file name will be the name used for all',
5/,10X,'other files produced by the BRLCB analysis.',
6/,10X,'This is the file name to use when asked in subsequent',
7/,10X,'options for a file name.'//)
READ (*, 5000) NAME1
5000 FORMAT(A20)
CALL CKNAME (NAME1)
C*****
C***** GAGE INFORMATION IS ENTERED *****
C*****
CALL CLEAR
VIN = 0.0
1000 CONTINUE
CALL CLEAR
WRITE (*, *) ' '
1010 CONTINUE
WRITE (*, *) 'Gage information is required.'
WRITE (*, *) '1. Input gage ID & use Gage File.'
WRITE (*, *) '2. Review Gage File.'
WRITE (*, *) '3. Enter gage information interactively.'
WRITE (*, *) 'Please Enter Your Choice (1-3):'
READ (*, *) ICHOICE
IF (ICHOICE .GT. 3) THEN
WRITE (*, *) 'Your choice is not a 1, 2 or 3.'
GO TO 1000
END IF
IF (ICHOICE .EQ. 2) THEN
CALL CLEAR
WRITE (*, 6020)
6020 FORMAT(///,10X,'First the gage information file will be',
1/,10X,'provided to allow the proper gage ID to be determined',
2/,10X,'the gage ID will be used to select the required',
3/,10X,'information.'//)
PAUSE
CALL MKGAGE
END IF
IF ((ICHOICE .EQ. 1) .OR. (ICHOICE .EQ. 2)) THEN
CALL CLEAR

```

```

WRITE (*, *) 'Please enter the gage I.D. as in the gage file.'
WRITE (*, *) 'All capital letters.'
READ (*, 5010) GAGE
5010 FORMAT(A6)
OPEN (UNIT = 3, FILE = 'GAGEFILE', STATUS = 'OLD')
READ (3, *) ITN
DO 1020 I = 1, ITN
    READ (3, 5020) G, H, CX, BX, AX
5020 FORMAT(A6,A9,3E12.5)
    IF (GAGE .EQ. G) THEN
        GO TO 1030
    END IF
1020 CONTINUE
CLOSE (UNIT = 3)
WRITE (*, *) 'The gage is not in the gage file, enter'
WRITE (*, *) 'the information interactively or add to the'
WRITE (*, *) 'the gage information to the gage file.'
PAUSE
CALL CLEAR
GO TO 1010
1030 CONTINUE
CLOSE (UNIT = 3)
END IF
IF (ICHOICE .EQ. 3) THEN
    CALL CLEAR
    WRITE (*, *) 'Enter the gage I.D.'
    WRITE (*, *) 'All capital letters.'
    READ (*, 5010) GAGE
    WRITE (*, *) 'Enter the coefficients of the second order fit'
    WRITE (*, *) 'for the conversion from voltage to MPa. Start'
    WRITE (*, *) 'with the coefficient of the second power. '
    WRITE (*, *)
    WRITE (*, *) 'Enter the coefficient of the second power. '
    READ (*, *) AX
    WRITE (*, *) 'Enter the coefficient of the linear term. '
    READ (*, *) BX
    WRITE (*, *) 'Enter the constant term. '
    READ (*, *) CX
END IF
C*****
C***** DATA CONVERSION *****
C*****
    CALL CLEAR
C*****
C***** VOLTAGE DATA IS READ AND CONVERTED TO MPa *****
C*****
    CALL CLEAR
    WRITE (*, 6030)

```

```

6030 FORMAT(///,10X,'Enter the time step for the data in millisecs.')
      WRITE (*, 6040)
6040 FORMAT(///)
      READ (*, *) DTT
      DTT = DTT/1000.
      CALL CLEAR
      WRITE (*, 6050)
6050 FORMAT(////////, '          DATA CONVERSION IN PROGRESS')
      TMIN = 0.0
      OPEN (UNIT = 7, FILE = NAME)
      REWIND (7)
      NV = 1
      READ (7, *) T1, VT (1)
      TEMP = VT (1)
      VT (1) = AX*TEMP*TEMP + BX*TEMP + CX
      PMAX = VT (1)
      PMIN = PMAX
1040 CONTINUE
      READ (7, *, END = 1050) T1, VT (NV + 1)
      NV = NV + 1
      IF (NV .GT. 2048) THEN
        CALL CLEAR
        WRITE (*, *) 'There are more than 2048 points in the file.'
        WRITE (*, *) 'Delete points and start over.'
        PAUSE
        RETURN
      END IF
      TEMP = VT (NV)
      VT (NV) = AX*TEMP*TEMP + BX*TEMP + CX
      IF (VT (NV) .LT. PMIN) PMIN = VT (NV)
      IF (VT (NV) .GT. PMAX) PMAX = VT (NV)
      GO TO 1040
1050 CONTINUE
      CLOSE (7)
      TMAX = (NV - 1)*DTT
C*****
C***** WRITING OUTPUT FILE *****
C*****
      OPEN (UNIT = 7, FILE = NAME1)
      REWIND (7)
      WRITE (7, 6060) NV
      WRITE (7, 6070) PMAX
      WRITE (7, 6070) PMIN
      WRITE (7, 6070) TMAX
      WRITE (7, 6070) TMIN
6060 FORMAT(' ',I5)
6070 FORMAT(' ',F15.6)
8000 FORMAT(' ',2F15.6)

```

```

DO 1060 I = 1, NV
  T1 = (I - 1)*DTT
  WRITE (7, 8000) T1, VT (I)
1060 CONTINUE
  WRITE (7, 6080) GAGE
6080 FORMAT(' ',A6)
  WRITE (7, *) VIN
  WRITE (7, *) CX
  WRITE (7, *) BX
  WRITE (7, *) AX
  CLOSE (7)
  RETURN
END

```

```

C*****
C***** SUBROUTINE DSCII *****
C*****

```

```

  SUBROUTINE DSCII
  DIMENSION VT (2050)
  CHARACTER*20 NAME, NAME1, GAGE*6, G*6
  CALL CLEAR
  WRITE (*, 6000)
6000 FORMAT(//,10X,'Enter the file name for the voltage data',
  1/,10X,'include drive and extension. ',
  2/,10X,'It is expected that the data will be in one',
  3/,10X,'column, voltage. A maximum',
  4/,10X,'of 2048 voltage points is allowed.'//)
  READ (*, 5000) NAME
  CALL CLEAR
  WRITE (*, 6010)
6010 FORMAT(////,10X,'Enter the file for the output file which will',
  1/,10X,'contain the pressure-time data ready for BRLCB. ',
  2/,10X,'Enter file name, all DOS path conventions apply.',
  3/,10X,'but there can be no extension for the file.',
  4/,10X,'Note: This file name will be the name used for all',
  5/,10X,'other files produced by the BRLCB analysis.',
  6/,10X,'This is the file name to use when asked in subsequent',
  7/,10X,'options for a file name.'//)
  READ (*, 5000) NAME1
5000 FORMAT(A20)
  CALL CKNAME (NAME1)

```

```

C*****
C***** GAGE INFORMATION IS ENTERED *****
C*****

```

```

  CALL CLEAR
  VIN = 0.0
  J CONTINUE
  CALL CLEAR
  WRITE (*, *) ' '

```

```

1010 CONTINUE
    WRITE (*, *) 'Gage information is required.'
    WRITE (*, *) '1. Input gage ID & use Gage File.'
    WRITE (*, *) '2. Review Gage File.'
    WRITE (*, *) '3. Enter gage information interactively.'
    WRITE (*, *) 'Please Enter Your Choice (1-3):'
    READ (*, *) ICHOICE
    IF (ICHOICE .GT. 3) THEN
        WRITE (*, *) 'Your choice is not a 1, 2 or 3.'
        GO TO 1000
    END IF
    IF (ICHOICE .EQ. 2) THEN
        CALL CLEAR
        WRITE (*, 6020)
6020  FORMAT(////,10X,'First the gage information file will be',
           1/,10X,'provided to allow the proper gage ID to be determined',
           2/,10X,'the gage ID will be used to select the required',
           3/,10X,'information.'//)
        PAUSE
        CALL MKGAGE
    END IF
    IF ((ICHOICE .EQ. 1) .OR. (ICHOICE .EQ. 2)) THEN
        CALL CLEAR
        WRITE (*, *) 'Please enter the gage I.D. as in the gage file.'
        WRITE (*, *) 'All capital letters.'
        READ (*, 5010) GAGE
5010  FORMAT(A6)
        OPEN (UNIT = 3, FILE = 'GAGEFILE', STATUS = 'OLD')
        READ (3, *) ITN
        DO 1020 I = 1, ITN
            READ (3, 5020) G, H, CX, BX, AX
5020  FORMAT(A6,A9,3E12.5)
            IF (GAGE .EQ. G) THEN
                GO TO 1030
            END IF
1020  CONTINUE
            CLOSE (UNIT = 3)
            WRITE (*, *) 'The gage is not in the gage file, enter'
            WRITE (*, *) 'the information interactively or add to the'
            WRITE (*, *) 'the gage information to the gage file.'
            PAUSE
            CALL CLEAR
            GO TO 1010
1030  CONTINUE
            CLOSE (UNIT = 3)
        END IF
        IF (ICHOICE .EQ. 3) THEN
            CALL CLEAR

```

```

WRITE (*, *) 'Enter the gage I.D.'
WRITE (*, *) 'All capital letters.'
READ (*, 5010) GAGE
WRITE (*, *) 'Enter the coefficients of the second order fit'
WRITE (*, *) 'for the conversion from voltage to MPa. Start'
WRITE (*, *) 'with the coefficient of the second power. '
WRITE (*, *)
WRITE (*, *) 'Enter the coefficient of the second power. '
READ (*, *) AX
WRITE (*, *) 'Enter the coefficient of the linear term. '
READ (*, *) BX
WRITE (*, *) 'Enter the constant term. '
READ (*, *) CX
END IF
C*****
C***** DATA CONVERSION *****
C*****
CALL CLEAR
C*****
C***** VOLTAGE DATA IS READ AND CONVERTED TO MPa *****
C*****
CALL CLEAR
WRITE (*, 6030)
6030 FORMAT(///,10X,'Enter the time step for the data in millisecs.')
WRITE (*, 6040)
6040 FORMAT(///)
READ (*, *) DTT
DTT = DTT/1000.
CALL CLEAR
WRITE (*, 6050)
6050 FORMAT(////////,' DATA CONVERSION IN PROGRESS')
TMIN = 0.0
NV = 1
OPEN (UNIT = 7, FILE = NAME)
REWIND (7)
READ (7, *) VT (1)
TEMP = VT (1)
VT (1) = AX*TEMP*TEMP + BX*TEMP + CX
PMAX = VT (1)
PMIN = PMAX
1040 CONTINUE
READ (7, *, END = 1050) VT (NV + 1)
NV = NV + 1
IF (NV .GT. 2048) THEN
CALL CLEAR
WRITE(*,*) ' There are more than 2048 points in the input file.'
WRITE (*, *) ' Delete sufficient points and start over.'
PAUSE

```

```

        RETURN
    END IF
    TEMP = VT (NV)
    VT (NV) = AX*TEMP*TEMP + BX*TEMP + CX
    IF (VT (NV) .LT. PMIN) PMIN = VT (NV)
    IF (VT (NV) .GT. PMAX) PMAX = VT (NV)
    GO TO 1040
1050 CONTINUE
    CLOSE (UNIT = 7)
    TMAX = (NV - 1)*DTT
C*****
C***** WRITING OUTPUT FILE *****
C*****
    OPEN (UNIT = 7, FILE = NAME1)
    REWIND (7)
    WRITE (7, 6060) NV
    WRITE (7, 6070) PMAX
    WRITE (7, 6070) PMIN
    WRITE (7, 6070) TMAX
    WRITE (7, 6070) TMIN
6060 FORMAT(' ',I5)
6070 FORMAT(' ',F15.6)
8000 FORMAT(' ',2F15.6)
    DO 1060 I = 1, NV
        T1 = (I - 1)*DTT
        WRITE (7, 8000) T1, VT (I)
1060 CONTINUE
    WRITE (7, 6080) GAGE
6080 FORMAT(' ',A6)
    WRITE (7, *) VIN
    WRITE (7, *) CX
    WRITE (7, *) BX
    WRITE (7, *) AX
    CLOSE (7)
    RETURN
END
C*****
C***** SUBROUTINE CKNAME *****
C*****
    SUBROUTINE CKNAME (NAME1)
    CHARACTER*20 NAME1, A*1
1000 CONTINUE
    DO 1010 I = 1, 20
        IF (I .GT. 15) THEN
            WRITE (*, *)'File name too long. Please enter a new name which'
            WRITE (*, *)'is less than six characters in length, not counting'

```

```

        WRITE (*, *) 'paths.'
        READ (*, 5000) NAME1
        GO TO 1000
    END IF
    A = NAME1 (I:I)
    IF (A .EQ. '.') THEN
        CALL CLEAR
        WRITE (*, *) 'The file name entered has an extension which is'
        WRITE (*, *) 'not allowed. Please enter a new name without '
        WRITE (*, *) 'an extension.'
        READ (*, 5000) NAME1
5000  FORMAT(A20)
        GO TO 1000
    END IF
    IF (A .EQ. ' ') THEN
        NAME1 (I:I + 3) = '.PVT'
        GO TO 1020
    END IF
1010 CONTINUE
1020 CONTINUE
    RETURN
    END

```

INTENTIONALLY LEFT BLANK.

APPENDIX I:
LISTING - PROGRAM MKINF.FOR

INTENTIONALLY LEFT BLANK.

```

PROGRAM MKINF
CHARACTER*20 NAME1, FPVT, A*1, NAME2, FILEO, FILEI, A1 (20)*20, A2 (6)*80
CHARACTER*20 GAGE
DIMENSION A3 (100), P (11, 15, 5)
C*****
C Version: 3.0, January 1992
C       Last Modified: 12/28/91 -- added counter to burn rate
C                               file
C                               2/9/92 -- Cleanup
C
C Written by: William Oberle, U.S. Army Research Laboratory
C
C Purpose: The purpose of this program is to obtain the necessary
C          additional information required to analyze a closed
C          chamber firing.
C*****
C***** FILE NAMES ARE OBTAINED AND *****
C***** MASTER FILE IS READ *****
C*****
1000 CONTINUE
      CALL CLEAR
      WRITE (*, 6000)
6000 FORMAT(20X, ' Firing Information'///,
110X, 'Information pertaining to an individual firing is entered'
2/,10X, 'using this option. The file is built from a Master'
3/,10X, 'File created using Option 1. Enter the name of the'
4/,10X, 'Master File which will be used in creating the current'
5/,10X, 'Information File, include drive and extension.'//)
      READ (*, 5000) NAME1
5000 FORMAT(A20)
      OPEN (UNIT = 9, FILE = NAME1, STATUS = 'OLD', ERR = 1010)
      REWIND (UNIT = 9)
C*****
C***** READING OLD MASTER FILE TO BUILD FROM *****
C*****
      DO 1020 I = 1, 6
        READ (9, 5010) A2 (I)
1020 CONTINUE
5010 FORMAT(A80)

```

```

DO 1030 I = 1, 20
  READ (9, 5000) A1 (I)
1030 CONTINUE
  DO 1040 I = 1, 100
    READ (9, *) A3 (I)
1040 CONTINUE
    DO 1050 I = 1, 11
      DO 1060 J = 1, 15
        DO 1070 K = 1, 5
          READ (9, *) P (I, J, K)
1070 CONTINUE
1060 CONTINUE
1050 CONTINUE
    CLOSE (UNIT = 9)
    GO TO 1080
1010 CONTINUE
  WRITE (*, 6010) NAME1
6010 FORMAT(//,10X,'The Master File: ',A20/,
  110X,'does not exist, check the file name and try again.')
  PAUSE
  GO TO 1000
C*****
C***** TYPE OF COMPUTATION IS DETERMINED *****
C*****
1080 CONTINUE
  CALL CLEAR
  WRITE (*, 6020)
6020 FORMAT(//,10X,'Enter the type of computation which is to be',
  1/,10X,'performed.'/,15X,'1. Burn Rate Reduction'/,15X,
  2'2. Inverse Analysis (Generate P/T)'/,15X,
  3'3. Surface Area Analysis'/,15X,'4. Interrupted Burner'/,15X,
  4'5. ETC Reduction')
  READ (*, *) ITYPE
  A3 (2) = ITYPE
C*****
C***** ALL OTHER FILE NAMES ARE CREATED *****
C*****
  IF (ITYPE .NE. 2) THEN
1090 CONTINUE
    CALL CLEAR
    WRITE (*, 6030)
6030 FORMAT(20X,' Firing Information'///,
  110X,'Enter the name of the Pressure/time File created in'
  2/,10X,'Option 3 which is associated with the Information File'
  3/,10X,'being created. This file must be created before',
  4/,10X,'the Information File is created. If the Pressure/time'
  5/,10X,'File does not exist, exit the program and complete'
  6/,10X,'Option 3. Remember the file name is given without '

```

```

7/,10X,'extension but all DOS path options are applicable.'//)
  READ (*, 5000) FPVT
  A1 (5) = FPVT
  CALL MKNAME (A1)
  OPEN (UNIT = 7, FILE = A1 (5), STATUS = 'OLD', ERR = 1100)
  REWIND (UNIT = 7)
  READ (7, *) NV
  READ (7, 5020) PMAX
  A3 (43) = PMAX
  READ (7, 5020) PMIN
  READ (7, 5020) TMAX
  READ (7, 5020) TMIN
5020 FORMAT(F15.6)
8000 FORMAT(2F15.6)
  DO 1110 I = 1, NV
    READ (7, 8000) T1, P1
1110  CONTINUE
    READ (7, 5030) GAGE
5030 FORMAT(A20)
    A1 (15) = GAGE
    READ (7, *) VIN
    A3 (19) = VIN
    READ (7, *) CX
    A3 (20) = CX
    READ (7, *) BX
    A3 (21) = BX
    READ (7, *) AX
    A3 (22) = AX
    CLOSE (UNIT = 7)
    CALL CLEAR
    DELTAT = (TMAX - TMIN)/(NV - 1.)*1000.
    CALL CLEAR
    WRITE (*, 6040) DELTAT
6040 FORMAT(///,10X,'The time step in recording the pressure data'
1/,10X,'is currently: ',F9.6,' ms. If this is not correct enter'
2/,10X,'a new value for the time step in milliseconds.'//,
315X,'1. Accept current value.'/,15X,'2. Enter new value.'/,
415X,'Enter a 1 or 2.'/)
    READ (*, *) IIC
    IF (IIC .EQ. 2) THEN
      WRITE (*, 6050)
6050 FORMAT(20X,'Enter a new time step in milliseconds.')
      READ (*, *) A3 (42)
    ELSE
      A3 (42) = DELTAT
    END IF
    GO TO 1120
1100  CONTINUE

```

```

        WRITE (*, 6060) A1 (5)
6060  FORMAT(/,10X,'The Pressure/time File: ',A20/,
        110X,'does not exist, check the file name and try again.')
        PAUSE
        GO TO 1090
1120  CONTINUE
        END IF
C*****
C***** OPTION 2 IS HANDLED *****
C*****
        IF (ITYPE.EQ. 2) THEN
            CALL CLEAR
            WRITE (*, 6070)
6070  FORMAT(/,10X,'Enter the file name for the file which is to',
            1/,10X,'contain the information for the pressure-time',
            2/,10X,'generation. The file is to have no extension, but all',
            3/,10X,'DOS path conventions are applicable.'//)
            READ (*, 5040) FPVT
5040  FORMAT(A20)
            A1 (5) = FPVT
            CALL MKNAME (A1)
            CALL CLEAR
            WRITE (*, 6080)
6080  FORMAT(///,10X,'In order to generate a pressure time file',
            1/,10X,'the time step in milliseconds must be entered. Enter',
            2/,10X,'the time step.')
            READ (*, *) A3 (42)
        END IF
C*****
C***** INFORMATION FILE IS OPENED *****
C*****
        OPEN (UNIT = 13, FILE = A1 (4))
        REWIND (UNIT = 13)
C*****
C***** SPECIAL INFORMATION FOR THE ETC FIRING *****
C*****
        IF (ITYPE.EQ. 5) THEN
            CALL CLEAR
            WRITE (*, 6090)
6090  FORMAT(/,10X,'For the ETC reduction a file of cumulative',
            1/,10X,'electrical energy versus time is required. The time',
            2/,10X,'must match the time in the pressure-time file, the',
            3/,10X,'units are seconds. The electrical energy is in units',
            4/,10X,'of megajoules. Enter the file name.')
            READ (*, 5050) A1 (19)
5050  FORMAT(A20)
            CALL CLEAR
            WRITE (*, 6100)

```

```

6100 FORMAT(/,10X,'Enter the total electrical energy input in MJ')
      READ (*, *) A3 (30)
      END IF
C*****
C***** FILE INFORMATION IS PROVIDED TO USER *****
C*****
      CALL CLEAR
      WRITE (*, 6110)
6110 FORMAT(/,5X,'The following file names will be used:')
      WRITE (*, 6120) A1 (3), A1 (4)
6120 FORMAT(5X,'Master File      : ',A20,' Created in Option 1',
1/,5X,'Information File : ',A20,' Created Now in Option 4')
      IF (ITYPE .NE. 2) THEN
        WRITE (*, 6130) A1 (5), A1 (14)
6130 FORMAT(5X,'Pressure/Time File: ',A20,' Created in Option 3',
1/,5X,'Smoothed Data File: ',A20,' Created in Option 5')
      END IF
      WRITE (*, 6140) A1 (10), A1 (17)
6140 FORMAT(5X,'Output File      : ',A20,' Created in Option 6',
1/,5X,'Graphics File      : ',A20,' Created in Option 6')
      IF ((ITYPE .EQ. 1) .OR. (ITYPE .EQ. 4) .OR. (ITYPE .EQ. 5)) THEN
        WRITE (*, 6150) A1 (18)
6150 FORMAT(5X,'Burn Rate File   : ',A20,' Created in Option 6')
      END IF
      IF (ITYPE .EQ. 5) THEN
        WRITE (*, 6160) A1 (19)
6160 FORMAT(5X,'Electrical Energy : ',A20,' Used in Option 6')
      END IF
      PAUSE
C*****
C***** BOMB, PROP & IGNITER INFO IS OBTAINED *****
C*****
      CALL CLEAR
      WRITE (*, *) ' Bomb Information'
      WRITE (*, *)
      WRITE (*, *) 'Enter the bomb volume (cc): '
      READ (*, *) A3 (23)
      WRITE (*, *)
      WRITE (*, *) 'Enter the initial temperature of'
      WRITE (*, *) 'the bomb in degrees Kelvin.'
      WRITE (*, *) 'Several common temperatures are:'
      WRITE (*, *) '      -25 F = 241 K'
      WRITE (*, *) '      -20 F = 244 K'
      WRITE (*, *) '      70 F = 294 K'
      WRITE (*, *) '      120 F = 322 K'
      WRITE (*, *)
      WRITE (*, *) 'Please Enter Your Choice: '
      READ (*, *) A3 (24)

```

```

      IF (ITYPE .EQ. 4) THEN
        A1 (16) = 'Interrupted Burner'
      ELSE
        A1 (16) = 'Closed Chamber'
      END IF
C*****
C***** PROPELLANT AND IGNITER INFORMATION IS ENTERED *****
C*****
      CALL CLEAR
      WRITE (*, *)
      WRITE (*, *) ' Propellant and Igniter Information'
      WRITE (*, *)
      WRITE (*, *) 'Enter the mass of the propellant in grams: '
      READ (*, *) A3 (25)
      A3 (46) = A3 (25)/A3 (3)
      CALL CLEAR
      WRITE (*, *)
      WRITE (*, *) 'Enter the mass of the igniter in grams: '
      READ (*, *) A3 (26)
      CALL CLEAR
      WRITE (*, *)
      WRITE (*, *) 'Enter the initial temperature of'
      WRITE (*, *) 'the propellant in degrees Kelvin.'
      WRITE (*, *) 'Several common temperatures are:'
      WRITE (*, *) ' -25 F = 241 K'
      WRITE (*, *) ' -20 F = 244 K'
      WRITE (*, *) ' 70 F = 294 K'
      WRITE (*, *) ' 120 F = 322 K'
      WRITE (*, *)
      WRITE (*, *) 'Please Enter Your Choice: '
      READ (*, *) A3 (27)
      WRITE (*, *)
      WRITE (*, *) 'Enter the initial temperature of'
      WRITE (*, *) 'the igniter in degrees Kelvin. '
      WRITE (*, *)
      WRITE (*, *) 'Please Enter Your Choice: '
      READ (*, *) A3 (28)
C*****
C***** SPECIAL INFORMATION FOR THE *****
C***** REDUCTIONS IS OBTAINED *****
C*****
      IF ((ITYPE .EQ. 2) .OR. (ITYPE .EQ. 3)) THEN
        CALL CLEAR
        WRITE (*, *) 'Enter the method by which the burn rate'
        WRITE (*, *) 'information will be entered.'
        WRITE (*, *) ' '
        WRITE (*, *) ' 1. bP^n burn rate law, one law for each layer.'
        WRITE (*, *) ' 2. Table of pressure versus rate.'

```

```

WRITE (*, *) ' '
1130  CONTINUE
WRITE (*, *) 'Enter your choice by number (1-2).'
WRITE (*, *) ' '
READ (*, *) ICHOICE
IF (ICHOICE .GT. 2) THEN
    WRITE (*, *) 'Your choice is not a 1 or a 2. Please select again.'
    GO TO 1130
END IF
IF (ICHOICE .EQ. 1) THEN
    CALL CLEAR
    N = INT (A3 (4) + .5)
    WRITE (*, 6170) N
6170  FORMAT(///,10X,'The grain has ',I2,' layers. One burn rate per'
1/,10X,'layer is required. '//)
    A3 (5) = - 1*N
    NEND = 48 + 2*N
    JNUM = 1
    DO 1140 J = 50, NEND, 2
        WRITE (*, 6180) JNUM
6180  FORMAT(' ', 'Enter the coefficient of the burn rate law; layer: ', I2)
        WRITE (*, *) ' '
        READ (*, *) A3 (J)
        WRITE (*, 6190) JNUM
6190  FORMAT(' ', 'Enter the exponent of the burn rate law layer: ', I2)
        WRITE (*, *) ' '
        READ (*, *) A3 (J + 1)
        JNUM = JNUM + 1
1140  CONTINUE
    END IF
    IF (ICHOICE .EQ. 2) THEN
        CALL CLEAR
        WRITE (*, *) 'How many points will be used '
        WRITE (*, *) 'to describe the burning rate?'
        WRITE (*, *) 'Maximum of 20 points.'
        WRITE (*, *) ' '
        READ (*, *) NP
        A3 (5) = NP
        DO 1150 I = 1, NP
            WRITE (*, 6200) I
6200  FORMAT(5X,'Enter the pressure and rate for point ',I2,', ',
1/,5X,'both on the same line.')
            JJ = 49 + I
            KK = 79 + I
            READ (*, *) A3 (JJ), A3 (KK)
1150  CONTINUE

```

```

END IF
END IF
C*****
C***** PREDEFINED BURN RATE RANGES ARE DEFINED *****
C*****
      IF ((ITYPE.EQ. 1) .OR. (ITYPE.EQ. 4) .OR. (ITYPE.EQ. 5)) THEN
        CALL CLEAR
        WRITE (*, 6210) PMAX
6210  FORMAT(/,10X,'One output of the burn rate analysis is a series',
          1/,10X,'of bP^n burn rate laws. The program has the option to',
          2/,10X,'determine the burn rate laws for any given pressure range',
          3/,10X,'up to the maximum pressure for the firing: ',F8.0)
        OPEN (UNIT = 7, FILE = A1 (18))
        WRITE (*, 6220)
6220  FORMAT(/,10X,'Three preset ranges can be used:'
          1/,30X,'5 - 10%'//,30X,'10 - 25%'//,30X,'25 - 75%',
          2//,10X,'1. Use predetermined ranges',
          3/,10X,'2. Enter new ranges.'//,10X,'Enter your choice.')
        READ (*, *) IIC
        DUMMY = 0.0
        IF (IIC.EQ. 1) THEN
          N = 3
          WRITE (7, *) N
          WRITE (7, *) .05*PMAX
          WRITE (7, *) .1*PMAX
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
          WRITE (7, *) .1*PMAX
          WRITE (7, *) .25*PMAX
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
          WRITE (7, *) .25*PMAX
          WRITE (7, *) .75*PMAX
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
          WRITE (7, *) DUMMY
        ELSE
          CALL CLEAR
          WRITE (*, *) 'Enter the number of ranges'
          READ (*, *) N
          WRITE (7, *) N
          DO 1160 III = 1, N
            WRITE (*, *) 'Enter pressures in MPa for range: ', III
            READ (*, *) X, Y
            WRITE (7, *) X
            WRITE (7, *) Y
          END DO
        END IF
      END IF

```

```

        WRITE (7, *) DUMMY
        WRITE (7, *) DUMMY
        WRITE (7, *) DUMMY
1160    CONTINUE
        END IF
        CLOSE (UNIT = 7)
        END IF
C*****
C***** INFORMATION FILE IS WRITTEN *****
C*****
1170 CONTINUE
        DO 1180 I = 1, 6
            WRITE (13, 6230) A2 (I)
6230    FORMAT(A80)
1180 CONTINUE
        DO 1190 I = 1, 20
            WRITE (13, 6240) A1 (I)
6240    FORMAT(A20)
1190 CONTINUE
        DO 1200 I = 1, 100
            WRITE (13, *) A3 (I)
1200 CONTINUE
        DO 1210 I = 1, 11
            DO 1220 J = 1, 15
                DO 1230 K = 1, 5
                    WRITE (13, *) P (I, J, K)
1230    CONTINUE
1220    CONTINUE
1210 CONTINUE
        CLOSE (UNIT = 13)
        END
C*****
C***** SUBROUTINE CLEAR *****
C*****
        SUBROUTINE CLEAR
        CHARACTER ST*4
        DATA ST/' [2J'/
        WRITE (*, 6000) ST
6000    FORMAT (1X,A4)
        RETURN
        END
C*****
C***** SUBROUTINE MKNAME *****
C*****
        SUBROUTINE MKNAME (A1)
        CHARACTER*20 A1 (20), A*1, NAME1
        NAME1 = A1 (5)
1000 CONTINUE

```

```

DO 1010 I = 1, 17
  A = NAME1 (I:I)
  IF (A .EQ. '.') THEN
    CALL CLEAR
    WRITE (*, *) 'The file name entered has an extension which is'
    WRITE (*, *) 'not allowed. Please enter a new name without '
    WRITE (*, *) 'an extension.'
    READ (*, 5000) NAME1
5000 FORMAT(A20)
    GO TO 1000
  END IF
  IF (A .EQ. ' ') THEN
    NAME1 (I:I + 3) = '.pvt'
    A1 (5) = NAME1
    NAME1 (I:I + 3) = '.inf'
    A1 (4) = NAME1
    NAME1 (I:I + 3) = '.pdt'
    A1 (14) = NAME1
    NAME1 (I:I + 3) = '.out'
    A1 (10) = NAME1
    NAME1 (I:I + 3) = '.dat'
    A1 (17) = NAME1
    NAME1 (I:I + 3) = '.br'
    A1 (18) = NAME1
    GO TO 1020
  END IF
1010 CONTINUE
  CALL CLEAR
  WRITE (*, *) 'FILE NAME TOO LONG TO CREATE OTHER FILES'
  PAUSE
  STOP
1020 CONTINUE
  RETURN

```

APPENDIX J:
LISTING - PROGRAM MKSMOOTH.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKSMOOTH

```

C *****
C*****
C Version: 3.0, January 1992; Last Modified 1/7/92
C                               1/9/92
C                               2/10/92; GRAPHICS ADDED
C
C Written by: William Oberle, U.S. Army Research Laboratory
C
C Purpose: This program prepares data by removing wild points, smoothing
C          the data, calculating the derivative.
C *****
C*****
COMMON /CNTROL/ ILENGT, NS, NE
COMMON /GRAPH/ START1, START2, DELTA1, DELTA2
CHARACTER*20 FPVT, FPD, FINF, A1 (20), A2 (6)*80, NAME1, A*1
CHARACTER*20 XAXISL, YAXISL, TITLE*40
DIMENSION T (2048), PR (2048), PS (2048), PDOT
1 (2048), A3 (100), P (11, 15, 5)
2 , EE (2048), ORGP (2048), ORGP1 (2048)
C*****
C***** OBTAIN PRINTER & SCREEN CHARACTERISTICS *****
C*****
CALL CLEAR
WRITE (*, 6000)
6000 FORMAT(/,10X,'This program has the capability of generating',
1/,10X,'hard copy plots of the pressure-burn rate data. To',
2/,10X,'generate plots information concerning your printer is',
3/,10X,'required. The output port is assumed to be LPT1.',
4/,10X,'Select printer by number.',
5/,15X,'1. Epson Dot Matrix',
6/,15X,'2. HP Laser Jet ',//,10X,'Enter Choice.')
```

```

READ (*, *) NAIS
IF (NAIS .EQ. 1) THEN
MODEL = 5
ELSE
MODEL = 62
END IF
C*****
C***** CONTROL FOR DOING ANOTHER SMOOTHING *****
C*****
IPLCH = 0
C*****
C***** .INF FILE IS REQUESTED *****
C*****
CALL CLEAR
1000 CONTINUE

```

```

WRITE (*, 6010)
6010 FORMAT(////,20X,'Enter file name for data to be reduced.'//)
READ (*, 5000) FINF
NAME1 = FINF
1010 CONTINUE
DO 1020 I = 1, 17
  A = NAME1 (I:I)
  IF (A .EQ. '.') THEN
    CALL CLEAR
    WRITE (*, *) 'The file name entered has an extension which is'
    WRITE (*, *) 'not allowed. Please enter a new name without '
    WRITE (*, *) 'an extension.'
    READ (*, 5000) NAME1
    GO TO 1010
  END IF
  IF (A .EQ. ' ') THEN
    NAME1 (I:I + 3) = '.inf'
    FINF = NAME1
    GO TO 1030
  END IF
1020 CONTINUE
5000 FORMAT(A20)
1030 CONTINUE
  OPEN (UNIT = 9, FILE = FINF, STATUS = 'OLD', ERR = 1040)
  REWIND (UNIT = 9)
C*****
C***** READING .INF FILE *****
C*****
  DO 1050 I = 1, 6
    READ (9, 5010) A2 (I)
1050 CONTINUE
5010 FORMAT(A80)
  DO 1060 I = 1, 20
    READ (9, 5020) A1 (I)
1060 CONTINUE
5020 FORMAT(A20)
  DO 1070 I = 1, 100
    READ (9, *) A3 (I)
1070 CONTINUE
  DO 1080 I = 1, 11
    DO 1090 J = 1, 15
      DO 1100 K = 1, 5
        READ (9, *) P (I, J, K)
1100 CONTINUE
1090 CONTINUE
1080 CONTINUE

```

```

        CLOSE (UNIT = 9)
        GO TO 1110
1040 CONTINUE
        WRITE (*, 6020) FINF
6020  FORMAT(/,10X,'An error has occurred in opening the file,',A20,
        1/,10X,'most likely the file does not exist.')
        PAUSE
        GO TO 1000
C*****
C***** SMOOTHING OPTIONS ARE LISTED *****
C*****
1110 CONTINUE
        CALL CLEAR
        WRITE (*, 6030)
6030  FORMAT(///,15X,' Smoothing & Differentiation Options',
        1/,5X,'1. Use a fixed bridge length (slivering not considered)',/,
        25X,'2. Use a "floating" bridge length (slivering not considered)'
        3/,5X,'3. Use a fixed bridge length (slivering considered)'
        4/,5X,'4. Use a floating bridge length (slivering considered)'
        5/,5X,'5. Exit'
        6//,15X,'Please Enter Your Choice: ')
        READ (*, *) ICHOICE
        IF (ICHOICE .EQ. 5) GO TO 1120
        IF (ICHOICE .GT. 5) THEN
            WRITE (*, *) 'Your choice is not between a 1 and 5'
            PAUSE
            GO TO 1110
        END IF
C*****
C***** CONTROLS ON HANDLING DATA ARE ENTERED *****
C*****
        CALL CLEAR
        WRITE (*, *) ' '
        WRITE (*, *) 'Enter the number of wild point removal passes: '
        READ (*, *) NUMW
        IF (NUMW .GT. 0) THEN
            WRITE (*, *) 'This program allows the user to enter a value'
            WRITE (*, *) 'to determine what the cutoff will be for'
            WRITE (*, *) 'discarding or keeping wildpoints or outliers.'
            WRITE (*, *) 'The value entered can be any number greater than'
            WRITE (*, *) 'zero. The closer to zero, the tighter the'
            WRITE (*, *) 'tolerance on the wildpoints.'
            WRITE (*, *) ' '
1130  CONTINUE
            WRITE (*, *) 'Enter the value for the tolerance on the wildpoints.'
            WRITE (*, *) 'A value of 5 is generally used.'
            READ (*, *) XWP

```

```

      IF (XWP .LE. 0.0) THEN
        WRITE (*, *) 'VALUE OUT OF RANGE, MUST BE LARGER THAN ZERO'
        PAUSE
        GO TO 1130
      END IF
    END IF
  WRITE (*, *) ' '
  WRITE (*, *) 'Enter the number of smoothing passes: '
  READ (*, *) NUMS
  IF (NUMS .LE. 0) GO TO 1140
  ILENGT = - 1
  IF ((ICHOICE .EQ. 1) .OR. (ICHOICE .EQ. 3)) THEN
1150  CONTINUE
    WRITE (*, *) ' '
    WRITE (*, *) 'Enter the fixed bridge length, a odd number'
    WRITE (*, *) 'between 5 and 35.'
    READ (*, *) ILENGT
C*****
C***** TEST TO DETERMINE IF ILENGT IS OK *****
C*****
    IFHRY = MOD (ILENGT, 2)
    IF ((IFHRY .EQ. 0) .OR. (ILENGT .GT. 35) .OR. (ILENGT .LT. 5)) THEN
      WRITE (*, *) 'The bridge length is not an odd number between'
      WRITE (*, *) '5 and 35. Please enter a new number for the bridge'
      WRITE (*, *) 'length.'
      GO TO 1150
    END IF
  END IF
  IF (ICHOICE .GE. 3) THEN
    CALL CLEAR
    WRITE (*, 6040)
6040  FORMAT(///,20X,'Please enter the slivering pressure in MPa.')
    READ (*, *) PCUT
  END IF
C*****
C***** SWITCH TO SKIP IF REDOING SMOOTHING *****
C*****
    IF (IPLCH .EQ. 1) GO TO 1160
C*****
C***** THE .PVT FILE IS READ *****
C*****
1140  CONTINUE
    FPVT = A1 (5)
    OPEN (UNIT = 15, FILE = FPVT)
    REWIND (15)

```

```

READ (15, *) NP
IF (NP .GT. 2048) THEN
  CALL CLEAR
  WRITE (*, *) 'THERE ARE TOO MANY POINTS IN THE P/T DATA FILE'
  WRITE (*, *) 'A MAXIMUM OF 2048 POINTS ALLOWED'
  PAUSE
  STOP
END IF
READ (15, *) PMAX
A3 (43) = PMAX
READ (15, *) PMIN
READ (15, *) TMAX
READ (15, *) TMIN
DELTAT = (TMAX - TMIN)/(NP - 1)
C*****
C***** OBTAINING CONSTANTS FOR GRAPHS *****
C*****
  T (1) = 0.0
  T (2) = TMAX
  CALL SCALE (T, 5.00, 2, 1)
  START1 = T (3)
  DELTA1 = T (4)
  T (1) = 0.0
  T (2) = PMAX
  CALL SCALE (T, 5.00, 2, 1)
  START2 = T (3)
  DELTA2 = T (4)
C*****
C***** TIME STEP IS CONFIRMED *****
C*****
  CALL CLEAR
  WRITE (*, 6050) A3 (42)
6050 FORMAT(///,20X,'The current time step is:',F10.6,' ms',
1//,20X,'Is this value correct? (Yes=1, No=2)',
2//,20X,'Enter your choice.')
  READ (*, *) III
  IF (III .EQ. 2) THEN
    WRITE (*, 6060)
6060 FORMAT(/,20X,'Enter correct time step in milliseconds.')
    READ (*, *) A3 (42)
    DELTAT = A3 (42)/1000.
  END IF
C*****
C***** PRESSURE-TIME FILE IS READ *****
C*****
1170 CONTINUE
DO 1180 I = 1, NP
  READ (15, *, ERR = 1190) T (I), PR (I)

```

```

1180 CONTINUE
      CLOSE (15)
      GO TO 1200
1190 CONTINUE
      WRITE (*, 6070) I
6070 FORMAT(/,5X,'AN ERROR HAS OCCURRED IN READING THE PRESSURE',
1/,5X,'AND TIME FILE. THE ERROR OCCURRED ON THE ',I4,' STEP.',
2/,5X,'CHECK THE DATA FILE AND TRY AGAIN.')
```

PAUSE
STOP

```

1200 CONTINUE
C*****
C*** THE ELECTRICAL ENERGY IS READ IF OPTION 5, ETC HAS BEEN *****
C***** SELECTED *****
C*****
      ITYPE = INT (A3 (2) + .5)
      IF (ITYPE .EQ. 5) THEN
        OPEN (UNIT = 15, FILE = A1 (19))
        DO 1210 I = 1, NP
          READ (15, *) DUMTT, DUMEE
          EE (I) = DUMEE
1210 CONTINUE
        END IF
C*****
C***** GRAPH OF PRESSURE TIME IS ALLOWED *****
C*****
      DO 1220 I = 1, 2048
        ORGP1 (I) = PR (I)
1220 CONTINUE
      CALL CLEAR
      WRITE (*, *) '1. Plot original pressure-time data'
      WRITE (*, *) '2. No plot'
      WRITE (*, *) ' '
      WRITE (*, *) 'Enter your choice (1 - 2)'
      READ (*, *) IYR
      IF (IYR .EQ. 2) GO TO 1230
      XAXISL = 'Time (Sec)'
      YAXISL = 'Pressure (MPa)'
      TITLE = 'Press Vs Time; File: '//A1 (5)
      CALL PL88LG (T, ORGP1, NP, XAXISL, YAXISL, TITLE, MODEL)
1230 CONTINUE
C*****
C*** THE INITIAL PRESSURE TO KEEP IS DETERMINED, MAX IGNITER PRESSURE **
C***** AND THEORETICAL MAX PRESSURE IS DETERMINED *****
C*****
      CALL PIG (PIGN, A3, P)
C*****
C***** FIRST WILDPPOINTS ARE REMOVED *****
```

```

C*****
1160 CONTINUE
  IF (NUMW .EQ. 0) GO TO 1240
  DO 1250 I = 1, NUMW
    CALL CLEAR
    WRITE (*, 6080) I, NUMW
6080  FORMAT(//, ' ', 'PROCESSING: Wildpoint Pass ', I2, ' of ', I2, ' ')
    CALL CBWILDPT (PR, NP, NUMW, XWP)
1250 CONTINUE
1260 CONTINUE
  CALL CLEAR
  WRITE (*, *) '1. Plot original data'
  WRITE (*, *) '2. Plot wildpointed data'
  WRITE (*, *) '3. Overplot original & wildpointed'
  WRITE (*, *) '4. No plot, proceed with program'
  WRITE (*, *) ' '
  WRITE (*, *) 'Enter your choice (1 - 4)'
  READ (*, *) IYR
  IF (IYR .EQ. 4) GO TO 1270
  XAXISL = 'Time (Sec)'
  YAXISL = 'Pressure (MPa)'
  TITLE = 'Press Vs Time; File: '//A1 (14)
C  TITLE='Pressure Versus Time'
  IF (IYR .EQ. 1) THEN
    CALL PL88LG (T, ORGP1, NP, XAXISL, YAXISL, TITLE, MODEL)
  END IF
  IF (IYR .EQ. 2) THEN
    CALL PL88LG (T, PR, NP, XAXISL, YAXISL, TITLE, MODEL)
  END IF
  IF (IYR .EQ. 3) THEN
    CALL PL88OP (T, ORGP1, PR, NP, XAXISL, YAXISL, TITLE, MODEL)
  END IF
  GO TO 1260
1270 CONTINUE
C*****
C***** PRESSURE DATA IS REDUCED *****
C*****
C***** FINDING LOCATION OF PMAX *****
C*****
1240 CONTINUE
  PMAX = PR (1)
  NHIGH = 1
  DO 1280 I = 1, NP
    IF (PR (I) .GE. PMAX) THEN
      PMAX = PR (I)
      NHIGH = I
    END IF
  END IF

```

```

1280 CONTINUE
C*****
C***** NEXT FIND LOCATION WHERE ALL POINTS ARE ABOVE *****
C***** 80% OF THE IGNITER PRESSURE *****
C*****
    NSN = 0
    DO 1290 I = 1, NHIGH
        IF (PR (I) .LT. PIGN) NSN = I + 1
1290 CONTINUE
    LOWERI = NSN
    IF (LOWERI .LE. 0) LOWERI = 1
C*****
C***** REWRITING THE ARRAY *****
C*****
    CALL CLEAR
    PIG2 = PIGN/.8
    NDIFF = NHIGH - LOWERI
    IF (NDIFF .GT. 1000) LOWERI = NHIGH - 999
    WRITE (*, 6090) PIG2
6090 FORMAT(' ', 'The pressure due to the igniter is: ', F12.3, ' MPa')
    WRITE (*, 6100) PR (LOWERI)
6100 FORMAT(' ', 'The starting pressure value is: ', F12.3, ' MPa')
    NPOINTS = LOWERI - 1
    TIMEDP = NPOINTS*A3 (42)
    WRITE (*, 6110) NPOINTS
6110 FORMAT(' ', I4, ' Points have been deleted.')
    WRITE (*, 6120) TIMEDP
6120 FORMAT(' ', 'This corresponds to a time delay of: ', F9.5, ' ms.')
    WRITE (*, 6130) LOWERI, NHIGH
6130 FORMAT(' ', 'The indices are: ', 2I6)
    PAUSE
    NP = 0
    DO 1300 I = LOWERI, NHIGH
        PR (NP + 1) = PR (I)
        ORGP (NP + 1) = PR (I)
        T (NP + 1) = T (I)
C    T(NP+1)=NP*A3(42)/1000.
        IF (ITYPE .EQ. 5) THEN
            EE (NP + 1) = EE (I)
        END IF
        NP = NP + 1
1300 CONTINUE
    IF (NUMS .LE. 0) THEN
        NE = NP
        NS = 1
    END IF
C*****
C***** CLEARING THE REMAINDER OF THE ARRAY TO ZERO *****

```

```

C*****
1310 CONTINUE
  NPP = NP + 1
  DO 1320 I = NPP, 2048
    PR (I) = 0.0
    ORGP (I) = 0.0
    IF (ITYPE .EQ. 5) THEN
      EE (I) = 0.0
    END IF
    T (I) = 0.0
  1320 CONTINUE
C*****
C***** MESSAGE FOR DATA SMOOTHING IN PROGRESS *****
C*****
1330 CONTINUE
  CALL CLEAR
  WRITE (*, 6140)
6140 FORMAT(///,10X,'*****',
1/,10X,'***** DATA IS NOW BEING PROCESSED *****',
2/,10X,'***** THIS CAN TAKE UP TO SEVERAL MINUTES *****',
3/,10X,'*****')
C*****
C***** The data is smoothed. *****
C*****
C***** FIRST OPTION 1 & 2 *****
C*****
  IF ((ICHOICE .EQ. 1) .OR. (ICHOICE .EQ. 2)) THEN
    NS = 1
    NE = NP
    CALL SMOOTH (T, PR, PS, PDOT, NUMS, DELTAT)
    GO TO 1340
  END IF
C*****
C***** NOW OPTION 3 & 4 *****
C*****
  IF ((ICHOICE .EQ. 3) .OR. (ICHOICE .EQ. 4)) THEN
    DO 1350 I = 1, NP
      IF (PR (I) .GT. PCUT) THEN
        NE = I - 1
        GO TO 1360
      END IF
    1350 CONTINUE
    NS = 1
    NE = NP
    CALL SMOOTH (T, PR, PS, PDOT, NUMS, DELTAT)
    GO TO 1340
  1360 CONTINUE
    NS = 1

```

```

        CALL SMOOTH (T, PR, PS, PDOT, NUMS, DELTAT)
        NS = NE + 1
        NE = NP
        CALL SMOOTH (T, PR, PS, PDOT, NUMS, DELTAT)
    END IF
    CALL CLEAR
    WRITE (*, *) 'The derivative has been calculated.'
C*****
C***** .INF FILE IS WRITTEN *****
C*****
1340 CONTINUE
    OPEN (UNIT = 5, FILE = FINF)
    REWIND (UNIT = 5)
    DO 1370 I = 1, 6
        WRITE (5, 6150) A2 (I)
1370 CONTINUE
    DO 1380 I = 1, 20
        WRITE (5, 6160) A1 (I)
1380 CONTINUE
    DO 1390 I = 1, 100
        WRITE (5, *) A3 (I)
1390 CONTINUE
    DO 1400 I = 1, 11
        DO 1410 J = 1, 15
            DO 1420 K = 1, 5
                WRITE (5, *) P (I, J, K)
1420 CONTINUE
1410 CONTINUE
1400 CONTINUE
6150 FORMAT(A80)
6160 FORMAT(A20)
    CLOSE (UNIT = 5)
C*****
C***** OPTION TO FORCE MONATOMIC PRESSURE *****
C*****
    IFFY = 0
C*****
C***** FIRST ARE POINTS OUT OF ORDER *****
C*****
    DO 1430 I = 1, NP - 1
        IF (PR (I) .GE. PR (I + 1)) IFFY = 1
1430 CONTINUE
C*****
C***** OPTION IS OFFERED *****
C*****
    IF (IFFY .EQ. 1) THEN
        CALL CLEAR
        WRITE (*, *) 'The pressure is not strictly increasing.'

```

```

WRITE (*, *) 'Do you wish to adjust the data so that'
WRITE (*, *) 'the pressure is strictly increasing?'
WRITE (*, *) 'Enter your choice. (Yes = 1, No = 2)'
READ (*, *) IFFY1
IF (IFFY1 .EQ. 2) GO TO 1440
C*****
C***** PRESSURE IS MADE STRICTLY INCREASING *****
C*****
DO 1450 I = NP - 1, 2, - 1
  IF (PR (I) .GT. PR (I - 1)) GO TO 1450
  DO 1460 K = 2, I - 1
    IF (PR (I) .GT. PR (I - K)) THEN
      IDX = I - K
      GO TO 1470
    END IF
  1460 CONTINUE
C***** POINT IS LOW POINT IN DATA *****
  IDX = 1
  1470 CONTINUE
  XBOT = I - IDX
  XSTEP = ABS ((PR (I) - PR (IDX))/XBOT)
  DO 1480 J = I - 1, IDX + 1, - 1
    PR (J) = PR (J + 1) - XSTEP
    IF (PR (J) .LE. PIGN) PR (J) = PIGN
  1480 CONTINUE
  1450 CONTINUE
  END IF
C*****
C***** GRAPH OF CURVES IS PRESENTED *****
C*****
  1440 CONTINUE
  CALL CLEAR
  WRITE (*, *) '1. Plot reduced data, wldpts removed, no smoothing'
  WRITE (*, *) '2. Plot reduced, smoothed data'
  WRITE (*, *) '3. Overplot 1. & 2.'
  WRITE (*, *) '4. No plot'
  WRITE (*, *) ' '
  WRITE (*, *) 'Enter your choice (1 - 4)'
  READ (*, *) IYR
  IF (IYR .EQ. 4) GO TO 1490
  XAXISL = 'Time (Sec)'
  YAXISL = 'Pressure (MPa)'
  TITLE = 'Press Vs Time; File: '//A1 (14)
C  TITLE='Pressure Versus Time'
  IF (IYR .EQ. 1) THEN
    CALL PL88LG (T, ORGP, NP, XAXISL, YAXISL, TITLE, MODEL)
  END IF
  IF (IYR .EQ. 2) THEN

```

```

      CALL PL88LG (T, PR, NP, XAXISL, YAXISL, TITLE, MODEL)
    END IF
    IF (IYR .EQ. 3) THEN
      CALL PL88OP (T, ORGP, PR, NP, XAXISL, YAXISL, TITLE, MODEL)
    END IF
C*****
C***** ANOTHER PLOT OR SMOOTHING *****
C*****
1500 CONTINUE
      CALL CLEAR
      WRITE (*, *) ' '
      WRITE (*, *) ' '
      WRITE (*, *) '1. Additional plot.'
      WRITE (*, *) '2. Wildpoints or Smoothing with different'
      WRITE (*, *) ' option or bridge length.'
      WRITE (*, *) '3. Save current smoothed data and exit. '
      WRITE (*, *) ' '
      WRITE (*, *) 'Enter your choice. [1 - 3]'
      READ (*, *) IANT
      IF (IANT .EQ. 1) GO TO 1440
      IF (IANT .EQ. 3) GO TO 1490
      IF (IANT .EQ. 2) THEN
        IPLCH = 1
        DO 1510 I = 1, 2048
          PR (I) = ORGP1 (I)
1510 CONTINUE
        GO TO 1110
      END IF
      WRITE (*, *) 'Your choice is not between 1 and 3.'
      PAUSE
      GO TO 1500
C*****
C***** The data is written to the output file. *****
C*****
1490 CONTINUE
      OPEN (UNIT = 3, FILE = A1 (14), STATUS = 'NEW', ERR = 1520)
      REWIND (3)
      IST = 1
      IF (NP .GT. 999) THEN
        IST = NP - 990
      END IF
      DO 1530 I = IST, NP
        TX = (I - IST)*A3 (42)/1000.
C      TX=T(I)
        IF (ITYPE .EQ. 5) THEN
          WRITE (3, *) TX, PR (I), PDOT (I), EE (I)
        ELSE
          WRITE (3, *) TX, PR (I), PDOT (I)

```

```

      END IF
1530 CONTINUE
      CLOSE (UNIT = 3)
      GO TO 1540
1520 CONTINUE
      CALL CLEAR
      WRITE (*, *) 'An error has occurred on opening the file to save'
      WRITE (*, *) 'the smoothed data. Most likely the file already'
      WRITE (*, *) 'exists. Enter a new file name.'
      READ (*, 5030) A1 (14)
5030 FORMAT(A20)
      GO TO 1340
1540 CONTINUE
      WRITE (*, *) ' '
      WRITE (*, *) 'This program has finished preparing the data.'
      WRITE (*, *) 'The prepared data is in the output file: ', A1 (14)
      WRITE (*, *) 'The information file is: ', FINF
      WRITE (*, *) ' '
      WRITE (*, *) 'Press Enter to Continue'
      READ (*, *)
      GO TO 1110
1120 CONTINUE
      END

```

C*****

SUBROUTINE PIG (PIGN, A3, P)

C*****

C Version: 3.0, January 1992

C

C Written by: William Oberle, U.S. Army Research Laboratory

C

C Purpose: This routine determines a value to be used as the lowest
 C pressure to be saved in the data file. It is based upon
 C 80% of the pressure due to the igniter after the pressure
 C has been adjusted for heat loss.

C

C*****

DIMENSION A3 (100), P (11, 15, 5)

C*****

C**** THE UNCORRECTED PRESSURE DUE TO THE IGNITER IS FOUND ****

C*****

N = INT (A3 (4) + .5)

SUM = 0.0

DO 1000 I = 1, N

SUM = SUM + P (8, I, 3)

1000 CONTINUE

SUM = SUM*A3 (46)

V = A3 (23) - A3 (17)*A3 (26) - SUM

A3 (29) = A3 (13)*A3 (26)/V

```

C*****
C***** THE THEORETICAL MAXIMUM PRESSURE IS DETERMINED *****
C*****
  SUMT1 = A3 (13)*A3 (26)
  SUMB1 = A3 (17)*A3 (26)
  SUMT = 0.0
  SUMB = 0.0
  DO 1010 I = 1, N
    SUMT = SUMT + P (4, I, 3)*P (2, I, 3)/P (8, I, 3)
    SUMB = SUMB + P (4, I, 3)*P (6, I, 3)/P (8, I, 3)
  1010 CONTINUE
  SUMT = SUMT*A3 (46) + SUMT1
  SUMB = A3 (23) - SUMB*A3 (46) - SUMB1
  A3 (44) = SUMT/SUMB
C*****
C***** THE LOWER LIMIT OF PRESSURE IS DETERMINED *****
C*****
  PIGN = A3 (29)*A3 (43)/A3 (44)*.8
  RETURN
  END
C*****
  SUBROUTINE CBWILDPT (PR, NP, NUMW, XWP)
C*****
C Subroutine: CBWILDPT.FOR
C
C Encoder: Kristopher Koehnen
C Applied Combustion Technology, Inc.
C P.O. Box 17885
C Orlando, FL 32714
C
C (305) 889-7337
C Summary: This subroutine is used to remove wild points from the data.
C This subroutine was modified from the original (see ref. 1)
C to be used in the development of a system to acquire, prepare
C and analyze data from the closed bomb experiments for the
C U.S. Army Research Laboratory - Advanced Ballistics Concepts
C Branch.
C
C Input: PR: array containing the pressure values
C NP: number of points in the array P
C
C Output: PR: pressure array with the wild points removed
C
C Reference 1: Domen, John K. "Closed Bomb Data Analysis and Reporting,
C (Revision of MIL-STD-286B, Methods 801.1.2 and 804)",
C Final Report, March 1986, AMCCOM AMSMC-QAH-T(D), Dover,
C NJ 07801-5001, Page 5
C

```

```

C *****
  DIMENSION PR (2048)
  DO 1000 KKL = 1, NUMW
    CN = 1./32.
    CNPOL = 1./70.
    CN2 = 1./224.
    NTEN = NP - 10
    DO 1010 I = 1, NTEN
      J = I + 5
      VA1 = PR (J - 5) + PR (J + 5)
      VA2 = PR (J - 3) + PR (J + 3)
      VA3 = PR (J - 1) + PR (J + 1)
      VB1 = PR (J - 5) - PR (J + 5)
      VB2 = PR (J - 3) - PR (J + 3)
      VB3 = PR (J - 1) - PR (J + 1)
      A = CN*( - 3.0*VA1 + 7.0*VA2 + 12.0*VA3)
      A1 = CNPOL*( - 5.0*VB1 - 3.0*VB2 - VB3)
      A2 = CN2*(5.0*VA1 - VA2 - 4.0*VA3)
      M = - 7
      YVAR = 0.0
      DO 1020 KK = 1, 11, 2
        K = KK - 6
1030      CONTINUE
        M = M + 2
        IF (M .EQ. 0) GO TO 1030
        YTEMP = A + A1*M + A2*M*M
        YABS = ABS (PR (J + K) - YTEMP)
        YVAR = YVAR + YABS
1020      CONTINUE
        YAVE = YVAR/6.
        YDIFF = ABS (PR (J) - A)
        IF (YDIFF .LT. 1.1) GO TO 1010
        IF (YDIFF .LE. XWP*YAVE) GO TO 1010
1040      CONTINUE
        PR (J) = A
1010      CONTINUE
1000 CONTINUE
      RETURN
    END
C *****
  SUBROUTINE SMOOTH (T, PR, PS, PDOT, NUMS, DELTAT)
  COMMON /CNTROL/ ILENGT, NS, NE
  DOUBLE PRECISION COFF (35)
  DIMENSION POINTS (35), T (2048), PR (2048), PS (2048), PDOT (2048)
  DIMENSION LIN (2048)
C *****
C  Version: 3.0, January 1992
C

```

```

C   Written by: William Oberle, U.S. Army Research Laboratory
C
C
C   Purpose: This subroutine will smooth closed bomb data and calculate
C   the derivative from the 3rd data point until the maximum
C   pressure is reached. This subroutine uses a second degree
C   least squares polynomial fit with a variable bridge length.
C   The bridge length is automatically determined and is equal
C   to the minimum number of points that are in a pressure window
C   about one of the data points with the window being 10% of the
C   maximum pressure. However, the range of the bridge is between
C   5 and 35. The program also uses a shrinking bridge to avoid
C   smoothing beyond Pmax. The work is based upon earlier work
C   by J. Domen and W. Lippencott.
C*****
C***** NOW THE SMOOTHING IS PERFORMED *****
C*****
      IF (NUMS.EQ. 0) GO TO 1000
      DO 1010 II = 1, NUMS
        WRITE (*, 6000) II, NUMS
      6000 FORMAT(' ', 'PROCESSING: Smoothing Pass ', I2, ' of ', I2, '.')
C*****
C***** EACH POINT IS DONE AS A WEIGHTED AVERAGE *****
C***** OR CONVOLUTION *****
C*****
C***** THE FIRST AND LAST TWO POINTS ARE NOT SMOOTHED *****
C*****
      LASTL = 0
      PS (NS) = PR (NS)
      PS (NS + 1) = PR (NS + 1)
      PS (NE - 1) = PR (NE - 1)
      PS (NE) = PR (NE)
C*****
C***** NOW THE REMAINING DATA POINTS *****
C*****
      DO 1020 I = NS + 2, NE - 2
C*****
C The pressure window (DELTP) is determined. It is 10% of the maximum
C pressure and it will be used to determine the bridge length.
C*****
      DELTP = 0.1*PR (NE)
C*****
C   First, the bridge length is determined. It's length will
C   depend upon the number of pressure points with in the
C   pressure interval DELTP. Ptop is the upper limit of the
C   interval. Pbot is the lower limit of the interval.
C*****
      IF (ILENGT .GT. 0) THEN

```

```

        LENG = ILENGT
        GO TO 1030
    END IF
C*****
C***** FLOATING BRIDGE *****
C*****
        LEN = 17
        DO 1040 J = 2, 17
            ITOP = I + J
            IF (ITOP .GT. NE) GO TO 1050
            IBOT = I - J
            IF (IBOT .LT. NS) GO TO 1050
            DELTA = PR (ITOP) - PR (IBOT)
            IF (DELTA - DELTP .GT. 0.0) THEN
                LENGTH = MAX0 (J - 1, 2)
                GO TO 1060
            END IF
1040     CONTINUE
1050     CONTINUE
        LENGTH = MAX0 (J - 1, 2)
1060     CONTINUE
        LEN = MIN0 (LENGTH, LEN)
        LENG = LENG*2 + 1
1030     CONTINUE
1070     CONTINUE
        ISTART = I - ((LENG - 1)/2)
        IF (ISTART .LE. 0) THEN
            LENG = LENG - 2
            GO TO 1070
        END IF
1080     CONTINUE
        IEND = I + ((LENG - 1)/2)
        IF (IEND .GT. NE) THEN
            LENG = LENG - 2
            GO TO 1080
        END IF
C*****
C***** THE DATA POINT IS SMOOTHED *****
C*****
C***** COEFFICIENTS ARE OBTAINED FOR CORRECT LENGTH *****
C*****
        IFLAG = 1
        LIN (I) = LENG
        IF (LENG .EQ. LASTL) GO TO 1090
        LASTL = LENG
        CALL ARRAY (IFLAG, LENG, COFF)
C*****
C***** Now the points are placed in the array Point. *****

```

```

C*****
1090    CONTINUE
      NN = 1
      DO 1100 I1 = ISTART, IEND
        POINTS (NN) = PR (I1)
        NN = NN + 1
1100    CONTINUE
C*****
C***** THE CONVOLUTION IS COMPUTED *****
C*****
      CONVOL = 0.0
      DO 1110 I2 = 1, LENG
        CONVOL = COFF (I2)*POINTS (I2) + CONVOL
1110    CONTINUE
      PS (I) = CONVOL
1020    CONTINUE
      DO 1120 I = NS, NE
        PR (I) = PS (I)
1120    CONTINUE
1010 CONTINUE
C*****
C***** THE DERIVATIVE IS COMPUTED *****
C*****
1000 CONTINUE
      IF (NUMS .EQ. 0) THEN
        DO 1130 I = NS, NE
          PS (I) = PR (I)
          LIN(I) = 5
1130    CONTINUE
        END IF
C*****
C***** dp/dt will be in units of million MPa/sec *****
C*****
C***** FIRST AND LAST TWO POINTS ARE FORWARD & BACKWARD DIFF*****
C*****
      PDOT (NS) = ((PS (NS + 1) - PS (NS))/1000.)/((T(NS + 1) - T (NS))*1000.)
      PDOT (NS + 1) = ((PS (NS + 2) - PS (NS + 1))/1000.)/((T(NS + 1) - T (NS))*1000.)
      PDOT (NE - 1) = ((PS (NE - 1) - PS (NE - 2))/1000.)/((T(NE - 1) - T (NE - 2))*1000.)
      PDOT (NE) = ((PS (NE) - PS (NE - 1))/1000.)/((T(NE) - T (NE - 1))*1000.)
      IFLAG = 2
      LASTL = 0
      DO 1140 I = NS + 2, NE - 2
C*****
C***** THE CONVOLUTION IS COMPUTED *****
C*****

```

```

      LENG = LIN (I)
      IF (LENG .NE. LASTL) THEN
        LASTL = LENG
        CALL ARRAY (IFLAG, LENG, COFF)
      END IF
      MOV = (LENG - 1)/2
      ISTART = I - MOV
      IEND = I + MOV
      NN = 1
      DO 1150 II = ISTART, IEND
        POINTS (NN) = PS (II)
        NN = NN + 1
1150  CONTINUE
      CONVOL = 0.0
      DO 1160 II = 1, LENG
        CONVOL = COFF (II)*POINTS (II) + CONVOL
1160  CONTINUE
      PDOT (I) = (CONVOL/1000.)/(DELTAT*1000.)
1140 CONTINUE
      RETURN
      END
C*****
C***** SUBROUTINE ARRAY *****
C*****
      SUBROUTINE ARRAY (IFLAG, LENGT, COFF)
      CHARACTER*7 FILENM
      DOUBLE PRECISION COFF (35)
      DO 1000 I = 1, 35
        COFF (I) = 0.0
1000 CONTINUE
      IF (IFLAG .EQ. 1) THEN
        IF (LENGT .EQ. 5) FILENM = 'S5.CFF'
        IF (LENGT .EQ. 7) FILENM = 'S7.CFF'
        IF (LENGT .EQ. 9) FILENM = 'S9.CFF'
        IF (LENGT .EQ. 11) FILENM = 'S11.CFF'
        IF (LENGT .EQ. 13) FILENM = 'S13.CFF'
        IF (LENGT .EQ. 15) FILENM = 'S15.CFF'
        IF (LENGT .EQ. 17) FILENM = 'S17.CFF'
        IF (LENGT .EQ. 19) FILENM = 'S19.CFF'
        IF (LENGT .EQ. 21) FILENM = 'S21.CFF'
        IF (LENGT .EQ. 23) FILENM = 'S23.CFF'
        IF (LENGT .EQ. 25) FILENM = 'S25.CFF'
        IF (LENGT .EQ. 27) FILENM = 'S27.CFF'
        IF (LENGT .EQ. 29) FILENM = 'S29.CFF'
        IF (LENGT .EQ. 31) FILENM = 'S31.CFF'
        IF (LENGT .EQ. 33) FILENM = 'S33.CFF'
        IF (LENGT .EQ. 35) FILENM = 'S35.CFF'
        OPEN (UNIT = 11, FILE = FILENM)

```

```

        REWIND (UNIT = 11)
        DO 1010 I = 1, LENGT
            READ (11, *) COFF (I)
1010    CONTINUE
        CLOSE (UNIT = 11)
        WRITE (*, *) 'Using file: ', FILENM
        RETURN
    END IF
    IF (IFLAG .EQ. 2) THEN
        IF (LENGT .EQ. 5) FILENM = 'D5.CFF'
        IF (LENGT .EQ. 7) FILENM = 'D7.CFF'
        IF (LENGT .EQ. 9) FILENM = 'D9.CFF'
        IF (LENGT .EQ. 11) FILENM = 'D11.CFF'
        IF (LENGT .EQ. 13) FILENM = 'D13.CFF'
        IF (LENGT .EQ. 15) FILENM = 'D15.CFF'
        IF (LENGT .EQ. 17) FILENM = 'D17.CFF'
        IF (LENGT .EQ. 19) FILENM = 'D19.CFF'
        IF (LENGT .EQ. 21) FILENM = 'D21.CFF'
        IF (LENGT .EQ. 23) FILENM = 'D23.CFF'
        IF (LENGT .EQ. 25) FILENM = 'D25.CFF'
        IF (LENGT .EQ. 27) FILENM = 'D27.CFF'
        IF (LENGT .EQ. 29) FILENM = 'D29.CFF'
        IF (LENGT .EQ. 31) FILENM = 'D31.CFF'
        IF (LENGT .EQ. 33) FILENM = 'D33.CFF'
        IF (LENGT .EQ. 35) FILENM = 'D35.CFF'
        OPEN (UNIT = 11, FILE = FILENM)
        REWIND (UNIT = 11)
        DO 1020 I = 1, LENGT
            READ (11, *) COFF (I)
1020    CONTINUE
        CLOSE (UNIT = 11)
        WRITE (*, *) 'Using file: ', FILENM
        RETURN
    END IF
    RETURN
END

C*****
C***** SUBROUTINE CLEAR *****
C*****
    SUBROUTINE CLEAR
    CHARACTER ST*4
    DATA ST/' [2J'/
    WRITE (*, 6000) ST
6000 FORMAT (1X,A4)
    RETURN
    END
C*****
    SUBROUTINE PL88LG (X, Y, NPTS, XAXISH, YAXISH, TITLEH, MODEL)

```

```

COMMON /GRAPH/ START1, START2, DELTA1, DELTA2
DIMENSION X (2050), Y (2050)
CHARACTER*20 XAXISH, YAXISH
CHARACTER*40 TITLEH
CHARACTER*20 YAXISL
CHARACTER*40 TITLE
CHARACTER*20 XAXISL
CHARACTER*1 TITA (40), XAXISA (20), YAXISA (20)
XAXISL = XAXISH
YAXISL = YAXISH
TITLE = TITLEH
DO 1000 I = 1, 20
  KI = 21 - I
  IF (XAXISA (KI) .NE. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
  NXA = KI
  DO 1020 I = 1, 20
    KI = 21 - I
    IF (YAXISA (KI) .NE. ' ') GO TO 1030
1020 CONTINUE
1030 CONTINUE
  NYA = KI
  DO 1040 I = 1, 40
    KI = 41 - I
    IF (TITA (KI) .NE. ' ') GO TO 1050
1040 CONTINUE
1050 CONTINUE
  NTA = KI
  WRITE (*, 6000)
  WRITE (*, 6010)
6010 FORMAT('O DO YOU WISH TO HAVE A HARDCOPY? ( 1 FOR YES,0 FOR NO)')
  READ (*, 5000) ICYN
5000 FORMAT(I1)
6000 FORMAT('O STRIKE ENTER WHEN FINISHED WITH PLOT')
  CALL PLOTS (0, 97, 97)
  CALL FACTOR (1.0)
  HT = .25
  CALL PLOT (2.50, 1.50, - 3)
  XT = 2.5 - HT*5./8.*.5*NTA
  CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
C   CALL SCALE(X,5.00,NPTS,1)
C   CALL SCALE(Y,5.00,NPTS,1)
  X (NPTS + 1) = START1
  X (NPTS + 2) = DELTA1
  Y (NPTS + 1) = START2
  Y (NPTS + 2) = DELTA2
  CALL STAXIS (.25, .25, .111, .112, 2)

```

```

CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X1 (NPTS + 1), X (NPTS + 2))
CALL STAXIS (.25, .25, .111, .112, - 1)
CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
CALL LINE (X, Y, NPTS, 1, 0, 0)
CALL PLOT (0.0, 0.0, 999)
IF (ICYN .EQ. 0) GO TO 1060
CALL PLOTS (0, 1, MODEL)
CALL FACTOR (1.0)
HT = .25
CALL PLOT (2.50, 1.50, - 3)
XT = 2.5 - HT*5./8.*.5*NTA
CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
C   CALL SCALE(X,5.00,NPTS,1)
C   CALL SCALE(Y,5.00,NPTS,1)
CALL STAXIS (.25, .25, .111, .112, 2)
CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
CALL STAXIS (.25, .25, .111, .112, - 1)
CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
CALL LINE (X, Y, NPTS, 1, 0, 0)
CALL PLOT (0.0, 0.0, 999)
1060 CONTINUE
RETURN
END
C*****
SUBROUTINE PL88OP (X, Y, Z, NPTS, XAXISH, YAXISH, TITLEH, MODEL)
COMMON /GRAPH/ START1, START2, DELTA1, DELTA2
DIMENSION X (2050), Y (2050), Z (2050)
CHARACTER*20 XAXISH, YAXISH
CHARACTER*40 TITLEH
CHARACTER*20 YAXISL
CHARACTER*40 TITLE
CHARACTER*20 XAXISL
CHARACTER*1 TITA (40), XAXISA (20), YAXISA (20)
XAXISL = XAXISH
YAXISL = YAXISH
TITLE = TITLEH
DO 1000 I = 1, 20
  KI = 21 - I
  IF (XAXISA (KI) .NE. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
  NXA = KI
  DO 1020 I = 1, 20
    KI = 21 - I
    IF (YAXISA (KI) .NE. ' ') GO TO 1030

```

```

1020 CONTINUE
1030 CONTINUE
    NYA = KI
    DO 1040 I = 1, 40
        KI = 41 - I
        IF (TITA (KI) .NE. ' ') GO TO 1050
1040 CONTINUE
1050 CONTINUE
    NTA = KI
    WRITE (*, 6000)
    WRITE (*, 6010)
6010 FORMAT('O DO YOU WISH TO HAVE A HARDCOPY? (1 FOR YES, 0 FOR NO)')
    READ (*, 5000) ICYN
5000 FORMAT(I1)
6000 FORMAT('O STRIKE ENTER WHEN FINISHED WITH PLOT')
    CALL PLOTS (0, 97, 97)
    CALL FACTOR (1.0)
    HT = .25
    CALL PLOT (2.50, 1.50, - 3)
    XT = 2.5 - HT*5./8.*.5*NTA
    CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
C    CALL SCALE(X,5.00,NPTS,1)
C    CALL SCALE(Y,5.00,NPTS,1)
C    CALL SCALE(Z,5.00,NPTS,1)
    X (NPTS + 1) = START1
    X (NPTS + 2) = DELTA1
    Y (NPTS + 1) = START2
    Y (NPTS + 2) = DELTA2
    Z (NPTS + 1) = START2
    Z (NPTS + 2) = DELTA2
    CALL STAXIS (.25, .25, .111, .112, 2)
    CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
    CALL STAXIS (.25, .25, .111, .112, - 1)
    CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
    CALL LINE (X, Y, NPTS, 1, 0, 0)
    CALL LINE (X, Z, NPTS, 1, 0, 0)
    CALL PLOT (0.0, 0.0, 999)
    IF (ICYN .EQ. 0) GO TO 1060
    CALL PLOTS (0, 1, MODEL)
    CALL FACTOR (1.0)
    HT = .25
    CALL PLOT (2.50, 1.50, - 3)
    XT = 2.5 - HT*5./8.*.5*NTA
    CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
C    CALL SCALE(X,5.00,NPTS,1)
C    CALL SCALE(Y,5.00,NPTS,1)

```

```
C  CALL SCALE(Z,5.00,NPTS,1)
    CALL STAXIS (.25, .25, .111, .112, 2)
    CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
    CALL STAXIS (.25, .25, .111, .112, - 1)
    CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
    CALL LINE (X, Y, NPTS, 1, 0, 0)
    CALL LINE (X, Z, NPTS, 1, 0, 0)
    CALL PLOT (0.0, 0.0, 999)
1060 CONTINUE
    RETURN
    END
```

APPENDIX K:
LISTING - PROGRAM MKCAL.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKCAL

```

C
C*****
C*****
C** "CLBCPV" = CLosed-Bomb Program with Continuous Property Variations
C*****
C**          2 December 1991 Version (with Oberle integrals) DEK
C*****
C*  MKCAL: VERSION 3.0, CLBCPV + ADDED ROUTINES  [30 Dec 91 / Kooker]
C****
C          Latest Revisions: 2/10/92; completed ETC option
C                                hex grain completed
C                                2/12/92; Newton-Raphson added
C                                2/22/92; Cleanup
C*****
C*****
COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /ETC/ EE (1000)
COMMON /CONTRL/ NTL, CHVOL, HLFRAC, DT, NMAX, CONVRG, ICTYP
COMMON /FIXED/ CH1, CH2, CH3, CAPEO, QWMAX, PCHAMX
COMMON /SUMS/ SH1, SH2, SH3, SH4
COMMON /NVALUE/ RHOSRN, ASURN, DPTHBN, TPMRGM
COMMON /SOLUT/ TIME (1000), PCH (1000), RBR1 (1000), TPMR (1000), SMS (1000), RHOS
1 (1000), TCH (1000), DPTHB (1000), ASUR (1000), LYR (1000)
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
COMMON /SOLIDL/ SMSOL (15), XI (15)
COMMON /AVEPROP/ RHOSAV (15), SESOAV (15), SBCGAV (15), RGCGAV (15),
1 CVCGAV (15)
COMMON /IGNITOR/ SMSIG, SESIG, RHOSIG, WTMIG, SBIG, RGIG, CVIG
COMMON /AIR/ SMSA, WTMA, SBA, RGA, CVA
COMMON /METH/ IMOD
CHARACTER A1 (20)*20, A2 (6)*80, FINF*20, NAME1*20, A*1
C*****
C*****
CONVRG = .00001
IMOD = 1
IQRE = 0
1000 CONTINUE
CALL CLEAR
WRITE (*, 6000)
6000 FORMAT(////,10X,'Enter file name for the calculation.')
READ (*, 5000) FINF
5000 FORMAT(A20)

```

```

IF (FINF.EQ. '10') THEN
  CALL CLEAR
  WRITE (*, *) 'ENTER THE CONVERGENCE CRITERIA'
  WRITE (*, *) 'THE DEFAULT VALUE IS 1.E-5'
  READ (*, *) CONVRG
  GO TO 1000
END IF
IF (FINF.EQ. '11') THEN
  IQRE = 100
  CALL CLEAR
C*****
C***** DEFAULT IS INTEGRALS *****
C*****
  WRITE (*, *) 'Do you wish to use the integrals instead of'
  WRITE (*, *) 'the fixed values? (Yes = 1, No = 2) Enter your choice'
  READ (*, *) ICONVR
  IF (ICONVR.EQ. 2) THEN
    WRITE (*, *) 'Enter a 1 if the grain has only one larger, else a 2.'
    READ(*,*)ISRS
  ELSE
    ISRS = 0.0
  END IF
  GO TO 1000
END IF
IF (FINF.EQ. '12') THEN
  CALL CLEAR
  WRITE (*, *) 'For homogeneous grains the search for the depth'
  WRITE (*, *) 'burned can be performed using two methods.'
  WRITE (*, *) ' '
  WRITE (*, *) '1. Bisection method: will always converge but slower'
  WRITE (*, *) '2. Newton-Raphson method: faster than bisection'
  WRITE (*, *) ' '
  WRITE (*, *) 'Select desired method'
  READ (*, *) IMOD
  GO TO 1000
END IF
NAME1 = FINF
1010 CONTINUE
DO 1020 I = 1, 17
  A = NAME1 (I:I)
  IF (A.EQ. '.') THEN
    CALL CLEAR
    WRITE (*, *) 'The file name entered has an extension which is'
    WRITE (*, *) 'not allowed. Please enter a new name without '
    WRITE (*, *) 'an extension.'

```

```

        READ (*, 5000) NAME1
        GO TO 1010
    END IF
    IF (A .EQ. ' ') THEN
        NAME1 (I:I + 3) = '.inf'
        FINF = NAME1
        GO TO 1030
    END IF
1020 CONTINUE
1030 CONTINUE
    OPEN (UNIT = 12, FILE = FINF, STATUS = 'OLD', ERR = 1040)
    GO TO 1050
1040 CONTINUE
    WRITE (*, 6010)
6010 FORMAT(//,10X,'An Error Has Occurred in Opening the File!'
    1/,10X,'The File Most Likely Does Not Exist. Try Again')
    PAUSE
    GO TO 1000
1050 CONTINUE
    REWIND (UNIT = 12)
C*****
C***** READING INFORMATION FILE FOR COMPUTATION *****
C*****
    DO 1060 I = 1, 6
        READ (12, 5010) A2 (I)
1060 CONTINUE
5010 FORMAT(A80)
    DO 1070 I = 1, 20
        READ (12, 5000) A1 (I)
1070 CONTINUE
    DO 1080 I = 1, 100
        READ (12, *) A3 (I)
1080 CONTINUE
    DO 1090 I = 1, 11
        DO 1100 J = 1, 15
            DO 1110 K = 1, 5
                READ (12, *) PX (I, J, K)
1110 CONTINUE
1100 CONTINUE
1090 CONTINUE
    CLOSE (UNIT = 12)
    A3 (100) = 0.0
    IF (IQRE .EQ. 100) THEN
        A3 (31) = ISRS
    END IF
C*****
C***** OUTPUT FILE IS OPENED *****
C*****

```

```

OPEN (UNIT = 11, FILE = A1 (10), STATUS = 'NEW')
C
C*****
C*****
C**** 7 10      20      30      40      50      60      70 2
C ***
C*****
C*****
C***** Define Reference Quantities for Non-Dimensional Scaling ***
C*****
C*****
C*****
C***** RHOSTR (g/cm3) = reference value of DENSITY
C**
      RHOSTR = 1.0
C**
C***** VOLSTR (cm3) = reference value of VOLUME (usually = chamber volu
Cme)
C**
      VOLSTR = 300.0
C**
C***** ELSTR (cm) = reference value of LENGTH
C**
      ELSTR = VOLSTR**(1./3.)
C**
C***** SURSTR (cm2) = reference value of SURFACE AREA
C**
      SURSTR = ELSTR*ELSTR
C**
C***** TRSTR (deg K) = reference value of TEMPERATURE
C**
      TRSTR = 3000.0
C**
C***** PRSTR (MPa) = reference value of PRESSURE
C**
      PRSTR = 100.0
C**
C*****
C*****
C***** Form Scale Factors *****
C*****
C*****
C*****
C***** SMSSCL (gm) = scale factor for MASS
C**
      SMSSCL = RHOSTR*VOLSTR
C**
C***** SERSCL (cal/g) = scale factor for ENERGY DENSITY

```

```

C**
  SERSCL = PRSTR/(4.184*RHOSTR)
C**
C***** CVRSCL (cal/g-deg K) = scale factor for SPECIFIC HEAT
C**
  CVRSCL = SERSCL/TRSTR
C**
C***** RBRSTR (cm/s) = reference value for RATE
C**
  RBRSTR = SQRT (1.0E+07*PRSTR/RHOSTR)
C**
C***** TIMSCL (mil-sec) = scale factor for TIME
C**
  TIMSCL = (ELSTR/RBRSTR)*1000.0
C*****
C*****
C** IMPORTANT NOTE: It is assumed here that all dimensional "PX" quantit
Cies *
C** have been loaded into the array at this point
C *
C*****
C*****
  CALL SETVAL
C*****
C*****
C***** Set Average Properties Across Each Layer *****
C*****
C*****
C*****
  DO 1120 I = 1, NTL
    RHOSAV (I) = (PX (4, I, 3)/PX (8, I, 3))/RHOSTR
    SESOAV (I) = (PX (11, I, 3)/PX (8, I, 3))/SERSCL
    SBCGAV (I) = (PX (6, I, 3)/PX (8, I, 3))*RHOSTR
    RGCNAV (I) = (PX (9, I, 3)/PX (8, I, 3))/CVRSCL
    CVCNAV (I) = (PX (10, I, 3)/PX (8, I, 3))/CVRSCL
  1120 CONTINUE
C*****
C*****
C***** Form Constants *****
C*****
C*****
C**
  CH1 = CHVOL - SBIG*SMSIG - SBA*SMSA
C**
  CH2 = RGIG*SMSIG + RGA*SMSA
C**
  CH3 = CVIG*SMSIG + CVA*SMSA

```

```

C**
C*****
C*****
C***** Compute Total Propellant Mass in Chamber at Initial Condition ***
C*****
C***** (Does NOT Include Ignitor Mass) *****
C*****
C*****
C*****
SUMI = 0.0
DO 1130 I = 1, NTL
    SUMI = SUMI + SMSOL (I)
1130 CONTINUE
C
    TPMR (I) = SUMI
C*****
C*****
C***** Compute Total Energy in Chamber at Initial Condition *****
C*****
C***** (Includes Ignitor Energy) *****
C*****
C*****
C*****
SUMI = 0.0
DO 1140 I = 1, NTL
    SUMI = SUMI + SESOAV (I)*SMSOL (I)
1140 CONTINUE
C
    CAPEO = SUMI + SESIG*SMSIG
C*****
C*****
C***** COMPUTE MAXIMUM HEAT LOSS FROM CHAMBER Based on "HLFRAC" *****
C*****
C*****
C*****
C***** ASSUME ALL PROPELLANT IS CONSUMED *****
C*****
C***** Set Value of "Xi" = 0.0 for Each Layer (burned) *****
C*****
C***** [Xi=(SMS/SMSO)i] *****
C*****
C*****
DO 1150 I = 1, NTL
    XI (I) = 0.0
1150 CONTINUE
C*****
C*****

```

```

C***** Compute Sums Which Remain Constant in Layer "J" (=NTL) *****
C*****
C**** CALL SHEVAL (J,NTL) [SH1,SH2,SH3,SH4 returned in COMMON/SUMS] *
C*****
C*****
C*****
C
      CALL SHEVAL (NTL, NTL)
C
C*****
C*****
C** Set average property values to the last "n-th" layer which will be "
CNTL"
C
      AMCVCG = SMSOL (NTL)*CVC GAV (NTL)
      AMSBCG = SMSOL (NTL)*SBC GAV (NTL)
      AMRGCG = SMSOL (NTL)*RGCGAV (NTL)
C
C*** Compute theoretical maximum chamber temperature "TCHTMX"
C [the relationships below assume that XI = 0.0]
C*****
C***** ETC COMPUTATION: ADD TOTAL ELECTRICAL ENERGY *****
C***** TO HEAT LOSS FACTOR IN COMPUTING THEORETICAL *****
C***** PRESSURE & TEMPERATURE & HEAT LOSS FACTOR *****
C*****
      IF (ICTYP .EQ. 5) THEN
        QWN = - A3 (30)*1000000./(4.184*SERSCL*SMSSCL)
      ELSE
        QWN = 0.0
      END IF
      TOP = CAPEO - QWN - SH4
      BTM = CH3 + SH3 + AMCVCG
      TCHTMX = TOP/BTM
C
C*** Compute theoretical maximum chamber pressure "PCHTMX"
C
      TOP = TCHTMX*(CH2 + SH2 + AMRGCG)
      BTM = CH1 - SH1 - AMSBCG
      PCHTMX = TOP/BTM
C
C*** Compute maximum chamber pressure WITH heat loss - "PCHAMX"
C*****
C***** PCHAMX WILL BE OBSERVED MAX PRESSURE *****
C***** EXCEPT FOR OPTION 2 & 4, PRESSURE GENERATION *****
C***** ASSUMING PCHTMX IS IN SCALED UNITS *****
C*****
      CALL CLEAR
      IF ((ICTYP .EQ. 1) .OR. (ICTYP .EQ. 3) .OR. (ICTYP .EQ. 5)) THEN

```

```

      HLFRAC = 1. - A3 (43)/(PCHTMX*PRSTR)
      WRITE (*, 6020) A3 (43), PCHTMX*PRSTR, HLFRAC
6020 FORMAT(///,5X,'** Based on the following pressure information:',
1/,5X,'Observed Maximum Pressure (MPa) value of =',F20.10,
2/,5X,'Maximum Theoretical Pressure (MPa) value of =',F20.10,
3/,5X,'the current heat loss fraction is: ',F12.5,
4/,10X,'Enter a different value? (Yes=1, No = 2)')
      READ (*, *) ILOSS
      IF (ILOSS.EQ. 2) GO TO 1160
      ELSE
        HLFRAC = 0.0
        CALL CLEAR
        WRITE (*, 6030)
6030 FORMAT(///,5X,'For the pressure generation option (Opt. 2) or',
1/,5X,'the interrupted burner option (Opt. 4), a heat loss',
2/,5X,'fraction must be entered. The heat loss fraction is in',
3/,5X,'the range of 0.0 to 1.0. For example, a value of 0.1',
4/,5X,'means 10% of the total energy will be considered lost',
5/,5X,'as heat to the chamber wall during the calculation.')
      END IF
      WRITE (*, 6040)
6040 FORMAT(///,5X,'Enter the value for the heat loss factor.'
1/,5X,'A decimal between 0.0 and 1.0')
      READ (*, *) HLFRAC
1160 CONTINUE
      PCHAMX = (1.0 - HLFRAC)*PCHTMX
C
C*** Compute maximum chamber temperature WITH heat loss - "TCHAMX"
C
      TCHAMX = PCHAMX*BTM/(CH2 + SH2 + AMRGCG)
C
C*** Solve for maximum cumulative heat loss "QWMAX"
C
      BTM = CH3 + SH3 + AMCVCG
C***** TOTAL HEAT LOSS ADJUSTED FOR ELECTRICAL ENERGY *****
      IF (ICTYP.EQ. 5) THEN
        QWMAX = CAPEO + A3 (30)*1000000./(4.184*SERSCL*SMSSCL) - BTM*TCHAMX
      ELSE
        QWMAX = CAPEO - BTM*TCHAMX
      END IF
C***** LINES ABOVE ADDED TO HANDLE THE ETC CASE *****
C ***** LINE BELOW IS THE ORIGINAL CODING *****
C      QWMAX = CAPEO - BTM*TCHAMX
C
C*****
C***** Compute Dimensional Heat Loss Quantities *****

```

```

C*****
C*****
C*****
C
    TMXP = TRSTR*TCHTMX
    TAMXP = TRSTR*TCHAMX
    PMXP = PRSTR*PCHTMX
    PAMXP = PRSTR*PCHAMX
    QWMXP = SERSCL*SMSSCL*QWMAX
C
    WRITE (11, 6050) PMXP, TMXP, HLFrac, QWMXP, PAMXP, TAMXP
C
6050 FORMAT(/,5X,' ***** Maximum Chamber Properties are *****'
    1/,5X,'Theoretical Maximum Chamber Pressure (MPa) =' ,E16.8
    2/,5X,'Theoretical Maximum Chamber Temperature (deg K) =' ,E16.8
    3/,5X,' **** Assuming a Heat-Loss-Fraction =' ,F10.3
    4/,5X,'Maximum Total Heat Loss from Chamber (cal) =' ,E16.8
    5/,5X,'Maximum Chamber Pressure (MPa) w/Heat Loss =' ,E16.8
    6/,5X,'Maximum Chamber Temperature (deg K) w/Heat Loss =' ,E16.8/)
C
C*****
C*****
C***** END OF HEAT LOSS COMPUTATION *****
C*****
C*****
C*****
C***** Set Value of "Xi" = 1.0 for Each Layer (Unburned) *****
C*****
C***** [Xi=(SMS/SMO)i] *****
C*****
C*****
C*****
C
    DO 1170 I = 1, NTL
        XI (I) = 1.0
    1170 CONTINUE
C
C*****
C*****
C
C***** COMPUTATION DECISION BASED ON VALUE OF "ICTYP" *****
C*****
C
C*****
C*****
C**** COMPUTATION OF SYNTHETIC "P-T" CURVE FROM KNOWN BURNING RATES ****
C****
C*****

```

```

C*****
C
  IF (ICTYP .EQ. 2) CALL PTGEN
C
C*****
C**** COMPUTATION OF BURNING RATES FROM GIVEN CHAMBER "P-T" CURVE *****
C*****
C***** ALSO SURFACE AREA COMPUTATION *****
C*****
C*****
C*****
C
  IF (ICTYP .NE. 2) CALL BRRED (A1)
C
C*****
C***** REWRITING THE .INF FILE *****
C*****
  OPEN (UNIT = 17, FILE = FINF)
1180 CONTINUE
  DO 1190 I = 1, 6
    WRITE (17, 6060) A2 (I)
6060 FORMAT(A80)
1190 CONTINUE
  DO 1200 I = 1, 20
    WRITE (17, 6070) A1 (I)
6070 FORMAT(A20)
1200 CONTINUE
  DO 1210 I = 1, 100
    WRITE (17, *) A3 (I)
1210 CONTINUE
  DO 1220 I = 1, 11
    DO 1230 J = 1, 15
      DO 1240 K = 1, 5
        WRITE (17, *) PX (I, J, K)
1240 CONTINUE
1230 CONTINUE
1220 CONTINUE
    CLOSE (UNIT = 17)
    CLOSE (UNIT = 11)
    CLOSE (UNIT = 13)
  STOP
  END
C
C*****
C*****
C*****
C*****

```

```

C***** END OF "MAIN" CONTROL PROGRAM *****
C*****
C*****
C*****
C*****
C*****
C

```

SUBROUTINE PTGEN

```

C
C*****
C*****
C** This routine generates chamber pressure as a function of time when
C*****
C** burning rates have been specified in Subroutine BURNRT
C*****
C*****
COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /ETC/ EE (1000)
C
COMMON /CONTRL/ NTL, CHVOL, HLFRAC, DT, NMAX, CONVRG, ICTYP
COMMON /FIXED/ CH1, CH2, CH3, CAPEO, QWMAX, PCHAMX
COMMON /SUMS/ SH1, SH2, SH3, SH4
COMMON /NVALUE/ RHOSRN, ASURN, DPTHBN, TPMRGM
COMMON /SOLUT/ TIME (1000), PCH (1000), RBR
1 (1000), TPMR (1000), SMS (1000)
2 , RHOS (1000), TCH (1000), DPTHB (1000),
3 ASUR (1000), LYR (1000)
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
COMMON /SOLIDL/ SMSOL (15), XI (15)
COMMON /AVEPROP/ RHOSAV (15), SESOAV (15), SBCGAV (15), RGCGAV (15),
1 CVCGAV (15)
COMMON /IGNITOR/ SMSIG, SESIG, RHOSIG, WTMIG, SBIG, RGIG, CVIG
COMMON /AIR/ SMSA, WTMA, SBA, RGA, CVA
COMMON /METH/ IMOD
C*****
C*****
C** Begin burning initial (outside) layer: J = 1
C
J = 1
C
C*****
C*****
C***** Compute Sums Which Remain Constant in Layer "J" (=1) *****
C*****
C**** CALL SHEVAL (J,NTL) [SH1,SH2,SH3,SH4 returned in COMMON/SUMS] *

```

```

C*****
C*****
C*****
C
    CALL SHEVAL (J, NTL)
C
C*****
C*****
C***** COMPUTE INITIAL CONDITIONS IN CHAMBER WITH IGNITOR CONSUMED **
C*****
C*****
C*****
    WRITE (11, 6000)
6000 FORMAT(/, ' ** Results: Computation of Chamber'
    1 , ' Pressure from a Known Burning Rate **', //
    2 , 3x, '*** N *** LAYER *** TIME(ms) *** PCH(MPa) *** TPMR(gm)'
    3 , ' *** RBR(cm/s) *** ASUR(cm2) *** TCH(degK) *** DEPTHB(cm) ***'
    4 , /)
C
C*** COMPUTE INITIAL VALUES OF "PCH" AND "TCH" BASED ON "XIN"=1.0 ***
C
    XIN = 1.0
C
C*****
C*****
C** Call "PTCALC" Subroutine to get initial values of chamber pressure a
Cnd
C** temperature PLUS the initial surface area AND all integrals
C
    CALL PTCALC (J, XIN, PCHN, TCHN, QWN)
C
C*****
C*** COMPUTE INITIAL BURNING RATE ***
C
    CALL BURNRT (PCHN, 1, RBRN)
C
C*****
C
    PCH (1) = PCHN
    TCH (1) = TCHN
    RHOS (1) = PX (4, J, 1)/RHOSTR
C
    SMS (1) = SMSOL (J)
    RBR (1) = RBRN
    ASUR (1) = ASURN
    DPTHB (1) = DPTHBN
    LYR (1) = J
C

```

```

    TIME (1) = 0.0
C
    CALL OUTPUT (1)
C
C*****
C*****
C***** BEGIN TIME INTEGRATION TO COMPUTE CHAMBER PRESSURE (TIME) ****
C*****
C*****
C*****
C
    N = 1
C
C** Initial guess for SMSN and PCHN
C (only used to get average properties)
C
    SMSN = SMS (N)
    PCHN = PCH (N)
C
C*****
C***** "490" LOOP COMPUTES CHAMBER PRESSURE vs TIME *****
C*****
C
    1000 CONTINUE
C
    N = N + 1
C*****
C** PRINT TO TELL IF SOMETHING IS HAPPENING ** [do NOT do this on Cray]*
C***** OBERLE 12/25/91 ***** [Kooker 30 Dec 91 ]***
C*****
    CALL CLEAR
    IETY = INT (A3 (31) + .5)
    IF ((IETY .EQ. 1) .OR. (IETY .EQ. 2)) THEN
        WRITE (*, *) ' Using Constant Values'
    ELSE
        WRITE (*, *) ' Using Integrals'
    END IF
    WRITE (*, 6010) N, PCH (N - 1)*PRSTR, TIME (N - 1)*TIMSCL
6010 FORMAT(///,2X,'Pressure Generation Step: ',I4,/
    15X,'** Previous Step: P(MPa) =',F10.4,
    23X,'and Time (millisec) =',F9.3)
    IF (N .GT. NMAX) GO TO 1010
    TIME (N) = TIME (N - 1) + DT
C
C*****
C**** BEGIN ITERATION FOR VALUE OF "PCHN" AT TIME STEP "N" *****
C*****
C** Parameters ending with "--N1" are previous converged values

```

```

C   at last time step "N-1"
C** Parameters ending with "--N" are current iterative values
C   at current time step "N"
C*****
C
C** Set Last Values of RHOS, ASUR and RBR
C
  RHOSN1 = RHOS (N - 1)
  ASURN1 = ASUR (N - 1)
  RBRN1 = RBR (N - 1)
C
C** Set Last Value of Mass (in Layer "J") Remaining in Chamber
C
  SMSN1 = SMS (N - 1)
C
C***** Begin Iterative "500" Loop to Find PCH(N): *****
C**** (Loop will try 20 times for convergence, KN <= 20) *****
C
  IDONE = 0
  KN = 0
  FRACT = 1.0
C
1020 CONTINUE
C
  IF (KN .LE. 20 .AND. IDONE .NE. 1) THEN
C
    KN = KN + 1
    PCHLAS = PCHN
    CALL BURNRT (PCHN, J, RBRN)
C
    XIN = SMSN/SMSOL (J)
C
    JP1 = J + 1
    SUMI = 0.0
C
    IF (JP1 .LE. NTL) THEN
      DO 1030 I = JP1, NTL
        SUMI = SUMI + SMSOL (I)
1030    CONTINUE
      END IF
C
    TPMRN = SUMI + SMSN
    TPMRGM = TPMRN*SMSSCL
C
C*****
C** Call "DEPTH-BURNED" Subroutine to get Surface area + Integrals
C*****
C

```

```

      CALL DEPTH (TPMRGM, J, 1, DEPTHCM, ASURCM2)
C
      DPTHBN = DEPTHCM/ELSTR
      ASURN = ASURCM2/SURSTR
      RHOSN = (PX (4, J, 1) + (PX (4, J, 2) - PX (4, J, 1))*
1      ((DEPTHCM-PX (1, J, 1))/PX(1, J, 2)))/RHOSTR
C
C*****
C
      DELSMS = FRACT*DT*(RHOSN*ASURN*RBRN
1      + RHOSN1*ASURN1*RBRN1)/2.0
      SMSN = SMSN1 - DELSMS
C
C*** Must Check for SMSN < 0.0 => Layer has Burned Out
C
      IF (SMSN .GE. 0.00) THEN
C
C** Layer "J" has not yet Burned Out
C
      FRACT = 1.0
      XIN = SMSN/SMSOL (J)
C
C*** COMPUTE VALUES OF "PCHN" AND "TCHN" FOR THIS VALUE OF "XIN"
C
      CALL PTCALC (J, XIN, PCHN, TCHN, QWN)
C
      IF (ABS ((PCHN - PCHLAS)/PCHLAS) .LT. CONVRG .OR. KN .GE. 20) THEN
        PCH (N) = PCHN
        TCH (N) = TCHN
        SMS (N) = SMSN
C
        RHOS (N) = RHOSN
        RBR (N) = RBRN
        ASUR (N) = ASURN
        DPTHB (N) = DPTHBN
        TPMR (N) = TPMRN
        LYR (N) = J
C*****
C** After successful completion of step, load "5" values into "4"
C
      DO 1040 KV = 2, 11
        PX (KV, J, 4) = PX (KV, J, 5)
1040      CONTINUE
C*****
      CALL OUTPUT (N)
      IDONE = 1
C

```

```

        ELSE
        END IF
C
    ELSE
C
C*****
C***** Layer "J" has Burned Out *****
C**** Must Reset Sums and Switch Layers to J = J + 1 *****
C*****
C** COMPUTE PROPERTIES AT BURNOUT OF LAYER "J"
C
    XIN = 0.0
C
    CALL PTCALC (J, XIN, PCHBO, TCHBO, QWBO)
C
C** Print/Write BURNOUT VALUES **
C
    TIMBOP = TIME (N)*TIMSCL
    PCHBOP = PCHBO*PRSTR
    TCHBOP = TCHBO*TRSTR
    DPTHOP = DPTHBN*ELSTR
C
    WRITE (11, 6020) J, N, TIMBOP, PCHBOP, TCHBOP, DPTHOP
6020 FORMAT(/' *** PROPELLANT LAYER J= ',I2,' has Burned '
1 , 'out on Step N= ',I5,' Before TIME (millisec) of ',E16.8/
2 ,15X,'Chamber Pressure (MPa) at Burnout =',E16.8/
3 ,15X,'Chamber Temperature (deg K) at Burnout =',E16.8,/
4 ,15X,'Depth of Layer (cm) at Burnout =',E16.8)
C
C** Set Final Value of "Xi" for this Layer "J"
C
    XI (J) = 0.0
C
C** SET INDEX OF NEW LAYER "J"
C
    J = J + 1
C
C** CHECK ON TOTAL PROPELLANT BURNOUT
C
    IF (J .GT. NTL) GO TO 1010
C
C** Set Properties for fractional time step integration
C
    FRACT = (1.0 - ABS (SMSN)/(SMSN1 - SMSN))
    SMSNO = SMSN
    KN = 1
C
    SMSN1 = SMSOL (J)

```

```

      SMSN = SMSOL (J) + SMSNO
      RHOSN1 = PX (4, J, 1)/RHOSTR
      CALL BURNRT (PCHBO, J, RBN1)
C
C** Compute total propellant mass remaining in chamber
C   when "old layer J" burned out
C
      SUMI = 0.
      DO 1050 I = 1, NTL
        SUMI = SUMI + XI (I)*SMSOL (I)
1050    CONTINUE
C
      TPMRGM = SUMI*SMSSCL
C
C*****
C** Call "DEPTH-BURNED" Subroutine to get Surface area (ONLY)
C*****
C
      CALL DEPTH (TPMRGM, J, 2, DEPTHCM, ASURCM2)
      ASURN1 = ASURCM2/SURSTR
C
C*****
C*****
C***** Now Reset All Sums Which Remain Constant in New Layer "J" *****
C*****
C**** CALL SHEVAL (J,NTL) [SH1,SH2,SH3,SH4 returned in COMMON/SUMS] *
C*****
C*****
C
      CALL SHEVAL (J, NTL)
C
C*****
C*****
C
      END IF
C
      GO TO 1020
C
      ELSE
C
      GO TO 1000
C
      END IF
C
1010 CONTINUE
C
      RETURN

```

```

END
C
C*****
C*****
C
SUBROUTINE BRRED (A1)
C
C*****
C*****
C** This routine computes effective linear regression rates based on ***
C*****
C** a given chamber pressure as a function of time (from UNIT 13) **
C*****
C*****
COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /ETC/ EE (1000)
C
COMMON /CONTRL/ NTL, CHVOL, HLFAC, DT, NMAX, CONVRG, ICTYP
COMMON /FIXED/ CH1, CH2, CH3, CAPEO, QWMAX, PCHAMX
COMMON /SUMS/ SH1, SH2, SH3, SH4
COMMON /NVALUE/ RHOSRN, ASURN, DPTHBN, TPMRGM
COMMON /SOLUT/ TIME (1000), PCH (1000), RBR
1 (1000), TPMR (1000), SMS (1000)
2 , RHOS (1000), TCH (1000), DPTHB (1000),
3 ASUR (1000), LYR (1000)
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
COMMON /SOLIDL/ SMSOL (15), XI (15)
COMMON /AVEPROP/ RHOSAV (15), SESOAV (15), SBCGAV (15), RCGGAV (15),
1 CVC GAV (15)
COMMON /IGNITOR/ SMSIG, SESIG, RHOSIG, WTMIG, SBIG, RGIG, CVIG
COMMON /AIR/ SMSA, WTMA, SBA, RGA, CVA
COMMON /METH/ IMOD
C
DIMENSION NLOW (15), NHIH (15)
CHARACTER*20 A1 (20)
C
C*****
C*****
1000 CONTINUE
OPEN (UNIT = 13, FILE = A1 (14), STATUS = 'OLD', ERR = 1010)
REWIND (13)
GO TO 1020
1010 CONTINUE
CALL CLEAR
WRITE (*, *) 'The Pressure-Time File Does NOT Exist'

```

```

      WRITE (*, *) 'Enter a New File Name For The P-T Data'
      READ (*, 5000) A1 (14)
      5000 FORMAT(A20)
      GO TO 1000
      1020 CONTINUE
C*****
C****
C
      J = 1
C
C*****
C****
C***** Compute Sums Which Remain Constant in Layer "J" (=1) *****
C****
C**** CALL SHEVAL (J,NTL) [SH1,SH2,SH3,SH4 returned in COMMON/SUMS] *
C****
C*****
C****
C
      CALL SHEVAL (J, NTL)
C
C*****
C****
C***** COMPUTE INITIAL CONDITIONS IN CHAMBER WITH IGNITOR CONSUMED **
C****
C*****
C****
      IF (ICTYP .NE. 3) THEN
        WRITE (11, 6000)
        6000 FORMAT(/, ' ** Results: Deduced Burning Rates Based'
          1 , ' on a given Chamber Pressure vs. Time **' //
          2 ,3x, ' *** N *** LAYER *** TIME(ms) *** PCH(MPa) *** TPMR(gm)'
          3 , ' *** RBR(cm/s) *** ASUR(cm2) *** TCH(degK) *** DEPTHB(cm) ***'
          4,/)
        ELSE
          WRITE (11, 6010)
          6010 FORMAT(/, ' ** Results: Deduced Surface Area Based'
            1 , ' on a given Chamber Pressure vs. Time **' //
            2 ,3x, ' *** N *** LAYER *** TIME(ms) *** PCH(MPa) *** TPMR(gm)'
            3 , ' *** RBR(cm/s) *** ASUR(cm2) *** TCH(degK) *** DEPTHB(cm) ***'
            4,/)
        END IF
C
C*** COMPUTE INITIAL VALUES OF "PCHN" AND "TCHN" FOR "XIN"=1.0 ***
C
      XIN = 1.0
C
C*****

```

```

C*****
C** Call "PTCALC" Subroutine to get initial values of chamber pressure a
Cnd
C** temperature PLUS the initial surface area AND all integrals
C
CALL PTCALC (J, XIN, PCHN, TCHN, QWN)
C
C*****
C
PCH (1) = PCHN
TCH (1) = TCHN
RHOS (1) = PX (4, J, 1)/RHOSTR
C
SMS (1) = SMSOL (J)
ASUR (1) = ASURN
DPTHB (1) = DPTHBN
LYR (1) = J
C
TIME (1) = 0.0
C
CALL OUTPUT (1)
C*****
C***** FIRST MUST DETERMINE LOCATION IN PRESSURE DATA *****
C***** SO THAT PRESSURE IS ALWAYS ABOVE IGNITER PRESSURE *****
C*****
NPNX = 0
LNPNX = 0
DUMTT = 0.0
CALL CLEAR
1030 CONTINUE
READ (13, *, END = 1040) DUMT, DUMP
NPNX = NPNX + 1
IF (DUMP .LE. PCH (1)*PRSTR) THEN
LNPNX = NPNX
DUMTT = DUMT
GO TO 1030
END IF
1040 CONTINUE
WRITE (*, 6020) LNPNX, DUMTT*1000. + A3 (42), PCH (1)*PRSTR
6020 FORMAT(///,5X,'For the calculation to be performed correctly',
1/,5X,'all pressures must be above the igniter pressure. For',
2/,5X,'the current data set points had to be deleted to obtain',
3/,5X,'all pressures above the igniter pressure. Information',
4/,5X,'relative to the deleted data points is:',
5//,5X,'Number of points deleted',:'.I5,
6/,5X,'Time interval of deleted points (ms):',F10.5,
7/,5X,'Igniter Pressure (MPa)',:'.F10.5//)
PAUSE

```

```

        REWIND (UNIT = 13)
        DO 1050 KXK = 1, LNPNX
            READ (13, *) DUMT, DUMP
1050 CONTINUE
        CALL CLEAR
C
C*****
C*****
C***** BEGIN COMPUTATION MASS REMAINING IN LAYER "J" (SMSN) VS. TIME **
C*****
C*****
C
        N = 1
C
C** Initial guess for SMSN and PCHN
C (only used to get average properties)
C
        SMSN = SMS (N)
        PCHN = PCH (N)
C
C*****
C***** "490" LOOP COMPUTES "SMS" vs TIME *****
C*****
C
1060 CONTINUE
C
        N = N + 1
C
        IF (N .GT. NMAX) GO TO 1070
C
C*****
C** READ "PCH(N)" AND "TIME(N)" FROM UNIT 13
C
C** NOTE: It is assumed that PCH on 13 has units "MPa"
C**          and TIME on 13 has units of "mil-sec"
C
C*****
C***** IF OPTION 5, ELECTRICAL ENERGY MUST BE READ *****
C***** MUST ALSO CONSIDER IF PRESSURE IS *****
C*****
        IF (ICTYP .EQ. 5) THEN
            READ (13, *, END = 1070, ERR = 1070) DUMT, DUMP, XXX, EEL
            EE (N) = EEL
        ELSE
            READ (13, *, END = 1070, ERR = 1070) DUMT, DUMP
        END IF
C

```

```

      PCH (N) = DUMP/PRSTR
C*****
C*** Compute Current Value of Cumulative Heat Loss "QWN" *****
C***** ADD ELECTRICAL ENERGY TO "QWN" WILL BE *****
C***** SUBTRACTED FROM "QWN" BACK FROM QWCOMP ***(Huh?)***
C*****
      CALL QWCOMP (QWN, PCH (N))
      IF (ICTYP .EQ. 5) THEN
        QWN = QWN - EE (N)*1000000./(4.184*SERSCL*SMSSCL)
      END IF
C*****
C***** ACTUAL TIME ON UNIT 13 IS SEC *****
C***** WILL CONVERT TO MIL-SEC *****
C*****
      TIME (N) = DUMT*1000./TIMSCL
      CALL CLEAR
      IETY = INT (A3 (31) + .5)
      IF (IETY .EQ. 1) THEN
        WRITE (*, *) ' Using Constant Values'
      ELSE
        WRITE (*, *) ' Using Integrals'
      END IF
      WRITE (*, 6030) N, DUMT*1000., DUMP, PCH (1)*PRSTR
6030 FORMAT(/,5X,'*** BR Computation on Step =',I4,
1/,8X,'where TIME (millisec) =',F9.3,
25X,'and Pressure (MPa) =',F10.5,
3/8X,'NOTE: Pressure must Exceed Ignitor Pressure (MPa)=',
4 F10.5)
C
C*****
C**** BEGIN ITERATION FOR VALUE OF "SMSN" AT TIME STEP "N" *****
C*****
C***** Begin Iterative "500" Loop to Find SMS(N): *****
C**** (Loop will try 20 times for convergence, KN <= 20) *****
C
      IDONE = 0
      KN = 0
C
1080 CONTINUE
C
      IF (KN .LE. 20 .AND. IDONE .NE. 1) THEN
C
        KN = KN + 1
        SMSLAS = SMSN
C
C**** It is assumed below that LAYER J is still burning
C
      XIN = SMSN/SMSOL (J)

```

```

C
C*****
C** Latest value of Total Propellant Mass Remaining in Chamber **
C*****
C
      JP1 = J + 1
      SUMI = 0.0
C
      IF (JP1 .LE. NTL) THEN
        DO 1090 I = JP1, NTL
          SUMI = SUMI + SMSOL (I)
1090    CONTINUE
      END IF
C
      TPMRN = SUMI + SMSN
      TPMRGM = TPMRN*SMSSCL
C
C*****
C** Call "DEPTH-BURNED" Subroutine to get surface area + all integrals
C*****
C
      CALL DEPTH (TPMRGM, J, 1, DEPTHCM, ASURCM2)
C
C*****
C
      DPTHBN = DEPTHCM/ELSTR
      ASURN = ASURCM2/SURSTR
C
      VOLRM = PX (8, J, 3) - PX (8, J, 5)
      RATV = PX (8, J, 5)/PX (8, J, 3)
C
      IF (RATV .LT. 0.02) THEN
        SBCGN = PX (6, J, 1)*RHOSTR
        RGCGN = PX (9, J, 1)/CVRSCCL
        CVCGN = PX (10, J, 1)/CVRSCCL
        RHOSRN = ((PX (4, J, 3) - PX (4, J, 5))/VOLRM)/RHOSTR
        SESRN = ((PX (11, J, 3) - PX (11, J, 5))/VOLRM)/SERSCL
      ELSE IF (RATV .GT. 0.98) THEN
        SBCGN = (PX (6, J, 5)/PX (8, J, 5))*RHOSTR
        RGCGN = (PX (9, J, 5)/PX (8, J, 5))/CVRSCCL
        CVCGN = (PX (10, J, 5)/PX (8, J, 5))/CVRSCCL
        RHOSRN = PX (4, J, 2)/RHOSTR
        SESRN = PX (11, J, 2)/SERSCL
      ELSE
        SBCGN = (PX (6, J, 5)/PX (8, J, 5))*RHOSTR
        RGCGN = (PX (9, J, 5)/PX (8, J, 5))/CVRSCCL
        CVCGN = (PX (10, J, 5)/PX (8, J, 5))/CVRSCCL
        RHOSRN = ((PX (4, J, 3) - PX (4, J, 5))/VOLRM)/RHOSTR

```

```

      SESRN = ((PX (11, J, 3) - PX (11, J, 5))/VOLRM)/SERSCL
END IF
C
      AMSESO = SMSOL (J)*SESRN
      AMCVCG = SMSOL (J)*CVCGN
      AMSBCG = SMSOL (J)*SBCGN
      AMRGCG = SMSOL (J)*RGCGN
      AMRHOS = SMSOL (J)/RHOSRN
C
C*****
C
      C1 = CH1 - SH1 - AMSBCG
      C2 = CH2 + SH2 + AMRGCG
      C3 = CH3 + SH3 + AMCVCG
      C4 = CAPEO - QWN - SH4
C
C*** COMPUTE CURRENT VALUE OF BURNOUT PRESSURE; PCHBO
C
      PCHBON = (C2*C4)/(C1*C3)
C
C*** Check to see if PCH(N) is less than current burnout pressure
C
      IF (PCH (N) .LE. PCHBON) THEN
C
      QA = AMRGCG*AMSESO + PCH (N)*AMCVCG*(AMSBCG - AMRHOS)
      QB = - (C2*AMSESO + C4*AMRGCG
1      + PCH (N)*(C3*(AMSBCG - AMRHOS) - C1*AMCVCG))
      QC = C2*C4 - PCH (N)*C1*C3
C
      DISC = QB*QB - 4.0*QA*QC
C
      IF (DISC .GE. 0.0) THEN
        XIN = - (QB + SQRT (DISC))/(2.0*QA)
      ELSE
C
        WRITE (11, 6040) N, KN, PCH (N), PCHBON, TIME (N), QA, QB, QC
6040 FORMAT(///, ' **** ERROR **** "DISC" IS NEGATIVE !!!',
1 ' ON STEP N =', I5, ' WHEN KN = ', I2, '/', 5X, 'PCH(N)=',
2 E16.8, ' PCHBON=', E16.8, ' AND TIME(N)=', E16.8, '/',
35X, '(QA,QB,QC)=', 3E18.8, //)
C
        STOP
      END IF
C
      SMSN = SMSOL (J)*XIN
C
      IF (ABS ((SMSN - SMSLAS)/SMSLAS) .LT. CONVRG .OR. KN .GE. 20) THEN

```

```

C
C** Compute latest value of Chamber Temperature
C
      TOP = CAPEO - QWN - SH4 - AMSESO*XIN
      BTM = CH3 + SH3 + AMCVCG*(1.0 - XIN)
      TCHN = TOP/BTM
C
C*****
C** Latest value of Total Propellant Mass Remaining in Chamber **
C*****
C
      JP1 = J + 1
      SUMI = 0.0
C
      IF (JP1 .LE. NTL) THEN
        DO 1100 I = JP1, NTL
          SUMI = SUMI + SMSOL (I)
1100      CONTINUE
        END IF
C
      TPMRN = SUMI + SMSN
      TPMRGM = TPMRN*SMSSCL
C
C*****
C** Call "DEPTH-BURNED" Subroutine to get Surface area (ONLY)
C*****
C
      CALL DEPTH (TPMRGM, J, 2, DEPTHCM, ASURCM2)
C
      DPTHBN = DEPTHCM/ELSTR
      ASURN = ASURCM2/SURSTR
      RHOSN = (PX (4, J, 1) + (PX (4, J, 2) - PX (4, J, 1))*
1      ((DEPTHCM-PX (1, J, 1))/PX(1, J, 2)))/RHOSTR
C
C*****
      TCH (N) = TCHN
      SMS (N) = SMSN
C
      RHOS (N) = RHOSN
      ASUR (N) = ASURN
      DPTHB (N) = DPTHBN
      TPMR (N) = TPMRN
      LYR (N) = J
C*****
C** After successful completion of step, load "5" values into "4"
C
      DO 1110 KV = 2, 11

```

```

          PX (KV, J, 4) = PX (KV, J, 5)
1110      CONTINUE
C*****
C
          IDONE = 1
C
          ELSE
          END IF
C
          ELSE
C
C*****
C***** Layer "J" has Burned Out *****
C**** Must Reset Sums and Switch Layers to J = J + 1 *****
C*****
C** COMPUTE PROPERTIES AT BURNOUT OF LAYER "J"
C
          XIN = 0.0
C
          CALL PTCALC (J, XIN, PCHBO, TCHBO, QWBO)
C
C** Print/Write BURNOUT VALUES **
C
          TIMBOP = TIME (N)*TIMSCL
          PCHBOP = PCHBO*PRSTR
          TCHBOP = TCHBO*TRSTR
          DPTHOP = DPTHBN*ELSTR
C
          WRITE (11, 6050) J, N, TIMBOP, PCHBOP, TCHBOP, DPTHOP
6050 FORMAT(/' *** PROPELLANT LAYER J= ',I2,' has Burned '
1 ,',out on Step N= ',I5,' Before TIME (millisec) of ',E16.8/
2 ,',15X,'Chamber Pressure (MPa) at Burnout =',E16.8/
3 ,',15X,'Chamber Temperature (deg K) at Burnout =',E16.8./
4 ,',15X,'Depth of Layer (cm) at Burnout =',E16.8)
C
C** Set Final Value of "Xi" for this Layer "J"
C
          XI (J) = 0.0
C
C** SET INDEX OF NEW LAYER "J"
C
          J = J + 1
C
C** CHECK ON TOTAL PROPELLANT BURNOUT
C
          IF (J .GT. NTL) GO TO 1070
C
C** Set Properties for Continuation of Integration

```

```

C
      KN = 1
      SMSN = SMSOL (J)
C
C*****
C*****
C***** Now Reset All Sums Which Remain Constant in New Layer "J" *****
C*****
C***** CALL SHEVAL (J,NTL) [SH1,SH2,SH3,SH4 returned in COMMON/SUMS] *
C*****
C*****
C*****
C
      CALL SHEVAL (J, NTL)
C
C*****
C*****
C
      END IF
C
      GO TO 1080
C
      ELSE
C
      GO TO 1060
C
      END IF
C
1070 CONTINUE
C
C*** BEGIN COMPUTATION OF RATE FROM SMS(N) AND TIME(N)
C
      NTOT = N - 1
      NTOT1 = NTOT - 1
C
C [NOTE: *EQUAL TIME STEPS* HAVE BEEN ASSUMED HERE]
C
      WRITE (11, 6060)
6060 FORMAT(/, ' *** DEDUCED BURNING RATES ***',/)
C
      DO 1120 N = 3, NTOT1
C
      JLNMI = LYR (N - 1)
      JLN = LYR (N)
      JLNPI = LYR (N + 1)
C
      IF (JLNMI .EQ. JLN .AND. JLNPI .EQ. JLN) THEN
          DMSDTN = (TPMR (N + 1) - TPMR (N - 1))/(TIME (N + 1) - TIME (N - 1))

```

```

ELSE IF (JLNM1 .EQ. JLN .AND. JLNPI .NE. JLN) THEN
  DMSDTN = ( - 3.0*TPMR (N) + 4.0*TPMR (N - 1) - TPMR (N -
1    2)))/(TIME (N - 2) - TIME (N))
ELSE
  DMSDTN = ( - 3.0*TPMR (N) + 4.0*TPMR (N + 1) - TPMR (N +
1    2)))/(TIME (N + 2) - TIME (N))
END IF

C*****
C***** NEED TO COMPUTE THE DESIRED QUANTITY *****
C***** OPTIONS 1, 4 & 5: BURN RATE *****
C***** OPTION 3: SURFACE AREA *****
C*****

IF (ICTYP .EQ. 3) THEN
  CALL BURNRT (PCH (N), LYR (N), RBR (N))
  ASUR (N) = - (DMSDTN/(RHOS (N)*RBR (N)))
ELSE
  RBR (N) = - (DMSDTN/(RHOS (N)*ASUR (N)))
END IF

C
  CALL OUTPUT (N)
C
1120 CONTINUE
C
  RETURN
  END
C
C*****
C*****
  SUBROUTINE SHEVAL (JL, NTO)
C*****
C*****
C** NOTATION: JL = LOCAL "J" LAYER TO BE EXCLUDED FROM SUMS      **
C*****
C**      NTO = TOTAL NUMBER OF PROPELLANT LAYERS      **
C*****
C** (output)  SH1,SH2,SH3,SH4  returned in COMMON/SUMS      **
C*****
C*****
C*****
  COMMON /SUMS/ SH1, SH2, SH3, SH4
  COMMON /SOLIDL/ SMSOL (15), XI (15)
  COMMON /AVEPROP/ RHOSAV (15), SESOAV (15), SBCGAV (15), RGCGAV (15),
1  CVCGAV (15)
C*****
C*****
C***** Form Sums Which Remain Constant in Layer "JL"
C
  SUMI = 0.0

```

```

      DO 1000 I = 1, NTO
        SUMI = SUMI + SMSOL (I)*(XI (I)/RHOSAV (I) + SBCGAV (I)*(1.0 - XI (I)))
      1000 CONTINUE
      C
        SH1 = SUMI - SMSOL (JL)*(XI (JL)/RHOSAV (JL) + SBCGAV (JL)*(1.0 - XI (JL)))
      C
      C*****
      C
        SUMI = 0.0
        DO 1010 I = 1, NTO
          SUMI = SUMI + SMSOL (I)*RGCGAV (I)*(1.0 - XI (I))
        1010 CONTINUE
      C
        SH2 = SUMI - SMSOL (JL)*RGCGAV (JL)*(1.0 - XI (JL))
      C
      C*****
      C
        SUMI = 0.0
        DO 1020 I = 1, NTO
          SUMI = SUMI + SMSOL (I)*CVCGAV (I)*(1.0 - XI (I))
        1020 CONTINUE
      C
        SH3 = SUMI - SMSOL (JL)*CVCGAV (JL)*(1.0 - XI (JL))
      C
      C*****
      C
        SUMI = 0.0
        DO 1030 I = 1, NTO
          SUMI = SUMI + SMSOL (I)*XI (I)*SESOAV (I)
        1030 CONTINUE
      C
        SH4 = SUMI - SMSOL (JL)*XI (JL)*SESOAV (JL)
      C
      C*****
      C*****
      C*****
      C*****
      RETURN
      END
      C*****
      C*****
      SUBROUTINE PTCALC (J, XIK, PCHK, TCHK, QWK)
      C*****
      C*****
      C*** Subroutine "PTCALC" calculates PCH and TCH for Layer "J" given the
      C ****

```

```

C*** "Kth" iterative value of the mass fraction "XI" ;[XI=SMS/SMSO]
C ****
C*****
C*****
COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /ETC/ EE (1000)
C
COMMON /FIXED/ CH1, CH2, CH3, CAPEO, QWMAX, PCHAMX
COMMON /SUMS/ SH1, SH2, SH3, SH4
C
COMMON /NVALUE/ RHOSRN, ASURN, DPTHBN, TPMRGM
COMMON /SOLIDL/ SMSOL (15), XI (15)
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
COMMON /METH/ IMOD
C*****
C*****
C** Compute the latest value of Total Propellant Mass Remaining in Chamb
Cer **
C*****
C*****
C
NLT = INT (A3 (4) + 0.5)
JP1 = J + 1
C
SUMI = 0.0
IF (JP1 .LE. NLT) THEN
DO 1000 I = JP1, NLT
SUMI = SUMI + SMSOL (I)
1000 CONTINUE
END IF
C
TPMRGM = (SUMI + XIK*SMSOL (J))*SMSSCL
C
C*****
C** Call "DEPTH-BURNED" Subroutine to get Surface area + all integrals
C*****
C
CALL DEPTH (TPMRGM, J, 1, DEPTHCM, ASURCM2)
C
C*****
DPTHBN = DEPTHCM/ELSTR
ASURN = ASURCM2/SURSTR
C
VOLRM = PX (8, J, 3) - PX (8, J, 5)
RATV = PX (8, J, 5)/PX (8, J, 3)
C
IF (RATV .LT. 0.02) THEN

```

```

SBCGN = PX (6, J, 1)*RHOSTR
RGCGN = PX (9, J, 1)/CVRSCSCL
CVCN = PX (10, J, 1)/CVRSCSCL
RHOSRN = ((PX (4, J, 3) - PX (4, J, 5))/VOLRM)/RHOSTR
SESRN = ((PX (11, J, 3) - PX (11, J, 5))/VOLRM)/SERSCL
ELSE IF (RATV .GT. 0.98) THEN
  SBCGN = (PX (6, J, 5)/PX (8, J, 5))*RHOSTR
  RGCGN = (PX (9, J, 5)/PX (8, J, 5))/CVRSCSCL
  CVCN = (PX (10, J, 5)/PX (8, J, 5))/CVRSCSCL
  RHOSRN = PX (4, J, 2)/RHOSTR
  SESRN = PX (11, J, 2)/SERSCL
ELSE
  SBCGN = (PX (6, J, 5)/PX (8, J, 5))*RHOSTR
  RGCGN = (PX (9, J, 5)/PX (8, J, 5))/CVRSCSCL
  CVCN = (PX (10, J, 5)/PX (8, J, 5))/CVRSCSCL
  RHOSRN = ((PX (4, J, 3) - PX (4, J, 5))/VOLRM)/RHOSTR
  SESRN = ((PX (11, J, 3) - PX (11, J, 5))/VOLRM)/SERSCL
END IF
C
AMSESO = SMSOL (J)*SESRN
AMCVCN = SMSOL (J)*CVCN
AMSBCN = SMSOL (J)*SBCGN
AMRGCG = SMSOL (J)*RGCGN
AMRHOS = SMSOL (J)/RHOSRN
C
C*** Iterate 3 times for "PCHK" and "TCHK"
C
DO 1010 KI = 1, 3
C
C*** Compute Current Value of Cumulative Heat Loss "QWK"
C
CALL QWCOMP (QWK, PCHK)
C
TOP = CAPEO - QWK - SH4 - AMSESO*XIK
BTM = CH3 + SH3 + AMCVCN*(1.0 - XIK)
TCHK = TOP/BTM
C
TOP = TCHK*(CH2 + SH2 + AMRGCG*(1.0 - XIK))
BTM = CH1 - SH1 - AMSBCN*(1.0 - XIK) - AMRHOS*XIK
PCHK = TOP/BTM
C
1010 CONTINUE
C
RETURN
END
C
C*****
C*****

```

```

SUBROUTINE QWCOMP (QWO, PCHNO)
C*****
C*****
COMMON /FIXED/ CH1, CH2, CH3, CAPEO, QWMAX, PCHAMX
C
QWO = QWMAX*(PCHNO/PCHAMX)
C
C*****
C*****
RETURN
END
C*****
C*****
C

SUBROUTINE BURNRT (PG, JL, RBRJ)
C
C PG: SCALED (ND) PRESSURE
C JL: LAYER BURNING
C RBRJ: SCALED (ND) BURN RATE
C*****
C*****
COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /CONTRL/ NTL, CHVOL, HLFRAC, DT, NMAX, CONVRG, ICTYP
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
COMMON /AVEPROP/ RHOSAV (15), SESOAV (15), SBCGAV (15), RGCGAV (15),
1 CVC GAV (15)
C*****
C***** ACTUAL PRESSURE IS PGMPA *****
C*****
PGMPA = PG*PRSTR
C*****
C GENERAL ROUTINE TO FIND BURN RATE, BASED UPON PRESSURE NOT THE LAYER
C*****
IF (A3 (5) .GT. 0.0) THEN
NNUM = A3 (5)
LAST = NNUM + 49
C*****
C***** CURRENT PRESSURE ABOVE ALL PRESSURE ENTRIES *****
C*****
IF (PGMPA .GT. A3 (LAST)) THEN
XY1 = ALOG10 (A3 (LAST + 30))
XY2 = ALOG10 (A3 (LAST + 29))
XY3 = ALOG10 (A3 (LAST))
XY4 = ALOG10 (A3 (LAST - 1))
XY5 = ALOG10 (PGMPA)
RBRJ = (XY1 - XY2)/(XY3 - XY4)*(XY5 - XY3) + XY1

```

```

        RBRJ = 10.**RBRJ
        RBRJ = RBRJ/RBRSTR
        RETURN
    END IF
C*****
C***** CURRENT PRESSURE BELOW ALL ENTRIES *****
C*****
    IF (PGMPA .EQ. A3 (50)) THEN
        RBRJ = A3 (80)
        RBRJ = RBRJ/RBRSTR
        RETURN
    END IF
    IF (PGMPA .LT. A3 (50)) THEN
        XY1 = ALOG10 (A3 (81))
        XY2 = ALOG10 (A3 (80))
        XY3 = ALOG10 (A3 (51))
        XY4 = ALOG10 (A3 (50))
        XY5 = ALOG10 (PGMPA)
        RBRJ = (XY1 - XY2)/(XY3 - XY4)*(XY5 - XY4) + XY2
        RBRJ = 10.**RBRJ
        RBRJ = RBRJ/RBRSTR
        RETURN
    END IF
C*****
C***** PRESSURE IN MIDDLE *****
C*****
    DO 1000 JJ = 1, NNUM
        JJJ = JJ + 49
        IF (PGMPA .LE. A3 (JJJ)) THEN
            JJJJ = JJ + 79
            XY5 = ALOG10 (PGMPA)
            XY1 = ALOG10 (A3 (JJJ - 1))
            XY2 = ALOG10 (A3 (JJJ))
            XY3 = ALOG10 (A3 (JJJJ))
            XY4 = ALOG10 (A3 (JJJJ - 1))
            RBRJ = (XY5 - XY1)/(XY2 - XY1)*(XY3 - XY4) + XY4
            RBRJ = 10.**RBRJ
            RBRJ = RBRJ/RBRSTR
            RETURN
        END IF
    1000 CONTINUE
    END IF
C*****
C***** NEXT THE ROUTINE IS BASED UPON THE LAYER *****
C*****
    IF (A3 (5) .LT. 0.0) THEN
        NNUM = ABS (A3 (5))
        INDEX = 50 + 2*(JL - 1)

```

```

      RBRJ = (A3 (INDEX)/RBRSTR)*PGMPA**A3 (INDEX + 1)
    END IF
    RETURN
  END
C*****
C****
C
  SUBROUTINE SETVAL
C
C*****
C****
  COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
  COMMON /ETC/ EE (1000)
  COMMON /CONTRL/ NTL, CHVOL, HLFRA, DT, NMAX, CONVRG, ICTYP
  COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
  1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
  COMMON /SOLIDL/ SMSOL (15), XI (15)
  COMMON /IGNITOR/ SMSIG, SESIG, RHOSIG, WTMIG, SBIG, RGIG, CVIG
  COMMON /AIR/ SMSA, WTMA, SBA, RGA, CVA
C*****
C****
C** Set Total Number of Propellant Layers (NTL), Chamber Volume (CHVOLS)
C;
C** Time Step (DTS IN MSEC), Maximum Number of Time Steps (NMAX),
C** and Convergence Tolerance (CONVRG)
C
  NTL = INT (A3 (4) + .5)
  CHVOLS = A3 (23)
  DTS = A3 (42)
  NMAX = 1200
  ICTYP = INT (A3 (2) + .5)
  HLFRA = 0.0
C
C** Scale Chamber Volume (CHVOL)
C
  CHVOL = CHVOLS/VOLSTR
C
C** Scale Time Step (DT)
C
  DT = DTS/TIMSCL
C
C** Set ND-Values of Propellant Mass (SMSOL)
C
  DO 1000 I = 1, NTL
    SMSOL (I) = PX (4, I, 3)*A3 (46)/SMSSCL
  1000 CONTINUE
C
  SMSIG = A3 (26)/SMSSCL

```

```

C*****
C***** AIR MASS IS DETERMINED *****
C*****
    SUM = 0.0
    DO 1010 I = 1, NTL
        SUM = SUM + PX (8, I, 3)
1010 CONTINUE
    SUM = SUM*A3 (46)
    SUM = SUM + A3 (26)/A3 (15)
    SUM = CHVOLS - SUM
    SMSA = SUM*.001204
    CALL CLEAR
    WRITE (*, 6000) SMSA
6000 FORMAT(///,10X,'The computed mass of air in the bomb is:',F10.5,
1/,10X,'grams. Do you wish to change this value to 0.0 gm?',
2/,10X,'(Yes = 1 / No = 2)'//,10X,'Enter your choice.'//)
    READ (*, *) IAIR
    IF (IAIR .EQ. 1) THEN
        SMSA = 0.0
    ELSE
        SMSA = SMSA/SMSSCL
    END IF
C
C** Set all values for "---IG" (ignitor) and "---A" (air)
C
    WTMIG = A3 (16)
    WTMA = 28.9
    GAMMAA = 1.4
C
    SBIG = A3 (17)*RHOSTR
    SBA = .98*RHOSTR
    RGIG = (1.98717/A3 (16))/CVRSCCL
    RGA = (1.98717/WTMA)/CVRSCCL
    CVIG = RGIG/(A3 (18) - 1.0)
    CVA = RGA/(GAMMAA - 1.0)
C
    SESIG = (A3 (14)/TRSTR)*CVIG
    RHOSIG = A3 (15)/RHOSTR
C
C*****
C***
C***** WRITE TABULATION OF INPUT DATA TO UNIT (11) *****
C***
C*****
C***
C
    WRITE (11, 6010)
    WRITE (11, 6020)

```

```

WRITE (11, 6020)
C
CHVOLP = CHVOL*VOLSTR
DTP = DT*TIMSCL
C
WRITE (11, 6030) NTL, CHVOLP, HLFRAC, DTP, NMAX, CONVRG, ICTYP
WRITE (11, 6020)
WRITE (11, 6040)
WRITE (11, 6020)
WRITE (11, 6050)
WRITE (11, 6020)
C
DO 1020 I = 1, NTL
    WRITE (11, 6060) I, SMSOL (I)*SMSSCL
1020 CONTINUE
    SMSIGP = SMSIG*SMSSCL
    SMSAP = SMSA*SMSSCL
    WRITE (11, 6070) SMSIGP
    WRITE (11, 6080) SMSAP
C***
    WRITE (11, 6020)
    WRITE (11, 6090)
    WRITE (11, 6020)
C***
    DO 1030 I = 1, NTL
        WRITE (11, 6060) I, PX (4, I, 1), PX (4, I, 2)
1030 CONTINUE
        WRITE (11, 6070) A3 (15)
C***
        WRITE (11, 6020)
        WRITE (11, 6100)
        WRITE (11, 6020)
C***
        DO 1040 I = 1, NTL
            WRITE (11, 6060) I, PX (2, I, 1), PX (2, I, 2)
1040 CONTINUE
            WRITE (11, 6070) A3 (13)
C***
            WRITE (11, 6020)
            WRITE (11, 6110)
            WRITE (11, 6020)
C***
            DO 1050 I = 1, NTL
                WRITE (11, 6060) I, PX (3, I, 1), PX (3, I, 2)
1050 CONTINUE
                WRITE (11, 6070) A3 (14)
C***
                WRITE (11, 6020)

```

```

WRITE (11, 6120)
WRITE (11, 6020)
C***
DO 1060 I = 1, NTL
    WRITE (11, 6060) I, PX (5, I, 1), PX (5, I, 2)
1060 CONTINUE
    WRITE (11, 6070) A3 (16)
    WRITE (11, 6080) WTMA
C***
    WRITE (11, 6020)
    WRITE (11, 6130)
    WRITE (11, 6020)
C***
DO 1070 I = 1, NTL
    WRITE (11, 6060) I, PX (9, I, 1), PX (9, I, 2)
1070 CONTINUE
    RGIGP = RGIG*CVRSCSCL
    WRITE (11, 6070) RGIGP
    RGAP = RGA*CVRSCSCL
    WRITE (11, 6080) RGAP
C***
    WRITE (11, 6020)
    WRITE (11, 6140)
    WRITE (11, 6020)
C***
DO 1080 I = 1, NTL
    WRITE (11, 6060) I, PX (7, I, 1), PX (7, I, 2)
1080 CONTINUE
    WRITE (11, 6070) A3 (18)
    WRITE (11, 6080) GAMMAA
C***
    WRITE (11, 6020)
    WRITE (11, 6150)
    WRITE (11, 6020)
C***
DO 1090 I = 1, NTL
    WRITE (11, 6060) I, PX (10, I, 1), PX (10, I, 2)
1090 CONTINUE
    CVIGP = CVIG*CVRSCSCL
    WRITE (11, 6070) CVIGP
    CVAP = CVA*CVRSCSCL
    WRITE (11, 6080) CVAP
C***
    WRITE (11, 6020)
    WRITE (11, 6160)
    WRITE (11, 6020)
C***
DO 1100 I = 1, NTL

```

```

        WRITE (11, 6060) I, PX (6, I, 1), PX (6, I, 2)
1100 CONTINUE
        WRITE (11, 6070) A3 (17)
        SBAP = SBA/RHOSTR
        WRITE (11, 6080) SBAP
C***
        WRITE (11, 6020)
        WRITE (11, 6020)
C*****
        RETURN
C*****
C***** FORMAT STATEMENTS *****
C*****
6010 FORMAT(//)
6020 FORMAT(' *****',
1 '*****')
6030 FORMAT(5X,'Total # Layers = ',I4/,
1 5X,'Chamber Volume (cm3) =',F10.3/,
2 5X,'Heat-Loss-Fraction (n-d) =',F10.3/,
3 5X,'Time Step (mil-sec) =',E16.8,5X,'Max Time Steps = ',I7/,
4 5X,'Convergence Criterion = ',E16.8/,
5 5X,'Computation Type (1=BURN-RATE, 2=PTGEN, 3=SURF AREA, 4=INTERR
6 BOMB, 5=ETC)',I5)
6040 FORMAT(20X,'Beginning(1) of Layer',10X,'End(2) of Layer')
6050 FORMAT(10X,'Propellant Mass in grams')
6090 FORMAT(10X,'Propellant Density in g/cc')
6110 FORMAT(10X,'Propellant Flame Temperature in deg K')
6120 FORMAT(10X,'Propellant Molecular Weight in g/g-mole')
6130 FORMAT(10X,'Propellant Gas Constant in cal/g-deg K')
6140 FORMAT(10X,'Gamma (Ratio of Specific Heats)')
C
6060 FORMAT(5X,I5,10X,F15.5,10X,F15.5)
6070 FORMAT(6X,'Ignitor',7X,F15.5)
6080 FORMAT(8X,'Air',9X,F15.5)
C
6150 FORMAT(10X,'Specific Heat (Cv) in cal/g-deg K')
6160 FORMAT(10X,'Covolume in cm3/g')
6100 FORMAT(10X,'Propellant Impetus in J/g')
C*****
        END
C*****
C*****
        SUBROUTINE OUTPUT (NT)
C*****
C*****
        COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
        COMMON /CONTRL/ NTL, CHVOL, HLFrac, DT, NMAX, CONVRG, ICTYP
        COMMON /SOLUT/ TIME (1000), PCH (1000), RBR

```

```

1 (1000), TPMR (1000), SMS (1000)
2 , RHOS (1000), TCH (1000), DPTHB (1000),
3 ASUR (1000), LYR (1000)
COMMON /SCALE/ RHOSTR, TRSTR, PRSTR, RBRSTR, VOLSTR, SURSTR, ELSTR
1 , SMSSCL, CVRSCL, SERSCL, TIMSCL
C*****
C****
TIMP = TIME (NT)*TIMSCL
PCHP = PCH (NT)*PRSTR
TCHP = TCH (NT)*TRSTR
TPMRP = TPMR (NT)*SMSSCL
RBRP = RBR (NT)*RBRSTR
ASURP = ASUR (NT)*SURSTR
DEPTHP = DPTHB (NT)*ELSTR
A3 (100) = NT
C
WRITE (11, 6000) NT, LYR (NT), TIMP, PCHP, TPMRP, RBRP
1 , ASURP, TCHP, DEPTH
6000 FORMAT(1X,'N=',I5,' LAYER=',I2,7E16.8)
C
RETURN
END
C*****
C***** SUBROUTINE CLEAR *****
C*****
SUBROUTINE CLEAR
CHARACTER ST*4
DATA ST/' 2J'/
WRITE (*, 6000) ST
6000 FORMAT (1X,A4)
RETURN
END
C*****
SUBROUTINE DEPTH (XMAS, J, IFLAG, DB, SA)
C*****
C Version: 3.0, January 1992
C Written by: William Oberle, U.S. Army Research Laboratory
C This program will determine the depth burned on a grain and
C compute the necessary integrals for the computation.
C
C XMAS: total mass of all grains remaining in the chamber in grams
C IFLAG: flag for desired information; 1 - compute integrals & Surf A
C 2 - compute surf Area only
C J: layer burning
C DB: depth burnt in cm
C SA: total burning surface area (all grains) in cm^2
C*****
C*****

```

```

COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
COMMON /CONTRL/ NTL, CHVOL, HLFRAC, DT, NMAX, CONVRG, ICTYP
COMMON /METH/ IMOD
PARAMETER (PI = 3.141593)

```

```

C*****
C***** TEST FOR NEGATIVE MASS *****
C*****
IF (XMAS .LE. 0.0) THEN
  DB = A3 (1)
  SA = 0.0
  RETURN
END IF
C*****
C***** DETERMINE MASS PER GRAIN & GRAIN TYPE *****
C*****
XMASG = XMAS/A3 (46)
ITYPE = INT (A3 (45) + .5)
C*****
C***** ISRS = 1 MEANS USE CONSTANT VALUES OTHERWISE *****
C***** ISRS = 2 MEANS USE CONSTANT VALUES FOR LAYERS *****
C***** WILL USE INTEGRALS TO PRODUCE THERMO PROPERTIES *****
C*****
ISRS = INT (A3 (31) + .5)
C*****
C***** DETERMINE MASS BURNED IN LAYER (ONE GRAIN) *****
C*****
NL = INT (A3 (4) + .5)
SUM = 0.0
DO 1000 I = J, NL
  SUM = SUM + PX (4, I, 3)
1000 CONTINUE
XMB = SUM - XMASG
C*****
C***** CHECK IF ON BOUNDARY OF LAYER *****
C*****
IF (ABS (XMB/PX (4, J, 3)) .LE. 1.0E-3) THEN
  DB = PX (1, J, 1)
  CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)
  SA = SFAREA*A3 (46)
  PX (4, J, 5) = 0.0
  PX (1, J, 5) = PX (1, J, 1)
  PX (6, J, 5) = 0.0
  PX (8, J, 5) = 0.0
  PX (9, J, 5) = 0.0
  PX (10, J, 5) = 0.0
  PX (11, J, 5) = 0.0
  RETURN
END IF

```

```

C*****
C***** FIRST IF THE PROPERTIES ARE CONSTANT *****
C***** AND WE HAVE A SINGLE GRAIN *****
C*****
IF (ISRS .EQ. 1) THEN
C*****
C***** Need to determine the depth burnt *****
C*****
C***** GVOL IS THE REMAINING VOLUME PER GRAIN *****
C*****
      GVOL = XMASG/PX (4, 1, 1)
      IF (GVOL .LE. 0.0) THEN
        WRITE (*, *) 'Grain Volume Zero in Depth'
        PAUSE
        RETURN
      END IF
C*****
C***** NOW WE DO THE SPHERE, THE DEPTH CAN BE SOLVED DIRECTLY *****
C*****
      IF (ITYPE .EQ. 1) THEN
        DB = A3 (8)/2. - (.75*GVOL/PI)**(1./3.)
        R = A3 (8)/2. - DB
        SA = 4.*PI*R*R*A3 (46)
        GO TO 1010
      END IF
C*****
C***** THE CORD WITH INHIBITED ENDS CAN ALSO BE SOLVED DIRECTLY *****
C*****
      IF (ITYPE .EQ. 11) THEN
        DB = (A3 (8)/2.) - (GVOL/(PI*A3 (7)))**(.1/2.)
        R = A3 (8)/2. - DB
        SA = 2.*PI*R*A3 (7)*A3 (46)
        GO TO 1010
      END IF
C*****
C***** THE CIGARETTE GRAIN *****
C*****
      IF (ITYPE .EQ. 13) THEN
        DB = A3 (7) - (4./PI)*(GVOL/(A3 (8)*A3 (8)))
        SA = A3 (8)*A3 (8)*PI/4.*A3 (46)
        GO TO 1010
      END IF
C*****
C***** SANDWICH WITH INHIBITED SIDES *****
C*****
      IF (ITYPE .EQ. 12) THEN
        SA = 2.*A3 (7)*A3 (10)*A3 (46)
        DB = (A3 (8)/2.) - (GVOL/(2.*A3 (7)*A3 (10)))

```

```

        GO TO 1010
    END IF
C*****
C***** USING NEWTON-RAPHSON METHOD *****
C***** NOT OBTAINING UNIQUE VALUES WILL NOT USE *****
C*****
    IF (IMOD .EQ. 2) THEN
        WRITE (*, *) 'Using Newton-Raphson'
        X2 = PX (1, 1, 5)
        CALL FORMT (ITYPE, SFAREA, VOLUNB, X2)
        VOL2 = VOLUNB
        PERC = (VOL2 - GVOL)/GVOL
        IF (ABS (PERC) .LE. CONVRG) THEN
            DB = X2
            SF = SFAREA*A3 (46)
            GO TO 1010
        END IF
        XYZST = (A3 (1) - X1)/100.
        X3 = X2 + XYZST
        CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)
        VOL3 = VOLUNB
        PERC = (VOL3 - GVOL)/GVOL
        IF (ABS (PERC) .LE. CONVRG) THEN
            DB = X3
            SF = SFAREA*A3 (46)
            GO TO 1010
        END IF
1020    CONTINUE
        X1 = X2
        X2 = X3
        VOL1 = VOL2
        VOL2 = VOL3
        X3 = (GVOL - VOL1)*(X2 - X1)/(VOL2 - VOL1) + X1
        CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)
        VOL3 = VOLUNB
        DELTAV = ABS (VOLUNB - GVOL)/GVOL
        IF (DELTAV .LE. CONVRG) THEN
            DB = X3
            SA = SFAREA*A3 (46)
            GO TO 1010
        END IF
        GO TO 1020
    END IF
C*****
C***** REMAINING GRAINS REQUIRE SEARCH *****
C*****
C WILL USE A BISECTION METHOD TO FIND THE DEPTH BURNT, CONSIDER THE
C FUNCTION WHICH IS THE DIFFERENCE BETWEEN THE REMAINING GRAIN

```

```

C VOLUME AND THE COMPUTED GRAIN VOLUME FOR A GIVEN DEPTH BURNT
C WE ARE LOOKING FOR THE ROOT OF THIS FUNCTION, IT MUST HAVE A
C UNIQUE SOLUTION BETWEEN ZERO AND MAX DEPTH BURNED, A3(1)
C*****
C THE THREE X VALUES USED IN THE METHOD WILL BE X1,X2,X3 IN THAT ORDER
C*****
C***** FIRST THE LOWER LIMIT FOR THE SEARCH *****
C*****
      WRITE (*, *) 'Using Bisection'
      IF (PX (1, 1, 5) .GT. 0.0) THEN
        X1 = PX (1, 1, 5)
      ELSE
        X1 = 0.0
      END IF
C*****
C***** TEST TO SEE IF PROPER LOWER VALUE *****
C***** GVOL-VOLUNB LESS THAN ZERO *****
C*****
      XYZST = (A3 (1) - X1)/10000.
      XYZST = ABS (XYZST)
1030  CONTINUE
      CALL FORMT (ITYPE, SFAREA, VOLUNB, X1)
      FUNCT = GVOL - VOLUNB
      IF (FUNCT .GE. 0.0) THEN
        X1 = X1 - XYZST
        IF (X1 .LE. 0.0) THEN
          X1 = 0.0
          CALL FORMT (ITYPE, SFAREA, VOLUNB, X1)
          GO TO 1040
        END IF
        GO TO 1030
      END IF
C*****
C***** NEXT THE UPPER LIMIT FOR THE BRACKET IS FOUND *****
C*****
1040  CONTINUE
      JSTOP = 0
      XYZST = (A3 (1) - X1)/10000.
1050  CONTINUE
      JSTOP = JSTOP + 1
      IF (JSTOP .EQ. 10001) THEN
        X3 = A3 (1)
        CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)
        GO TO 1060
      END IF
C  X3=X1+JSTOP*XYZST
      X3 = X1 + XYZST
      CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)

```

```

C*****
C***** SEE IF WE HAVE A MATCH *****
C*****
    FUNCT = GVOL - VOLUNB
    IF (ABS (FUNCT) .LT. .000001) THEN
        DB = X3
        SA = SFAREA*A3 (46)
        GO TO 1010
    END IF
    IF (FUNCT .LE. 0.0) THEN
        X1 = X3
        VOL1 = VOL3
        GO TO 1050
    END IF
    VOL3 = VOLUNB
1060  CONTINUE
C*****
C***** BRACKET HAS BEEN FOUND *****
C*****
C*****
C***** NOW WE START THE BISECTION METHOD *****
C*****
C***** FIRST THE MIDPT IS COMPUTED *****
C*****
1070  CONTINUE
    X2 = (X1 + X3)/2.
    CALL FORMT (ITYPE, SFAREA, VOLUNB, X2)
    F2 = GVOL - VOLUNB
    PERC = F2/GVOL
    IF ((ABS (PERC) .LT. CONVRG) .OR. (ABS (X3 - X1) .LT. .000001)) THEN
        DB = X2
        SA = SFAREA*A3 (46)
        GO TO 1010
    END IF
    IF (F2 .GT. 0.0) THEN
        X3 = X2
    ELSE
        X1 = X2
    END IF
    GO TO 1070
C*****
C***** ALL DEPTH BURNED AND SURFACE ARE FOUND *****
C*****
1010  CONTINUE
    PX (1, 1, 5) = DB
    IF (IFLAG .EQ. 2) THEN
        RETURN

```

```

        ELSE
            DO 1080 ICH = 2, 11
                PX (ICH, 1, 5) = PX (ICH, 1, 1)
1080     CONTINUE
            RETURN
        END IF
    END IF

C*****
C***** NOW CONSTANT PROPERTIES WITH *****
C***** SEVERAL LAYERS - NOT DETERRED *****
C*****

    IF (ISRS .EQ. 2) THEN
        GO TO 1090
1100     CONTINUE
        PX (1, J, 5) = DB
        IF (IFLAG .EQ. 2) THEN
            RETURN
        ELSE
            DO 1110 ICH = 2, 11
                PX (ICH, J, 5) = PX (ICH, J, 1)
1110     CONTINUE
            RETURN
        END IF

C***** NEED VOLUME OF LAYERS BELOW CURRENT LAYER *****
1090     CONTINUE
        WRITE (*, *) 'Using Bisection; Layered Grain'
        GGVOL = 0.0
        DO 1120 I = J + 1, NL
            GGVOL = GGVOL + PX (I, 1, 3)
1120     CONTINUE
C***** TARGET IS UNBURNED VOLUME IN LAYER *****
        TARGET = (PX (4, J, 3) - XMB)/PX (4, J, 1)
C***** NEED BRACKET FOR SEARCH, BISECTION METHOD *****
        IF (PX (1, J, 5) .GT. 0.0) THEN
            X1 = PX (1, J, 5)
        ELSE
            X1 = PX (1, J, 1)
        END IF

C*****
C***** TEST TO SEE IF PROPER LOWER VALUE *****
C***** TARGET-TESTV LESS THAN ZERO *****
C*****

        IF (J .EQ. NL) THEN
            ENDL = A3 (1)
        ELSE
            ENDL = PX (1, J + 1, 1)
        END IF
        XYZST = (ENDL - X1)/10000.

```

```

      XYZST = ABS (XYZST)
1130  CONTINUE
      CALL FORMT (ITYPE, SFAREA, VOLUNB, X1)
      TESTV = VOLUNB - GGVOL
      FUNCT = TARGET - TESTV
      IF (FUNCT .GE. 0.0) THEN
        X1 = X1 - XYZST
        IF (X1 .LT. PX (1, J, 1)) THEN
          X1 = PX (1, J, 1)
          GO TO 1140
        END IF
        GO TO 1130
      END IF
C*****
C***** NEXT THE UPPER LIMIT FOR THE BRACKET IS FOUND *****
C*****
1140  CONTINUE
      JSTOP = 0
1150  CONTINUE
      JSTOP = JSTOP + 1
      IF (JSTOP .EQ. 10001) THEN
        IF (J .EQ. NL) THEN
          X3 = A3 (1)
        ELSE
          X3 = PX (1, J + 1, 1)
        END IF
        CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)
        TESTV = VOLUNB - GGVOL
        PERC = (TARGET - TESTV)/PX (8, J, 3)
        IF (ABS (PERC) .LE. CONVRG) THEN
          SA = SFAREA*A3 (46)
          DB = X3
          GO TO 1100
        END IF
        GO TO 1160
      END IF
      X3 = X1 + XYZST
      CALL FORMT (ITYPE, SFAREA, VOLUNB, X3)
C*****
C***** SEE IF WE HAVE A MATCH *****
C*****
      TESTV = VOLUNB - GGVOL
      PERC = (TARGET - TESTV)/PX (8, J, 3)
      IF (ABS (PERC) .LT. CONVRG) THEN
        DB = X3
        SA = SFAREA*A3 (46)
        GO TO 1100
      END IF

```

```

    FUNCT = TARGET - TESTV
    IF (FUNCT .LE. 0.0) THEN
        X1 = X3
        GO TO 1150
    END IF
1160  CONTINUE
C*****
C***** BRACKET HAS BEEN FOUND *****
C*****
C***** NOW WE START THE BISECTION METHOD *****
C*****
C***** FIRST THE MIDPT IS COMPUTED *****
C*****
1170  CONTINUE
    X2 = (X1 + X3)/2.
    CALL FORMT (ITYPE, SFAREA, VOLUNB, X2)
    TESTV = VOLUNB - GGVOL
    F2 = TARGET - TESTV
    PERC = F2/PX (8, J, 3)
    IF ((ABS (PERC) .LT. CONVRG) .OR. (ABS (X3 - X1) .LT. .000001)) THEN
        DB = X2
        SA = SFAREA*A3 (46)
        GO TO 1100
    END IF
    IF (F2 .GT. 0.0) THEN
        X3 = X2
    ELSE
        X1 = X2
    END IF
    GO TO 1170
END IF
C*****
C***** LOWER LIMIT OF INTEGRATION *****
C*****
    XS = PX (1, J, 1)
C*****
C Need to determine an interval for the upper limit of integration
C will then use a bisection method to determine the depth burned.
C*****
C***** FIRST THE MAXIMUM POSSIBLE UPPER LIMIT IS DETERMINED *****
C*****
    IF (J .EQ. NL) THEN
        XMAX = A3 (1)
    ELSE
        XMAX = PX (1, J + 1, 1)
    END IF

```

```

C*****
C***** NEXT THE INTERVAL FOR THE UPPER LIMIT IS DETERMINED *****
C*****
C***** CASE I: WE HAVE INTEGRATED ON THIS LAYER BEFORE *****
C*****

IF (PX (4, J, 5) .GT. 0.0) THEN
  IF (PX (4, J, 5) .EQ. XMB) THEN
    DB = PX (1, J, 5)
    CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)
    SA = SFAREA*A3 (46)
    IF (IFLAG .EQ. 2) THEN
      RETURN
    ELSE
      CALL IGRALS (XS, J, DB)
      RETURN
    END IF
  ELSE IF (PX (4, J, 5) .GT. XMB) THEN
    XL = PX (1, J, 5)
    VAL = PX (4, J, 5)
    Q = (XL - PX (1, J, 1))/10.
    DO 1180 II = 1, 10
      XR = XL - Q
      IF (XR .LE. XS) THEN
        XR = XS
        VAR = 0.0
        GO TO 1190
      END IF
      XDEL = XR - XS
      XSTEP = XDEL/300.
      PDEL = PX (4, J, 2) - PX (4, J, 1)
      XDEL1 = XMAX - XS
      SUM = 0.0
      DO 1200 I = 1, 301
        XI = XS + (I - 1)*XSTEP
        CALL FORMT (ITYPE, ASURF, VOLUNB, XI)
        FCN = ((PDEL/XDEL1)*(XI - XS) + PX (4, J, 1))*ASURF
        IF ((I .EQ. 1) .OR. (I .EQ. 301)) THEN
          SUM = SUM + FCN
        ELSE
          SUM = SUM + 2.*FCN
        END IF
      CONTINUE
      SUM = SUM*(XDEL/600.)
      IF (SUM .EQ. XMB) THEN
        DB = XR
        PX (1, J, 5) = DB
        PX (4, J, 5) = SUM
        CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)

```

```

        SA = SFAREA*A3 (46)
        IF (IFLAG .EQ. 2) THEN
            RETURN
        ELSE
            CALL IGRALS (XS, J, DB)
            RETURN
        END IF
        ELSE IF (SUM .LT. XMB) THEN
            VAR = SUM
            GO TO 1190
        ELSE
            XL = XR
            VAL = SUM
            END IF
1180    CONTINUE
        ELSE
            XR = PX (1, J, 5)
            VAR = PX (4, J, 5)
            Q = (XMAX - XR)/10.
            DO 1210 II = 1, 10
                XL = XR + Q
                IF (XL .GT. XMAX) THEN
                    XL = XMAX
                    VAL = PX (4, J, 3)
                    GO TO 1190
                END IF
                XDEL = XL - XS
                XSTEP = XDEL/300.
                PDEL = PX (4, J, 2) - PX (4, J, 1)
                XDEL1 = XMAX - XS
                SUM = 0.0
                DO 1220 I = 1, 301
                    XI = XS + (I - 1)*XSTEP
                    CALL FORMT (ITYPE, ASURF, VOLUNB, XI)
                    FCN = ((PDEL/XDEL1)*(XI - XS) + PX (4, J, 1))*ASURF
                    IF ((I .EQ. 1) .OR. (I .EQ. 301)) THEN
                        SUM = SUM + FCN
                    ELSE
                        SUM = SUM + 2.*FCN
                    END IF
                END IF
1220    CONTINUE
            SUM = SUM*(XDEL/600.)
            IF (SUM .EQ. XMB) THEN
                DB = XL
                PX (1, J, 5) = DB
                PX (4, J, 5) = SUM
                CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)
                SA = SFAREA*A3 (46)

```

```

        IF (IFLAG .EQ. 2) THEN
            RETURN
        ELSE
            CALL IGRALS (XS, J, DB)
            RETURN
        END IF
    ELSE IF (SUM .GT. XMB) THEN
        VAL = SUM
        GO TO 1190
    ELSE
        XR = XL
        VAR = SUM
    END IF
1210    CONTINUE
    END IF
ELSE
C*****
C***** NO PREVIOUS INFORMATION *****
C*****
        XR = XS
        VAR = 0.0
        XL = XMAX
        VAL = PX (4, J, 3)
    END IF
1190 CONTINUE
C*****
C***** MIDPOINT USED TO FIND DEPTH *****
C*****
C***** TEST TO END THE SEARCH: DELTA X < .00001 *****
C*****
        XDELTA = XL - XR
        IF (XDELTA .LT. .00001) THEN
            DB = (XL + XR)/2.
            PX (1, J, 5) = DB
            PX (4, J, 5) = (VAL + VAR)/2.
            CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)
            SA = SFAREA*A3 (46)
            IF (IFLAG .EQ. 2) THEN
                RETURN
            ELSE
                CALL IGRALS (XS, J, DB)
                RETURN
            END IF
        END IF
C*****
C***** INTEGRATION IS PERFORMED FOR MIDPOINT *****
C*****
        XTRY = (XL + XR)/2.

```

```

XDEL = XTRY - XS
XSTEP = XDEL/300.
PDEL = PX (4, J, 2) - PX (4, J, 1)
XDEL1 = XMAX - XS
SUM = 0.0
DO 1230 I = 1, 301
  XI = XS + (I - 1)*XSTEP
  CALL FORMT (ITYPE, ASURF, VOLUNB, XI)
  FCN = ((PDEL/XDEL1)*(XI - XS) + PX (4, J, 1))*ASURF
  IF ((I .EQ. 1) .OR. (I .EQ. 301)) THEN
    SUM = SUM + FCN
  ELSE
    SUM = SUM + 2.*FCN
  END IF
1230 CONTINUE
SUM = SUM*(XDEL/600.)
PERC = ABS ((SUM - XMB)/XMB)
IF (PERC .LT. CONVRG) THEN
  DB = XTRY
  PX (1, J, 5) = DB
  PX (4, J, 5) = SUM
  CALL FORMT (ITYPE, SFAREA, VOLUNB, DB)
  SA = SFAREA*A3 (46)
  IF (IFLAG .EQ. 2) THEN
    RETURN
  ELSE
    CALL IGRALS (XS, J, DB)
    RETURN
  END IF
ELSE IF (SUM .LT. XMB) THEN
  XR = XTRY
  VAR = SUM
  GO TO 1190
ELSE
  XL = XTRY
  VAL = SUM
  GO TO 1190
END IF
RETURN
END
C*****
C*
C***** SUBROUTINE FORMT *****
C*
C*****
C*
C   ICODE: code for type of grain
C   R : burn depth

```

C GL: unburned grain length
 C D : unburned outer diameter
 C PD: unburned perforation diameter
 C WI, WM, WO: inner, middle and outer webs respectively
 C

C Output:

C SFAREA: surface area
 C FRCSFA: surface area/initial surface area
 C VOLUNB: unburned volume
 C VOLBRN: burned volume
 C FRCBRN: burned volume/initial volume
 C VOLMAO: unburned volume of outer layer
 C VOLMBO: unburned volume of inner layer
 C VOLABR: burned volume of outer layer
 C VOLBBR: burned volume of inner layer
 C

C*****

SUBROUTINE FORMT (ICODE, SFAREA, VOLUNB, R)
 COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
 DIMENSION S7 (4), S19 (3, 4)
 DATA RT/1.732050808/, PI3/1.047197551/, PI/3.141592654/

C*****

C***** SET GRAIN GEOMETRY *****

C*****

GL = A3 (7)
 D = A3 (8)
 PD = A3 (9)
 WI = A3 (10)
 WM = A3 (11)
 WO = A3 (12)

C*****

C***** Set U = 2*(depth burned) and branch to grain type *****

C*****

U = 2.0*R

C*****

C***** CIGARETTE GRAIN *****

C*****

IF (ICODE .EQ. 13) THEN
 SFAREA = PI*D*D/4.
 VOLUNB = (GL - R)*SFAREA
 RETURN
 END IF

C*****

C***** ALL OTHER GRAINS ARE HANDLED *****

C*****

GO TO (1000, 1010, 1020, 1030, 1040, 1050, 1060, 1070,
 1 1080, 1090, 1100, 1110), ICODE

C*****

```

C***** CODE 1: 7-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns.
C*****
1050 CONTINUE
    D = 3.0*PD + 2.0*(WI + WO)
    E0 = PI*(D**2 - 7.0*PD**2)/4.0
    S0 = PI*(D + 7.0*PD)*GL + 2.0*E0
    V0 = E0*GL
    WW = WI + PD
    DO 1120 K = 1, 3
        S7 (K) = WW
1120 CONTINUE
    WEBC = AMIN1 (WO, WI, GL)
C*****
C*** This part does the calculations for the burning grain.
C*****
    GRL = AMAX1 (GL - U, 0.0)
    OD = D - U
    PRFD = PD + U
    IF (U .GT. WEBC) GO TO 1130
    E = PI*(OD**2 - 7.0*PRFD**2)/4.0
    SFAREA = PI*(OD + 7.0*PRFD)*GRL + 2.0*E
    FRCSFA = SFAREA/S0
    VOLUNB = E*GRL
    VOLBRN = V0 - VOLUNB
    FRCBRN = VOLBRN/V0
    RETURN
C*****
C*** This part does the calculations for when the grain slivers. *****
C*****
1130 CONTINUE
    CALL GENIS (S7, PRFD, GRL, SF1, GV1)
    CALL GENOS (S7, PRFD, GRL, 0.5*OD, SF2, GV2)
    SFAREA = 6.0*(SF1 + SF2)
    FRCSFA = SFAREA/S0
    VOLUNB = 6.0*(GV1 + GV2)
    VOLBRN = V0 - VOLUNB
    FRCBRN = VOLBRN/V0
    RETURN
C*****
C***** CODE 2: 1-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns.
C*****
1030 CONTINUE
    D = PD + 2.0*WI
    E0 = PI*(D**2 - PD**2)/4.0

```

```

S0 = PI*(D + PD)*GL + 2.0*E0
V0 = E0*GL
WEBC = AMIN1 (GL, WI)
C*****
C***** This part does the calculations for the burning grain. *****
C*****
IF (U .GE. WEBC) THEN
  GRL = 0.0
  E = 0.0
  GO TO 1140
END IF
GRL = GL - U
OD = D - U
PRFD = PD + U
E = PI*(OD**2 - PRFD**2)/4.0
1140 CONTINUE
SFAREA = PI*(OD + PRFD)*GRL + 2.0*E
FRCSFA = SFAREA/S0
VOLUNB = E*GRL
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 3: CORD GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns. *****
C*****
1010 CONTINUE
S0 = GL*PI*D + PI*D**2/2.0
V0 = GL*PI*D**2/4.0
C*** This part does the calculations for the burning grain.
GRL = AMAX1 (GL - U, 0.0)
OD = AMAX1 (D - U, 0.0)
E = PI*OD**2/4.0
SFAREA = PI*OD*GRL + 2.0*E
FRCSFA = SFAREA/S0
VOLUNB = GRL*PI*OD**2/4.0
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 4: RECTANGULAR STRIP GRAIN *****
C*****
1020 CONTINUE
S0 = 2.0*(GL*D + D*WI + GL*WI)
V0 = GL*D*WI
GRL = AMAX1 (GL - U, 0.0)
DS = AMAX1 (D - U, 0.0)

```

```

WIS = AMAX1 (WI - U, 0.0)
SFAREA = 2.0*(GRL*DS + DS*WIS + WIS*GRL)
FRCSFA = SFAREA/S0
VOLUNB = GRL*DS*WIS
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 5: SPHERICAL GRAIN *****
C*****
1000 CONTINUE
S0 = PI*D**2
V0 = PI*D**3/6.0
OD = AMAX1 (D - U, 0.0)
SFAREA = PI*OD**2
FRCSFA = SFAREA/S0
VOLUNB = PI*OD**3/6.0
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 6: SLOTTED-TUBE GRAIN *****
C*****
C*** This part does the calculations before the grain burns. *****
C*****
1040 CONTINUE
SLOT = 0.5*WM
SO = 0.5*D
SI = 0.5*PD
THETA = ASIN (SLOT/SO)
ALPHA = ASIN (SLOT/SI)
E0 = (PI - ALPHA)*(SO**2 - SI**2) + (SO - SI)**2*ALPHA
S0 = 2.0*((PI - ALPHA)*SI + (PI - THETA)*SO + (SO*COS (THETA)
1 - SI*COS (ALPHA)))*GL + 2.0*E0
V0 = GL*E0
WI = SO - SI
WEBC = AMIN1 (GL, WI)
C*****
C*** This part does the calculations for the burning grain. *****
C*****
IF (U .GE. WEBC) THEN
GRL = 0.0
E = 0.0
GO TO 1150
END IF
SLOT = 0.5*(WM + U)
SO = 0.5*(D - U)
SI = 0.5*(PD + U)

```

```

GRL = GL - U
THETA = ASIN (SLOT/SO)
ALPHA = ASIN (SLOT/SI)
E = (PI - ALPHA)*(SO**2 - SI**2) + (SO - SI)**2*ALPHA
1150 CONTINUE
SFAREA = 2.0*((PI - ALPHA)*SI + (PI - THETA)*SO + (SO*COS (THETA)
1 - SI*COS (ALPHA)))*GRL + 2.0*E
FRCSFA = SFAREA/SO
VOLUNB = E*GRL
VOLBRN = V0 - VOLUNB
FRCBRN = VOLBRN/V0
RETURN
C*****
C***** CODE 7: ROUND-HEX 37-PERF GRAIN *****
C*****
1090 CONTINUE
SO = 18.
SI = 54.
NPERF = 37
D = 7.0*PD + 6.0*WI + 2.0*WO
GO TO 1160
C*****
C***** CODE 8: ROUND-HEX 19-PERF GRAIN *****
C*****
1080 CONTINUE
SO = 12.0
SI = 24.0
NPERF = 19
D = 5.0*PD + 4.0*WI + 2.0*WO
C*****
C***** CALCULATIONS FOR CODES 7,8,10 *****
C*****
1160 CONTINUE
WW = WI + PD
WW2 = WW**2
PRFD = PD + U
PRFD2 = PRFD**2
GRL = AMAX1 (GL - U, 0.0)
E = 0.0
THETA = 2.0*ACOS (AMIN1 (WW/PRFD, 1.0))
ALPHA = ACOS (AMIN1 ((2.0*WO + PD - U)/PRFD, 1.0))
IF (U .LT. WO) E = 0.25*PI*((2.0*WO + PD - U)**2 - PRFD2)
IF (THETA .GE. PI/3) GO TO 1170
E = E + SI*0.25*(WW2*RT - 1.5*PRFD2*(SIN (THETA) + PI/3 - THETA))
1170 CONTINUE
IF (ALPHA .GE. 0.5*(PI - THETA)) GO TO 1180
E = E + SO*0.125*(2.0*(2.0*WO + PD - U)*(2.0*WW - PRFD*SIN (ALPHA))
1 - PRFD2*(SIN (THETA) + PI - 2.0*ALPHA - THETA))

```

```

1180 CONTINUE
  IF (2.0*WO + PD .LT. WI) THEN
    WRITE (*, *) '*FORMT* BAD HEX PROP'
    PAUSE
    GO TO 1190
  END IF
  VOLUNB = E*GRL
C***** TEST TO SEE IF GRAIN CONSUMED *****
  IF (VOLUNB .LE. 0.0) THEN
    SFAREA = 0.0
    VOLUNB = 0.0
    GO TO 1190
  END IF
C*****NOW THE SURFACE AREA*****
  PH = D/2. - WO - PD/2.
  IF (U .EQ. 0.0) THEN
    SFAREA = 2.*E + GRL*PH*6. + NPERF*PI*PD*GRL + PI*GRL*(2*WO + PD)
    GO TO 1190
  END IF
C*****NO SLIVERING YET*****
  IF ((WO .GT. U) .AND. (WI .GT. U)) THEN
    SFAREA = 2.*E + NPERF*(PD + U)*GRL*PI + 6.*PH*
1    GRL + PI*(2.*WO + PD - U)*GRL
    GO TO 1190
  END IF
C***** NOW SLIVERING *****
  SFAREA = 2.*E
C***** FIRST THE INNER SLIVERS *****
  IF (THETA .GE. PI/3) THEN
    GO TO 1200
  ELSE
    SFAREA = SFAREA + 1.5*PRFD*GRL*(PI/3 - THETA)*SI
  END IF
1200 CONTINUE
C***** NOW THE OUTER SLIVERS & CORNERS *****
C***** CORNERS NOT CONSUMED *****
  IF (WO .GT. U) THEN
    SFAREA = SFAREA + PI*(PD + 2.*WO - U + PRFD)*GRL
  END IF
C*****NOW OUTER SLIVERS*****
  IF (ALPHA .LT. .5*(PI - THETA)) THEN
    SFAREA = SFAREA + (WW - PRFD*SIN (ALPHA))*GRL*SO +
1    PRFD*GRL*(PI/2. - ALPHA - THETA/2.)*SO
  END IF
1190 CONTINUE
  RETURN

```

```

C*****
C***** CODE 9: 19-PERF GRAIN *****
C*****
C*** This part calculates the conditions before the grain burns *****
C*****
1070 CONTINUE
    D = 5.0*PD + 2.0*(WI + WM + WO)
    E0 = PI*(D**2 - 19.0*PD**2)/4.0
    S0 = PI*(D + 19.0*PD)*GL + 2.0*E0
    V0 = E0*GL
    S19 (1, 1) = WI + PD
    S19 (2, 1) = S19 (1, 1)
    S19 (3, 1) = S19 (1, 1)
    S19 (1, 2) = 0.5*SQRT (3.0*(WM + PD)**2 + (WI + PD)**2)
    S19 (2, 2) = S19 (1, 2)
    S19 (3, 2) = S19 (1, 1)
    S19 (1, 3) = PD + 0.5*(WI + WM)
    S19 (2, 3) = S19 (1, 2)
    S19 (3, 3) = WM + PD
    S19 (1, 4) = S19 (1, 3)
    S19 (2, 4) = 2.0*S19 (1, 3)
    S19 (3, 4) = S19 (1, 3)*RT
    WEBC = AMIN1 (WO, WM, WI, S19 (1, 3) - PD, S19 (1, 2) - PD, GL)
C*****
C*** This part does the calculations for the burning grain. *****
C*****
    GRL = AMAX1 (GL - U, 0.0)
C*****
    OD = D - U
    PRFD = PD + U
    IF (U .GE. WEBC) GO TO 1210
    E = 0.25*PI*(OD**2 - 19.0*PRFD**2)
    SFAREA = PI*(OD + 19.0*PRFD)*GRL + 2.0*E
    FRCSFA = SFAREA/S0
    VOLUNB = E*GRL
    VOLBRN = V0 - VOLUNB
    FRCBRN = VOLBRN/V0
    RETURN
1210 CONTINUE
    SUMSA = 0.0
    SUMGV = 0.0
    DO 1220 K = 1, 2
        CALL GENIS (S19 (1, K), PRFD, GRL, SA, GV)
        SUMSA = SUMSA + 6.0*SA
        SUMGV = SUMGV + 6.0*GV
1220 CONTINUE
    CALL GENIS (S19 (1, 3), PRFD, GRL, SA, GV)
    SUMSA = SUMSA + 12.0*SA

```

```

SUMGV = SUMGV + 12.0*GV
CALL GENOS (S19 (1, 4), PRFD, GRL, .5*OD, SA, GV)
SUMSA = SUMSA + 12.0*SA
SUMGV = SUMGV + 12.0*GV
SFAREA = SUMSA
FRCSFA = SFAREA/SO
VOLUNB = SUMGV
VOLBRN = VO - VOLUNB
FRCBRN = VOLBRN/VO
RETURN
C*****
C***** CODE 10: ROUND-HEX 7-PERF GRAIN *****
C*****
1060 CONTINUE
    SO = 6.0
    SI = 6.0
    NPERF = 7
    D = 3.*PD + 2*WI + 2*WO
    GO TO 1160
C*****
C***** CODE 11: CORD WITH INHIBITED ENDS *****
C*****
C This routine will only calculate the surface area of the lateral surfa
Cces.
C It will not calculate the surface area of the inhibited ends. *****
C*****
1100 CONTINUE
    SO = GL*PI*D
    VO = GL*PI*D**2/4.0
C **** This part does the calculations for the burning grain
    OD = AMAX1 (D - U, 0.0)
    SFAREA = PI*OD*GL
    FRCSFA = SFAREA/SO
    VOLUNB = GL*PI*OD**2/4.0
    VOLBRN = VO - VOLUNB
    FRCBRN = VOLBRN/VO
    RETURN
C*****
C***** CODE 12: RECTANGULAR STRIP GRAIN WITH INHIBITED SIDES *****
C*****
C This routine will only calculate the surface area of the two burning s
Cides.
C It will not calculate the surface area of the inhibited sides.
C*****
1110 CONTINUE
    SO = 2.0*GL*WI
    VO = GL*D*WI
    VOLMAO = (D - WI)*GL*WO

```

```

VOLMBO = WI*GL*WO
C*****
C *** This part does the calculations for the burning grain
  DS = AMAX1 (D - U, 0.0)
  SFAREA = 2.0*GL*WI
  FRCSFA = SFAREA/SO
  VOLUNB = GL*DS*WI
  VOLBRN = VO - VOLUNB
  FRCBRN = VOLBRN/VO
  IF (DS .GE. WI) THEN
    VOLABR = VOLBRN
    VOLBBR = 0.
  ELSE
    VOLABR = VOLMAO
    VOLBBR = VOLBRN - VOLABR
  END IF
  RETURN
  END
C*****
C*****
C
C SUBROUTINE *GENIS*: calculate surface area and volume for a
C                      general inner sliver of a burning grain
C                      with length = GRL & perforation dia. = PRFD.
C
  SUBROUTINE GENIS (S, PRFD, GRL, SURF, VOL)
  DIMENSION S (3), A (4)
  DATA PI2/1.5707963/
C
C
C ***** : Store angles A1,A2,A3 and area of triangle
C           with sides S(1),S(2),S(3) into A(1)...A(4)
C
  A (1) = ACOS ((S (2)**2 + S (3)**2 - S (1)**2)/(2.0*S (2)*S (3)))
  A (2) = ACOS ((S (1)**2 + S (3)**2 - S (2)**2)/(2.0*S (1)*S (3)))
  A (3) = ACOS ((S (1)**2 + S (2)**2 - S (3)**2)/(2.0*S (1)*S (2)))
  A (4) = 0.5*S (1)*S (3)*SIN (A (2))
C
C ...check for error condition: find if triangle acceptable...
C
  J = 0
  DO 1000 I = 1, 3
    IF (A (I) .LT. 0.5*PI2) J = J + 1
  1000 CONTINUE
  IF (J .GT. 1) STOP ' GENIS ERROR'
C
C
C succeeding passes until burnout: find auxiliary angles

```

```

C
  TAU12 = ACOS (AMIN1 (1.0, S (3)/PRFD))
  TAU13 = ACOS (AMIN1 (1.0, S (2)/PRFD))
  TAU23 = ACOS (AMIN1 (1.0, S (1)/PRFD))
C
C ...and branch to 25 if sliver fails burnout criteria. If not
C   then sliver is burned and go to 30.
C
  IF (TAU12 + TAU13 + TAU23 .LT. PI2 .AND. GRL .GT. 0.0) THEN
    GO TO 1010
  ELSE
    GO TO 1020
  END IF
C
C
C sliver not burned out: determine end area, volume and surface area
C
1010 CONTINUE
  E = A (4) - 0.25*PRFD*(S (1)*SIN (TAU23) + S (2)*SIN (TAU13)
1   + S (3)*SIN (TAU12) + PRFD*(PI2 - TAU12 - TAU13 - TAU23))
C
  VOL = E*GRL
C
  SURF = 2.0*E + GRL*PRFD*(PI2 - TAU12 - TAU13 - TAU23)
C
C ...and RETURN
C
  RETURN
C
C sliver is burned out: return with zero volume and surface area.
C
1020 CONTINUE
  VOL = 0.0
  SURF = 0.0
  RETURN
  END
C*****
C SUBROUTINE "GENOS" : Calculates surface area and volume for a
C                       general outer sliver of a burning grain
C                       with length = GRL, radius = RAD, and
C                       perforation diameter = PRFD
C
C SUBROUTINE GENOS (S, PRFD, GRL, RAD, SURF, VOL)
C DIMENSION S (3), A (4)
C
C ***** Store angles A1,A2,A3 and area of triangle
C           with sides S(1),S(2),S(3) into A(1) ...A(4)
C

```

```

A (1) = ACOS ((S (2)**2 + S (3)**2 - S (1)**2)/(2.0*S (2)*S (3)))
A (2) = ACOS ((S (1)**2 + S (3)**2 - S (2)**2)/(2.0*S (1)*S (3)))
A (3) = ACOS ((S (1)**2 + S (2)**2 - S (3)**2)/(2.0*S (1)*S (2)))
A (4) = 0.5*S (1)*S (3)*SIN (A (2))
C
C
C succeeding passes until burnout: determine auxiliary angles
C
TAU1 = ACOS (AMIN1 (1., (S (2)**2 + RAD**2 - 0.25*
1 PRFD**2)/(2.*S (2)*RAD)))
TAU2 = ACOS (AMIN1 (1., (S (3)**2 + RAD**2 - 0.25*
1 PRFD**2)/(2.*S (3)*RAD)))
TAU3 = ACOS (AMAX1 (- 1.0, (S (2)**2 - RAD**2 + 0.25*
1 PRFD**2)/(S (2)*PRFD)))
TAU4 = ACOS (AMAX1 (- 1.0, (S (3)**2 - RAD**2 + 0.25*
1 PRFD**2)/(S (3)*PRFD)))
C
SIG = ACOS (AMIN1 (1.0, S (1)/PRFD))
C
C ...then check error conditions...
C
IF (TAU3 .LT. A (3) .OR. TAU4 .LT. A (2)) STOP ' *GENOS* ERROR'
C
C ...IF ok, check if sliver burned out. If not burned out go to 25.
C If burned out go to 30.
C
IF (TAU1 + TAU2 .LT. A (1) .AND. GRL .GT. 0.0) THEN
GO TO 1000
ELSE
GO TO 1010
END IF
C
C
C sliver not burned out: determine end area, volume and surface area.
C
1000 CONTINUE
E = 0.5*RAD*(S (2)*SIN (TAU1) + RAD*(A (1) - TAU1 - TAU2)
1 + S (3)*SIN (TAU2)) - A (4) - 0.25*PRFD*(S (1)*SIN (SIG)
2 + 0.5*PRFD*(TAU3 + TAU4 - 2.0*SIG - A (2) - A (3)))
C
VOL = E*GRL
C
SURF = 2.0*E + GRL*(RAD*(A (1) - TAU1 - TAU2) + 0.5*PRFD*(TAU3
1 + TAU4 - 2.0*SIG - A (2) - A (3)))
C
C .. and RETURN.
C
RETURN

```

```

C
C sliver is burned out: return with zero volume and surface area.
C
1010 CONTINUE
    VOL = 0.0
    SURF = 0.0
    RETURN
    END
C*****
C*****SUBROUTINE IGRALS *****
C*****
    SUBROUTINE IGRALS (XS, J, DB)
C*****
C Version 3.0, January 1992
C
C This subroutine is used to compute the integrals from beginning of
C the layer to the depth burned.
C
C*****
    COMMON /ARRAYS/ A3 (100), PX (11, 15, 5)
    ITYPE = INT (A3 (45) + .5)
    NL = INT (A3 (4) + .5)
    DO 1000 I = 2, 11
        PX (I, J, 5) = 0.0
    1000 CONTINUE
C*****
C***** FIRST THE LENGTH OF THE INTERVAL FOR PROPERTY IS DETERMINED ****
C*****
    IF (J .EQ. NL) THEN
        XMAX = A3 (1)
    ELSE
        XMAX = PX (1, J + 1, 1)
    END IF
    XDEL1 = XMAX - XS
C*****
C***** ALWAYS 300 SUBDIVISIONS *****
C*****
    XDEL = DB - XS
    XSTEP = XDEL/300.
    DO 1010 I = 1, 301
        XI = XS + (I - 1)*XSTEP
C*****
        CALL FORMT (ITYPE, ASURF, VOLUNB, XI)
C*****
    DO 1020 K = 2, 11
        PDEL = PX (K, J, 2) - PX (K, J, 1)
C*****
C***** INTEGRATION IS PERFORMED *****

```

```

C*****
      FCN = ((PDEL/XDEL1)*(XI - XS) + PX (K, J, 1))*ASURF
      IF ((I .EQ. 1) .OR. (I .EQ. 301)) THEN
        PX (K, J, 5) = PX (K, J, 5) + FCN
      ELSE
        PX (K, J, 5) = PX (K, J, 5) + 2.*FCN
      END IF
1020  CONTINUE
1010 CONTINUE
C*****
C***** NOW THE INTEGRAL VALUE IS DETERMINED *****
C*****
      DO 1030 K = 2, 11
        PX (K, J, 5) = PX (K, J, 5)*(XDEL/600.)
1030 CONTINUE
1040 CONTINUE
      RETURN
      END

```

APPENDIX L:
LISTING - PROGRAM MKOUT.FOR

INTENTIONALLY LEFT BLANK.

PROGRAM MKOUT

```

C*****
C Version: 3.0, January 1992; Last Modified 12/31/91
C                               2/28/92, cleanup
C
C Written by: William Oberle, U.S. Army Research Laboratory
C
C*****
  CHARACTER*20 FNAME, A1 (20), A2 (6)*80, ITEMP*
  1  60, LINE*132, WORD1*9,
  2  WORD2*10, CHART*2, FGRF*20, NAMY, APL (30, 60)*1, A*1, NAME1
  DIMENSION A3 (100), FRAT (1024, 3), XLFRAT (1024, 3), P (11, 15, 5)
  DIMENSION TAL (15, 2)
  DATA APL/1800*' '/
  CALL CLEAR
C*****
C***** Reading the information file *****
C*****
  WRITE (*, 6000)
  6000 FORMAT(///,10X,'Enter the file name.')
  READ (*, 5000) FNAME
  5000 FORMAT(A20)
  NAME1 = FNAME
  1000 CONTINUE
  DO 1010 I = 1, 17
    A = NAME1 (I:I)
    IF (A .EQ. '.') THEN
      CALL CLEAR
      WRITE (*, *) 'The file name entered has an extension which is'
      WRITE (*, *) 'not allowed. Please enter a new name without '
      WRITE (*, *) 'an extension.'
      READ (*, 5000) NAME1
      GO TO 1000
    END IF
    IF (A .EQ. ' ') THEN
      NAME1 (I:I + 3) = '.inf'
      GO TO 1020
    END IF
  1010 CONTINUE
  1020 CONTINUE
  FNAME = NAME1
  OPEN (UNIT = 9, FILE = FNAME)
  REWIND (UNIT = 9)
C*****

```

```

DO 1030 I = 1, 6
  READ (9, 5010) A2 (I)
1030 CONTINUE
5010 FORMAT(A80)
DO 1040 I = 1, 20
  READ (9, 5000) A1 (I)
1040 CONTINUE
DO 1050 I = 1, 100
  READ (9, *) A3 (I)
1050 CONTINUE
DO 1060 I = 1, 11
  DO 1070 J = 1, 15
    DO 1080 K = 1, 5
      READ (9, *) P (I, J, K)
1080 CONTINUE
1070 CONTINUE
1060 CONTINUE
CLOSE (UNIT = 9)
C*****
C***** INFORMATION ABOUT THE FIRING IS OBTAINED *****
C*****
CALL CLEAR
WRITE (*, *) ' '
WRITE (*, *) ' '
WRITE (*, *) ' Enter the date of the closed chamber firing.'
READ (*, 5000) A1 (20)
WRITE (*, *) 'Enter any comments concerning the closed chamber'
WRITE (*, *) 'firing. Three lines are required and the lines'
WRITE (*, *) 'may be blank. Include operators.'
READ (*, 5010) A2 (1)
READ (*, 5010) A2 (2)
READ (*, 5010) A2 (3)
CALL CLEAR
WRITE (*, *) 'Enter any comments concerning the data reduction.'
WRITE (*, *) 'Three lines are required and the lines may be blank.'
WRITE (*, *) 'Include date and operator.'
READ (*, 5010) A2 (4)
READ (*, 5010) A2 (5)
READ (*, 5010) A2 (6)
C*****
C***** INFORMATION IS REWRITTEN *****
C*****
OPEN (UNIT = 7, FILE = FNAME)
REWIND (UNIT = 7)
DO 1090 I = 1, 6
  WRITE (7, 5010) A2 (I)
1090 CONTINUE

```

```

DO 1100 I = 1, 20
  WRITE (7, 5000) A1 (I)
1100 CONTINUE
  DO 1110 I = 1, 100
    WRITE (7, *) A3 (I)
1110 CONTINUE
    DO 1120 I = 1, 11
      DO 1130 J = 1, 15
        DO 1140 K = 1, 5
          WRITE (7, *) P (I, J, K)
1140   CONTINUE
1130   CONTINUE
1120   CONTINUE
      CLOSE (UNIT = 7)
C*****
C***** OUTPUT FORM PROGRAM *****
C***** THIS PROGRAM WILL OUTPUT THE INFORMATION FROM A BRLCB *****
C***** ANALYSIS. *****
C*****
CALL CLEAR
NAMY = A1 (10)
WRITE (*, 6010)
6010 FORMAT(//,10X,'The program will produce both hard copy to',
1/,10X,'the printer and a disk copy of the output. Select the',
2/,10X,'option.',
3//,15X,'1. Hard copy to printer only.',
4/,15X,'2. Disk copy only.',
5/,15X,'3. Both disk and hard copy.',
6//,10X,'Enter your choice (1-3).')
READ (*, *) IOUTCH
IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
  FGRF = 'LPT1'
  WRITE (*, 6020) FGRF
6020 FORMAT(//,10X,'The current printer is: ',A20,
1/,10X,'1. Accept Current Printer',/,10X,'2. Enter new choice',/
2/,15X,'Enter choice',/)
  READ (*, *) IPRTE
  IF (IPRTE .EQ. 2) THEN
    WRITE (*, *) 'Enter new printer name.'
    READ (*, 5020) FGRF
5020 FORMAT(A20)
  END IF
  OPEN (UNIT = 7, FILE = FGRF)
END IF
IF (IOUTCH .NE. 1) THEN
  WRITE (*, 6030) A1 (10)
6030 FORMAT(//,10X,'Enter the file name for the output.',
1/,10X,'WARNING THIS FILE CANNOT BE NAMED:/',

```

```

2/,15X,A20)
  READ (*, 5000) FGRF
  NAMEY = FGRF
  OPEN (UNIT = 2, FILE = FGRF)
  REWIND (UNIT = 2)
END IF
C*****
C***** OPENING GRAPHICS FILE *****
C*****
  OPEN (UNIT = 9, FILE = A1 (17))
  REWIND (UNIT = 9)
C*****
C***** HEADINGS ARE PRINTED *****
C*****
  ITYPE = INT (A3 (2) + .5)
C*****
C***** LOADING ARRAY TAL() FOR SUMMARY OUTPUT *****
C*****
  IF ((ITYPE .NE. 2) .AND. (ITYPE .NE. 3)) THEN
    TAL (1, 1) = 20.
    TAL (2, 1) = 72.
    TAL (3, 1) = 100.
    TAL (4, 1) = 140.
    TAL (5, 1) = 200.
    TAL (6, 1) = 240.
    TAL (7, 1) = 300.
    TAL (8, 1) = 340.
    TAL (9, 1) = 400.
    TAL (10, 1) = 440.
    TAL (11, 1) = 500.
    TAL (12, 1) = 540.
    TAL (13, 1) = 600.
    TAL (14, 1) = 640.
    TAL (15, 1) = 700.
  END IF
C*****
C***** NOW THE HEADINGS *****
C*****
  IF (ITYPE .EQ. 1) THEN
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (7, 6040)
6040  FORMAT(32X,'BURNING RATE ANALYSIS',/,37X,'BRLCB V3.0',
1      /,22X,'ADVANCED BALLISTIC CONCEPTS BRANCH - BRL',///)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6040)
    END IF
  END IF

```

```

IF (ITYPE .EQ. 2) THEN
  IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
    WRITE (7, 6050)
6050  FORMAT(32X,'PRESSURE-TIME GENERATION',/,37X,'BRLCB V3.0',
1      /,22X,'ADVANCED BALLISTIC CONCEPTS BRANCH - BRL'///)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6050)
    END IF
  END IF
  IF (ITYPE .EQ. 3) THEN
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (7, 6060)
6060  FORMAT(32X,'SURFACE AREA ANALYSIS',/,37X,'BRLCB V3.0',
1      /,22X,'ADVANCED BALLISTIC CONCEPTS BRANCH - BRL'///)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6060)
    END IF
  END IF
  IF (ITYPE .EQ. 4) THEN
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (7, 6070)
161  FORMAT(25X,'INTERRUPTED BURNER ANALYSIS',/,37X,'BRLCB V3
1.0',/,22X,'ADVANCED BALLISTIC CONCEPTS BRANCH - BRL'///)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6070)
    END IF
  END IF
  IF (ITYPE .EQ. 5) THEN
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (7, 6080)
6080  FORMAT(32X,'ETC BURN RATE ANALYSIS',/,37X,'BRLCB V3.0',
1      /,22X,'ADVANCED BALLISTIC CONCEPTS BRANCH - BRL'///)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6080)
    END IF
  END IF
  IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
    WRITE (7, 6090) A1 (1), A1 (2)
6090  FORMAT(' Project : ',A20,' Requested by      : ',A20)
    WRITE (7, 6100) A1 (4), A1 (3)
6100  FORMAT(' Inf File: ',A20,' Created From .MAS File : ',A20)

```

```

        WRITE (7, 6110) A1 (5), A1 (10)
6110  FORMAT(' P/T File: ',A20,' Calculation Output File: ',A20)
        WRITE (7, 6120) A1 (14), A1 (17)
6120  FORMAT(' Smoothed: ',A20,' Graphics File          : ',A20)
        IOPTT = INT (A3 (2) + .5)
        IF (IOPTT .EQ. 5) THEN
            WRITE (7, 6130) A1 (19)
6130  FORMAT(' EE File: ',A20)
        END IF
        WRITE (7, 6140) A1 (20)
6140  FORMAT(' Fired on: ',A20)
        IF (A2 (1) .NE. ' ') THEN
            WRITE (7, *) 'FIRING REMARKS:'
            WRITE (7, 6150) A2 (1)
6150  FORMAT(' ',A80)
        END IF
        IF (A2 (2) .NE. ' ') WRITE (7, 6150) A2 (2)
        IF (A2 (3) .NE. ' ') WRITE (7, 6150) A2 (3)
        IF (A2 (4) .NE. ' ') THEN
            WRITE (7, *) 'REDUCTION REMARKS:'
            WRITE (7, 6150) A2 (4)
        END IF
        IF (A2 (5) .NE. ' ') WRITE (7, 6150) A2 (5)
        IF (A2 (6) .NE. ' ') WRITE (7, 6150) A2 (6)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6090) A1 (1), A1 (2)
        WRITE (2, 6100) A1 (4), A1 (3)
        WRITE (2, 6110) A1 (5), A1 (10)
        WRITE (2, 6120) A1 (14), A1 (17)
        IOPTT = INT (A3 (2) + .5)
        IF (IOPTT .EQ. 5) THEN
            WRITE (2, 6130) A1 (19)
        END IF
        WRITE (2, 6140) A1 (20)
        IF (A2 (1) .NE. ' ') THEN
            WRITE (2, *) 'FIRING REMARKS:'
            WRITE (2, 6150) A2 (1)
        END IF
        IF (A2 (2) .NE. ' ') WRITE (2, 6150) A2 (2)
        IF (A2 (3) .NE. ' ') WRITE (2, 6150) A2 (3)
        IF (A2 (4) .NE. ' ') THEN
            WRITE (2, *) 'REDUCTION REMARKS:'
            WRITE (2, 6150) A2 (4)
        END IF
        IF (A2 (5) .NE. ' ') WRITE (2, 6150) A2 (5)
        IF (A2 (6) .NE. ' ') WRITE (2, 6150) A2 (6)
    END IF

```

```

C*****
C***** Igniter Information is printed next *****
C*****
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6160)
6160  FORMAT(/, ' ,20X, 'IGNITER INFORMATION')
        WRITE (7, 6170) A1 (11), A1 (12)
6170  FORMAT(' The Igniter Used Is      : ',A20,' Lot: ',A20)
        WRITE (7, 6180) A1 (13)
6180  FORMAT(' The Source For The Igniter Is: ',A20,/)
        WRITE (7, *) '  IGNITER THERMOCHEMICAL PROPERTIES'
        WRITE (7, 6190) A3 (13), A3 (16)
6190  FORMAT(' Impetus (J/g)      : ',F10.1,' Molecular Weight :',
      1F10.5)
        WRITE (7, 6200) A3 (14), A3 (17)
6200  FORMAT(' Flame Temperature (K):',F10.1,' Covolume (cc/g)  :',
      1F10.5)
        WRITE (7, 6210) A3 (15), A3 (18)
6210  FORMAT(' Density (g/cc)      : ',F10.5,' Gamma          :',
      1F10.5)
        END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6160)
        WRITE (2, 6170) A1 (11), A1 (12)
        WRITE (2, 6180) A1 (13)
        WRITE (2, *) '  IGNITER THERMOCHEMICAL PROPERTIES:'
        WRITE (2, 6190) A3 (13), A3 (16)
        WRITE (2, 6200) A3 (14), A3 (17)
        WRITE (2, 6210) A3 (15), A3 (18)
      END IF
C*****
C***** PROPELLANT INFORMATION *****
C*****
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6220)
6220  FORMAT(/,20X,'PROPELLANT INFORMATION')
        WRITE (7, 6230) A1 (6), A1 (7)
6230  FORMAT(' The Propellant Used Is      : ',A20,' Lot: ',A20)
        WRITE (7, 6240) A1 (8)
6240  FORMAT(' The Source For The Propellant Is: ',A20)
        WRITE (7, 6250)
6250  FORMAT(' ')
        WRITE (7, *) ' Propellant Thermochemical Properties: Following'
        WRITE (7, *) ' Sheets of Output'
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6220)
        WRITE (2, 6230) A1 (6), A1 (7)

```

```

WRITE (2, 6240) A1 (8)
WRITE (2, 6250)
WRITE (2, *) ' Propellant Thermochemical Properties: Following'
WRITE (2, *) ' Sheets of Output'
END IF
C*****
C***** OUTPUT OF GRAIN GEOMETRY *****
C*****
IF ((IOUTCH.EQ. 1) .OR. (IOUTCH.EQ. 3)) THEN
  WRITE (7, 6260)
6260 FORMAT(/,20X,'PROPELLANT GRAIN GEOMETRY')
  WRITE (7, 6270) A1 (9)
6270 FORMAT(/,' Grain Type:',A20)
  ICODEP = INT (A3 (45) + .5)
  IF ((ICODEP.EQ. 11) .OR. (ICODEP.EQ. 13)) ICODEP = 2
  IF (ICODEP.EQ. 12) ICODEP = 3
8000 FORMAT(' ',10X,'Length --- (cm.):',F10.6)
8010 FORMAT(' ',10X,'Outer Diam.(cm.):',F10.6)
8020 FORMAT(' ',10X,'Width ---- (cm.):',F10.6)
8030 FORMAT(' ',10X,'Thickness (cm.):',F10.6)
8040 FORMAT(' ',10X,'Perf Diam. (cm.):',F10.6)
8050 FORMAT(' ',10X,'Slot Width (cm.):',F10.6)
8060 FORMAT(' ',10X,'Inner Web (cm.):',F10.6)
8070 FORMAT(' ',10X,'Middle Web (cm.):',F10.6)
8080 FORMAT(' ',10X,'Outer Web (cm.):',F10.6)
C*****
C***** Sphere *****
C*****
IF (ICODEP.EQ. 1) THEN
  WRITE (7, 8010) A3 (8)
  GO TO 1150
END IF
C*****
C***** Remaining grains all need length. *****
C*****
WRITE (7, 8000) A3 (7)
C*****
C***** Next the rectangular strip is done *****
C*****
IF (ICODEP.EQ. 3) THEN
  WRITE (7, 8020) A3 (8)
  WRITE (7, 8030) A3 (10)
  GO TO 1150
END IF
C*****
C***** The remaining grains all need a diameter *****
C*****
WRITE (7, 8010) A3 (8)

```

```

C*****
C***** Cord needs only a length and width *****
C*****
      IF (ICODEP .EQ. 3) GO TO 1150
C*****
C***** All remaining grains need perf diameter *****
C*****
      WRITE (7, 8040) A3 (9)
C*****
C***** Next the slotted tube *****
C*****
      IF (ICODEP .EQ. 5) THEN
        WRITE (7, 8050) A3 (11)
        GO TO 1150
      END IF
C*****
C***** All remaining grains need inner web *****
C*****
      WRITE (7, 8060) A3 (10)
C*****
C***** 1-perf cylinder is finished *****
C*****
      IF (ICODEP .EQ. 4) GO TO 1150
C*****
C***** Next 7 perf, does not need middle web *****
C*****
      IF (ICODEP .GT. 7) WRITE (7, 8070) A3 (11)
C*****
C***** Finally the outer web *****
C*****
      WRITE (7, 8080) A3 (12)
      END IF
1150 CONTINUE
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6260)
        WRITE (2, 6270) A1 (9)
        ICODEP = INT (A3 (45) + .5)
        IF ((ICODEP .EQ. 11) .OR. (ICODEP .EQ. 13)) ICODEP = 2
        IF (ICODEP .EQ. 12) ICODEP = 3
C*****
C***** Sphere *****
C*****
        IF (ICODEP .EQ. 1) THEN
          WRITE (2, 8010) A3 (8)
          GO TO 1160
        END IF
C*****
C***** Remaining grains all need length. *****

```

```

C*****
WRITE (2, 8000) A3 (7)
C*****
C***** Next the rectangular strip is done *****
C*****
IF (ICODEP.EQ. 3) THEN
WRITE (2, 8020) A3 (8)
WRITE (2, 8030) A3 (10)
GO TO 1160
END IF
C*****
C***** The remaining grains all need a diameter *****
C*****
WRITE (2, 8010) A3 (8)
C*****
C***** Cord needs only a length and width *****
C*****
IF (ICODEP.EQ. 3) GO TO 1160
C*****
C***** All remaining grains need perf diameter *****
C*****
WRITE (2, 8040) A3 (9)
C*****
C***** Next the slotted tube *****
C*****
IF (ICODEP.EQ. 5) THEN
WRITE (2, 8050) A3 (11)
GO TO 1160
END IF
C*****
C***** All remaining grains need inner web *****
C*****
WRITE (2, 8060) A3 (10)
C*****
C***** 1-perf cylinder is finished *****
C*****
IF (ICODEP.EQ. 4) GO TO 1160
C*****
C***** Next 7 perf, does not need middle web *****
C*****
IF (ICODEP.GT. 7) WRITE (2, 8070) A3 (11)
C*****
C***** Finally the outer web *****
C*****
WRITE (2, 8080) A3 (12)
END IF
1160 CONTINUE
C*****

```

```

C***** Next the hardware information is given *****
C*****
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6280)
        6280 FORMAT(/, ' ',5X,'Bomb Information',15X,'Gage Information')
        WRITE (7, 6290)
        6290 FORMAT(' -----')
        WRITE (7, 6300) A1 (16), A1 (15)
        6300 FORMAT(' ', 'Bomb Type  ',A20,2X,'Gage I.D.  ',A20)
        WRITE (7, 6310) A3 (23), A3 (19)
        6310 FORMAT(' ', 'Bomb Vol (cc):',F10.1,12X,'Input Voltage:',F8.4)
        WRITE (7, 6320)
        6320 FORMAT(' ',36X,'Constants For Fit: A+Bx+C^2')
        WRITE (7, 6330) A3 (20)
        6330 FORMAT(' ',36X,'A: ',E12.5)
        WRITE (7, 6340) A3 (21)
        6340 FORMAT(' ',36X,'B: ',E12.5)
        WRITE (7, 6350) A3 (22)
        6350 FORMAT(' ',36X,'C: ',E12.5)
        WRITE (7, 6250)
C*****
C***** Specific Information for the firing *****
C*****
        WRITE (7, 6360)
        6360 FORMAT(' ',8X,'Temperature and Charge Mass Information')
        WRITE (7, 6290)
        WRITE (7, 6370) A3 (25), A3 (26)
        6370 FORMAT(' ', 'Propellant Mass (g)  ',F9.4,6X,'Igniter Mass (g):', F9.4)
        WRITE (7, 6380) A3 (27), A3 (28)
        6380 FORMAT(' ', 'Initial Temp. Prop.(K):',F6.0,9X,'Igniter Temp.(K):', F6.0)
        WRITE (7, 6390) A3 (24)
        6390 FORMAT(' ', 'Initial Bomb Temp. (K):',F6.0)
        WRITE (7, 6400) A3 (46)
        6400 FORMAT(' ', 'Number of Propellant Grains:',F9.2)
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6280)
        WRITE (2, 6290)
        WRITE (2, 6300) A1 (16), A1 (15)
        WRITE (2, 6310) A3 (23), A3 (19)
        WRITE (2, 6320)
        WRITE (2, 6330) A3 (20)
        WRITE (2, 6340) A3 (21)
        WRITE (2, 6350) A3 (22)
        WRITE (2, 6250)
C*****

```

```

C***** Specific Information for the firing *****
C*****
    WRITE (2, 6360)
    WRITE (2, 6290)
    WRITE (2, 6370) A3 (25), A3 (26)
    WRITE (2, 6380) A3 (27), A3 (28)
    WRITE (2, 6390) A3 (24)
    WRITE (2, 6400) A3 (46)
END IF
C*****
C***** NEXT THE RESULTS ARE PRINTED -- SIXTY LINES TO A PAGE *****
C*****
    CALL CLEAR
    WRITE (*, *) 'Normally 60 lines are printed per page'
    WRITE (*, *) 'However, this value can be change to suit'
    WRITE (*, *) 'the printer being used. Do you wish to change'
    WRITE (*, *) 'the default setting [Y/N]. Default is No.'
    READ (*, 5030) A
5030 FORMAT(A1)
    CALL CLEAR
    IF ((A .EQ. 'Y') .OR. (A .EQ. 'y')) THEN
        WRITE (*, *) 'Enter the number of lines per page.'
        READ (*, *) NPAH
    ELSE
        NPAH = 60
    END IF
    OPEN (UNIT = 12, FILE = A1 (10))
    REWIND (UNIT = 12)
    CALL CLEAR
    NUMB = INT (A3 (100) + .5)
    WRITE (*, 6410) NUMB
6410 FORMAT(10X,'There are',I5,' lines of output from the',
1/,10X,'computation. The amount of lines printed can',
2/,10X,'be reduced by specifying a "skip" factor. A',
3/,10X,'skip factor of 1 will print all lines, a skip',
4/,10X,'factor of 2 will print every other line, etc. ',
5/,10X,'Enter your choice for a skip factor.'//)
    READ (*, *) ISKIP
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6420) NAMY
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6420) NAMY
    END IF
6420 FORMAT('1', ' OUTPUT FILE: ',A20)
C*****
C***** FIRST THREE LINES OF OUTPUT ARE BLANK *****
C*****

```

```

      READ (12, 5040) LINE
      READ (12, 5040) LINE
      READ (12, 5040) LINE
      NUMPAG = 0
1170 CONTINUE
      READ (12, 5040, END = 1180) LINE
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6430) LINE
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6430) LINE
      END IF
      NUMPAG = NUMPAG + 1
      IF (NUMPAG .EQ. NPAH) THEN
        IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (7, 6420) NAMY
        END IF
        IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (2, 6420) NAMY
        END IF
        NUMPAG = 0
      END IF
      IF ((LINE (1:1) .NE. 'N') .AND. (LINE (2:2) .NE. 'N')) GO TO 1170
      NSKIP = 0
1190 CONTINUE
      READ (12, 5040, END = 1180) LINE
5040  FORMAT(A132)
6430  FORMAT(' ',A132)
      NSKIP = NSKIP + 1
      IF (NSKIP .EQ. ISKIP) THEN
        NSKIP = 0
        NUMPAG = NUMPAG + 1
        IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (7, 6430) LINE
        END IF
        IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (2, 6430) LINE
        END IF
        IF (NUMPAG .EQ. NPAH) THEN
          NUMPAG = 0
          IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
            WRITE (7, 6420) NAMY
          END IF
          IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
            WRITE (2, 6420) NAMY
          END IF
        END IF
      END IF

```

```

GO TO 1190
1180 CONTINUE
  IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
    WRITE (7, 6430) LINE
  END IF
  IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
    WRITE (2, 6430) LINE
  END IF
  CLOSE (UNIT = 12)
C*****
C***** NEED SUBROUTINE TO STRIP OFF THE INFORMATION DESIRED *****
C***** BURN RATE FOR OPTIONS 1, 4 & 5, SURFACE AREA FOR OPTION 3 *****
C***** AND TIME PRESSURE FOR OPTION 2 *****
C*****
1200 CONTINUE
  KNOUT = 0
  IOPT = INT (A3 (2) + .5)
  OPEN (UNIT = 12, FILE = A1 (10))
  REWIND (UNIT = 12)
  DO 1210 I = 1, 5000
    READ (12, 5050, END = 1220) CHART
5050  FORMAT(A2)
    IF ((CHART .EQ. ' N') .OR. (CHART .EQ. 'N ')) THEN
      BACKSPACE (UNIT = 12)
      KNOUT = KNOUT + 1
      READ (12, 5060) WORD1, WORD2, X1, X2, X3, X4, X5, X6, X7
5060  FORMAT(A9,A10,7E16.8)
      IF (KNOUT .EQ. 1) A3 (48) = X5
    ELSE
      GO TO 1210
    END IF
    IF (X4 .LE. 0.0) THEN
      KNOUT = KNOUT - 1
      GO TO 1210
    END IF
    IF ((IOPT .EQ. 1) .OR. (IOPT .EQ. 4) .OR. (IOPT .EQ. 5)) THEN
      FRAT (KNOUT, 1) = X2
      FRAT (KNOUT, 2) = X4
      FRAT (KNOUT, 3) = 0.0
    END IF
    IF (IOPT .EQ. 3) THEN
      FRAT (KNOUT, 1) = (A3 (25) - X3)/A3 (25)
      FRAT (KNOUT, 3) = X5/A3 (48)
      FRAT (KNOUT, 2) = X7
    END IF
    IF (IOPT .EQ. 2) THEN
      FRAT (KNOUT, 1) = X1
      FRAT (KNOUT, 2) = X2

```

```

        FRAT (KNOUT, 3) = 0.0
    END IF
1210 CONTINUE
1220 CONTINUE
    CLOSE (UNIT = 12)
C*****
C***** GRAPHICS FILE IS WRITTEN *****
C*****
    DO 1230 J = 1, KNOUT
        WRITE (9, *) FRAT (J, 1), FRAT (J, 2), FRAT (J, 3)
1230 CONTINUE
    CLOSE (UNIT = 9)
C*****
C***** BURN RATES ARE WRITTEN *****
C*****
    IF ((IOPT .EQ. 2) .OR. (IOPT .EQ. 3)) GO TO 1240
C*****
C***** FIRST SEE IF FFT IS DESIRED *****
C*****
    CALL CLEAR
    IFLAG = 1
    WRITE (*, 6440)
6440 FORMAT(///,15X,'The burn rate versus pressure often requires',
    1/,15X,'filtering to remove oscillations. Do you wish to',
    2/,15X,'determine the effect of the FFT? (Yes=1, No=2) ',
    3/,15X,'Enter your choice.'/)
    READ (*, *) IFFT
    IF (IFFT .EQ. 1) THEN
        IFLAG = 2
    END IF
    CALL BRFFT (A1, IFLAG)
C***** PRINTING INFORMATION ON THE FFT *****
    IF (IFFT .EQ. 1) THEN
        CALL CLEAR
        WRITE (*, *) 'Is FFT filtered data being used for the burn ratio?'
        WRITE (*, *) 'Enter your choice. (Yes = 1, No = 2)'
        READ (*, *) JRTI
        IF (JRTI .EQ. 1) THEN
            IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (7, 3914)
3914 FORMAT(///,10X,'FFT filter used on burn rate data.')
            END IF
            IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (2, 3914)
            END IF
        END IF
    END IF
C*****

```

```

C***** NEED TO DETERMINE WHERE THE FFT DATA IS STORED *****
C***** LOAD FFT DATA INTO ARRAY FRAT, AND RECOPY IT TO *****
C***** GRAPHICS FILE IF FILTERED DATA IS TO BE USED *****
C*****
C***** NEED TO INTERPOLATE AT STEPS OF 4 MPa *****
C***** STORE THE DATA IN FRAT WITH NUMBER AS KKN *****
C***** WILL STORE A COPY IN XLFRAT *****
C*****
C***** REREAD GRAPHICS FILE IN CASE FILTERED DATA *****
C***** IS TO BE USED, IFLAG=2 MEANS FILTERED DATA *****
C*****
      IF (IFLAG .EQ. 2) THEN
        OPEN (UNIT = 9, FILE = A1 (17))
        KNOUT = 1
1250    CONTINUE
        READ (9, *, END = 1260) FRAT(KNOUT, 1), FRAT1
           (KNOUT, 2), FRAT (KNOUT, 3)
        KNOUT = KNOUT + 1
        GO TO 1250
1260    CONTINUE
        KNOUT = KNOUT - 1
        CLOSE (UNIT = 9)
      END IF
      DO 1270 I = 1, KNOUT
        XLFRAT (I, 1) = FRAT (I, 1)
        XLFRAT (I, 2) = FRAT (I, 2)
1270    CONTINUE
C*****
C***** DETERMINE UPPER AND LOWER MULTIPLE OF 4 MPA *****
C*****
      DO 1280 I = 1, 1000
        TEST = I*4
        IF (FRAT (1, 1) .LE. TEST) THEN
          NS = I*4
          GO TO 1290
        END IF
1280    CONTINUE
1290    CONTINUE
        NT = INT (FRAT (KNOUT, 1)/4.)*4
C*****
C***** NOW THE INTERPOLATION *****
C*****
      JQRY = 0
      PMAX = FRAT (KNOUT, 1)
      KKN = 0
      DO 1300 I = NS, NT, 4
        KKN = KKN + 1
        FRAT (KKN, 1) = I

```

```

DO 1310 J = 1, KNOUT
  X = I
  IF (X .LE. XLFRAT (J, 1)) THEN
    FRAT (KKN, 2) = (X - XLFRAT (J - 1, 1))/(XLFRAT (J, 1) - XLFRAT (J - 1, 1))*
1    (XLFRAT (J, 2) - XLFRAT (J - 1, 2)) + XLFRAT (J - 1, 2)
    DO 1320 KIX = 1, 15
      ITESTIT = INT (TAL (KIX, 1) + .5)
      IF (ITESTIT .EQ. I) THEN
        TAL (KIX, 2) = FRAT (KKN, 2)
        JQRY = JQRY + 1
      END IF
1320    CONTINUE
      GO TO 1300
    END IF
1310  CONTINUE
1300 CONTINUE
C*****
C***** NOW PRINT THE RATES *****
C*****
  CALL PRAT (FRAT, KKN, IOUTCH)
C*****
C***** SUBROUTINE TO DETERMINE THE BURN RATE LAWS *****
C*****
  CALL BRNRT (A1, IOUTCH)
1240 CONTINUE
  CLOSE (UNIT = 2)
C*****
C***** SUMMARY OUTPUT ROUTINE GOES HERE *****
C***** ALWAYS TO PRINTER *****
C*****
  IF ((IOPT .EQ. 2) .OR. (IOPT .EQ. 3)) GO TO 1330
  CALL CLEAR
  WRITE (*, 6460)
6460 FORMAT(/,10X,'Summary Output Sheet to Printer? (Yes=1, No=2)')
  READ (*, *) ISUMM
  IF (ISUMM .EQ. 2) GO TO 1340
  IF (IOUTCH .EQ. 2) OPEN (UNIT = 7, FILE = 'LPT1')
  WRITE (7, 6420) NAMY
  WRITE (7, 6470) A1 (4)
6470 FORMAT(' Information File:',A20,8X,'Pressure',10X,'Burn Rate')
  WRITE (7, 6480) A1 (1)
6480 FORMAT(' Project Name :',A20,10X,'(MPa)',12X,'(cm/sec)')
  IF (JQRY .GE. 1) THEN
    WRITE (7, 6490) A1 (6), TAL (1, 1), TAL (1, 2)
6490 FORMAT(' Propellant Type :',A20,F14.1,9X,F9.3)
  ELSE

```

```

WRITE (7, 6500) A1 (6)
6500 FORMAT(' Propellant Type :',A20)
END IF
IF (JQRY .GE. 2) THEN
    WRITE (7, 6510) A1 (9), TAL (2, 1), TAL (2, 2)
6510 FORMAT(' Grain Geometry :',A20,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6520) A1 (9)
6520 FORMAT(' Grain Geometry :',A20)
END IF
IF (JQRY .GE. 3) THEN
    WRITE (7, 6530) A3 (7), TAL (3, 1), TAL (3, 2)
6530 FORMAT(' Grain Length(cm):',F20.5,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6540) A3 (7)
6540 FORMAT(' Grain Length(cm):',F20.5)
END IF
IF (JQRY .GE. 4) THEN
    WRITE (7, 6550) A3 (8), TAL (4, 1), TAL (4, 2)
6550 FORMAT(' Outer Diameter :',F20.5,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6560) A3 (8)
6560 FORMAT(' Outer Diameter :',F20.5)
END IF
IF (JQRY .GE. 5) THEN
    WRITE (7, 6570) A3 (9), TAL (5, 1), TAL (5, 2)
6570 FORMAT(' Perf Diameter :',F20.5,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6580) A3 (9)
6580 FORMAT(' Perf Diameter :',F20.5)
END IF
IF (JQRY .GE. 6) THEN
    WRITE (7, 6590) A3 (10), TAL (6, 1), TAL (6, 2)
6590 FORMAT(' Inner Web :',F20.5,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6600) A3 (10)
6600 FORMAT(' Inner Web :',F20.5)
END IF
IF (JQRY .GE. 7) THEN
    WRITE (7, 6610) A3 (11), TAL (7, 1), TAL (7, 2)
6610 FORMAT(' Middle Web :',F20.5,F14.1,9X,F9.3)
ELSE
    WRITE (7, 6620) A3 (11)
6620 FORMAT(' Middle Web :',F20.5)
END IF
IF (JQRY .GE. 8) THEN
    WRITE (7, 6630) A3 (12), TAL (8, 1), TAL (8, 2)
6630 FORMAT(' Outer Web :',F20.5,F14.1,9X,F9.3)

```

```

ELSE
  WRITE (7, 6640) A3 (12)
6640 FORMAT(' Outer Web      :',F20.5)
END IF
C*****
C***** CHECK FOR DETERRED OR LAYERED GRAIN *****
C*****
NL = INT (A3 (4) + .5)
IFLAGY = 0
IF (NL .GT. 1) IFLAGY = 1
IF (JQRY .GE. 9) THEN
  WRITE (7, 6650) TAL (9, 1), TAL (9, 2)
6650 FORMAT(' Prop Thermochems:',20X,F14.1,9X,F9.3)
ELSE
  WRITE (7, 6660)
6660 FORMAT(' Prop Thermochems:')
END IF
IF (IFLAGY .EQ. 1) THEN
  IF (JQRY .GE. 10) THEN
    WRITE (7, 6670) TAL (9, 1), TAL (9, 2)
6670 FORMAT(' Layered/Deterred:',20X,F14.1,9X,F9.3)
  ELSE
    WRITE (7, 6680)
6680 FORMAT(' Layered/Deterred:')
  END IF
  IF (JQRY .GE. 11) WRITE (7, 6690) TAL (11, 1), TAL (11, 2)
  IF (JQRY .GE. 12) WRITE (7, 6690) TAL (12, 1), TAL (12, 2)
  IF (JQRY .GE. 13) WRITE (7, 6690) TAL (13, 1), TAL (13, 2)
  IF (JQRY .GE. 14) WRITE (7, 6690) TAL (14, 1), TAL (14, 2)
  IF (JQRY .GE. 15) WRITE (7, 6690) TAL (15, 1), TAL (15, 2)
6690 FORMAT(' ',37X,F14.1,9X,F9.3)
END IF
IF (IFLAGY .EQ. 0) THEN
  IF (JQRY .GE. 10) THEN
    WRITE (7, 6700) P (2, 1, 1), TAL (10, 1), TAL (10, 2)
6700 FORMAT(' Impetus (J/g)  :',F20.5,F14.1,9X,F9.3)
  ELSE
    WRITE (7, 6710) P (2, 1, 1)
6710 FORMAT(' Impetus (J/g)  :',F20.5)
  END IF
  IF (JQRY .GE. 11) THEN
    WRITE (7, 6720) P (3, 1, 1), TAL (11, 1), TAL (11, 2)
6720 FORMAT(' Flame Temp (K)  :',F20.5,F14.1,9X,F9.3)
  ELSE
    WRITE (7, 6730) P (3, 1, 1)
6730 FORMAT(' Flame Temp (K)  :',F20.5)
  END IF
  IF (JQRY .GE. 12) THEN

```

```

        WRITE (7, 6740) P (4, 1, 1), TAL (12, 1), TAL (12, 2)
6740 FORMAT(' Density (g/cc) :',F20.5,F14.1,9X,F9.3)
        ELSE
            WRITE (7, 6750) P (4, 1, 1)
6750 FORMAT(' Density (g/cc) :',F20.5)
        END IF
        IF (JQRY .GE. 13) THEN
            WRITE (7, 6760) P (5, 1, 1), TAL (13, 1), TAL (13, 2)
6760 FORMAT(' Molecular Weight:',F20.5,F14.1,9X,F9.3)
        ELSE
            WRITE (7, 6770) P (5, 1, 1)
6770 FORMAT(' Molecular Weight:',F20.5)
        END IF
        IF (JQRY .GE. 14) THEN
            WRITE (7, 6780) P (6, 1, 1), TAL (14, 1), TAL (14, 2)
6780 FORMAT(' Covolume (cc/g) :',F20.5,F14.1,9X,F9.3)
        ELSE
            WRITE (7, 6790) P (6, 1, 1)
6790 FORMAT(' Covolume (cc/g) :',F20.5)
        END IF
        IF (JQRY .GE. 15) THEN
            WRITE (7, 6800) P (7, 1, 1), TAL (15, 1), TAL (15, 2)
6800 FORMAT(' Gamma :',F20.5,F14.1,9X,F9.3)
        ELSE
            WRITE (7, 6810) P (7, 1, 1)
6810 FORMAT(' Gamma :',F20.5)
        END IF
    END IF
C*****
C***** NEXT BURN RATE LAWS *****
C*****
    OPEN (UNIT = 18, FILE = A1 (18))
    REWIND (UNIT = 18)
    READ (18, *) NN
    IF (NN .GT. 8) THEN
        NS = NN - 7
        DO 1350 I = 1, NS - 1
            READ (18, *) X, Y, Z, W, T
1350    CONTINUE
    ELSE
        NS = 1
    END IF
    WRITE (7, 6820)
6820 FORMAT(' ', 'Burn Rate Laws (MPa & cm/sec)')
    WRITE (7, 6830)
2001 FORMAT(' Pressure -- Pressure', 2X, 'Coeff', 7X, 'Exp', 4X, 'Corr Coeff')
    DO 1360 I = NS, NN

```

```

      READ (18, *) X, Y, Z, W, T
      WRITE (7, 6840) X, Y, Z, W, T
6840  FORMAT(' ',F8.2,4X,F8.2,F9.6,2X,F8.4,2X,F10.7)
1360  CONTINUE
      CLOSE (UNIT = 18)
C*****
C*****  FINALLY THE GRAPH OF THE PRESSURE VS RATE *****
C*****
C*****  FIRST FIND THE RANGES ON AXIS *****
C*****  PRESSURE IS FIRST AND LAST ENTRIES *****
C*****  SEARCH THE RATES *****
C*****
C*****  NEED TO REREAD THE GRAPHICS FILE *****
C*****  MAY NEED TO USE FILTERED DATA *****
C*****
      OPEN (UNIT = 14, FILE = A1 (17))
      REWIND (UNIT = 14)
      KNOUT = 0
1370  CONTINUE
      KNOUT = KNOUT + 1
      READ (14, *, END = 1380) XLFRAT (KNOUT, 1), XLFRAT (KNOUT, 2), XXX
      GO TO 1370
1380  CONTINUE
      KNOUT = KNOUT - 1
      CLOSE (UNIT = 14)
      PMAXY = XLFRAT (KNOUT, 1)
      PMINY = XLFRAT (1, 1)
      RMAXY = XLFRAT (1, 2)
      RMINY = XLFRAT (1, 2)
      DO 1390 I = 1, KNOUT
        IF (XLFRAT (I, 2) .LT. RMINY) RMINY = XLFRAT (I, 2)
        IF (XLFRAT (I, 2) .GT. RMAXY) RMAXY = XLFRAT (I, 2)
1390  CONTINUE
      WRITE (7, 6850) PMINY, PMAXY, RMINY, RMAXY
6850  FORMAT(/, ' ', 'Hor. Axis:', F8.2, ' to ', F8.2, ' MPa :: Ver. Axis:',
      1F9.5, ' to ', F9.5, ' cm/sec')
C*****
C*****  AXIS INCREMENT IS DETERMINED *****
C*****  AXIS LOADED INTO ARRAY *****
C*****
      V1 = RMINY
      V4 = RMAXY
      H1 = PMINY
      H4 = PMAXY
      PMAXY = ALOG10 (PMAXY)
      PMINY = ALOG10 (PMINY)
      RMAXY = ALOG10 (RMAXY)
      RMINY = ALOG10 (RMINY)

```

```

HSTEP = (PMAXY - PMINY)/60.
VSTEP = (RMAXY - RMINY)/30.
DO 1400 I = 2, 60
  APL (1, I) = '-'
1400 CONTINUE
  APL (1, 20) = 'I'
  APL (1, 40) = 'I'
  APL (1, 60) = 'I'
  DO 1410 I = 1, 30
    APL (I, 1) = 'I'
1410 CONTINUE
  APL (1, 1) = '+'
  APL (10, 1) = '-'
  APL (20, 1) = '-'
  APL (30, 1) = '-'
C*****
C***** PLOTTING POINTS IN ARRAY *****
C*****
  DO 1420 I = 1, KNOUT
    X = ALOG10 (XLFRAT (I, 1)) - PMINY
    Y = ALOG10 (XLFRAT (I, 2)) - RMINY
    IX = INT (X/HSTEP + .5) + 1
    IY = INT (Y/VSTEP + .5) + 1
    IF ((IX .GT. 60) .OR. (IX .LT. 0) .OR. (IY .GT. 30) .OR. (IY .LT. 0)) GO TO 1420
    APL (IY, IX) = '*'
1420 CONTINUE
C*****
C***** PLOTTING THE GRAPH TO PRINTER *****
C*****
8090 FORMAT(' ',F9.5,X,A60)
8100 FORMAT(' ',14X,A60)
  DO 1430 I = 30, 1, - 1
    DO 1440 J = 1, 60
      ITEMP (J:J) = APL (I, J)
1440 CONTINUE
    IF (I .EQ. 30) THEN
      WRITE (7, 8090) V4, ITEMP
    ELSE IF (I .EQ. 20) THEN
      V3 = 10.**(20.*VSTEP + RMINY)
      WRITE (7, 8090) V3, ITEMP
    ELSE IF (I .EQ. 10) THEN
      V2 = 10.**(10.*VSTEP + RMINY)
      WRITE (7, 8090) V2, ITEMP
    ELSE IF (I .EQ. 1) THEN
      WRITE (7, 8090) V1, ITEMP
    ELSE
      WRITE (7, 8100) ITEMP

```

```

      END IF
1430 CONTINUE
C*****
C***** FINALLY SCALE ON HORIZONTAL AXIS *****
C*****
      H2 = 10.*(20.*HSTEP + PMINY)
      H3 = 10.*(40.*HSTEP + PMINY)
      WRITE (7, 6860) H1, H2, H3, H4
6860 FORMAT(' ',9X,F6.0,14X,F6.0,14X,F6.0,14X,F6.0)
C*****
C***** FINISHED THE OUTPUT *****
C*****
1330 CONTINUE
1340 CONTINUE
      END
C*****
C***** SUBROUTINE TO PRINT RATES IN TABLE *****
C*****
      SUBROUTINE PRAT (FRAT, KNOUT, IOUTCH)
C*****
C  THIS PROGRAM WILL WRITE A TABLE OF PRESSURE VS RATE
C  WRITTEN BY: WILLIAM OBERLE, BRL MARCH 1987
C*****
      DIMENSION FRAT (1024, 3), PRESS (1024), RATE (1024)
      DO 1000 I = 1, KNOUT
        PRESS (I) = FRAT (I, 1)
        RATE (I) = FRAT (I, 2)
1000 CONTINUE
      NN = KNOUT
      N11 = 1
1010 CONTINUE
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6000)
        WRITE (7, 6010)
        WRITE (7, 6020)
        WRITE (7, *)
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6000)
        WRITE (2, 6010)
        WRITE (2, 6020)
        WRITE (2, *)
      END IF
      IF (NN .LE. 165) THEN
        N1 = N11
        N2 = N11 + NN - 1
        NFLAG = 1
      ELSE

```

```

N1 = N11
N2 = N11 + 164
NN = NN - 165
N11 = N11 + 165
NFLAG = 0
END IF
NUM = N2 - N1 + 1
XNUM = NUM
NROWS = XNUM/3
NLEFT = NUM - NROWS*3.
IF (NLEFT .LT. 1) THEN
  DO 1020 I = 1, NROWS
    NI = N1 + I - 1
    IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (7, 6030) PRESS (NI), RATE
1      (NI), PRESS (NI + NROWS),
2      RATE (NI + NROWS), PRESS (NI + 2*NROWS),
3      RATE (NI + 2*NROWS)
    END IF
    IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
      WRITE (2, 6030) PRESS (NI), RATE
1      (NI), PRESS (NI + NROWS),
2      RATE (NI + NROWS), PRESS (NI + 2*NROWS),
3      RATE (NI + 2*NROWS)
    END IF
1020  CONTINUE
  ELSE
    IF (NLEFT .EQ. 1) THEN
      DO 1030 I = 1, NROWS
        NI = N1 + I - 1
        IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (7, 6030) PRESS (NI), RATE (NI),
1          PRESS (NI + NROWS + 1)
2          , RATE (NI + NROWS + 1), PRESS (NI + 2*NROWS + 1)
3          , RATE (NI + 2*NROWS + 1)
        END IF
        IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
          WRITE (2, 6030) PRESS (NI), RATE (NI),
1          PRESS (NI + NROWS + 1)
2          , RATE (NI + NROWS + 1), PRESS (NI + 2*NROWS + 1)
3          , RATE (NI + 2*NROWS + 1)
        END IF
1030  CONTINUE
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6040) PRESS (NROWS + N1), RATE (NROWS + N1)
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6040) PRESS (NROWS + N1), RATE (NROWS + N1)
      END IF
    END IF
  END IF

```

```

        END IF
    ELSE
        DO 1040 I = 1, NROWS
            NI = N1 + I - 1
            IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (7, 6030) PRESS (NI), RATE (NI),
1             PRESS (NI + NROWS + 1)
2             , RATE (NI + NROWS + 1), PRESS (NI + 2*NROWS + 2)
3             , RATE (NI + 2*NROWS + 2)
            END IF
            IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (2, 6030) PRESS (NI), RATE (NI),
1             PRESS (NI + NROWS + 1)
2             , RATE (NI + NROWS + 1), PRESS (NI + 2*NROWS + 2)
3             , RATE (NI + 2*NROWS + 2)
            END IF
1040    CONTINUE
            IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (7, 6050) PRESS (NROWS + N1), RATE (NROWS + N1),
1             PRESS (N1 + 2*NROWS + 1), RATE (N1 + 2*NROWS + 1)
            END IF
            IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
                WRITE (2, 6050) PRESS (NROWS + N1), RATE (NROWS + N1),
1             PRESS (N1 + 2*NROWS + 1), RATE (N1 + 2*NROWS + 1)
            END IF
        END IF
    END IF
    IF (NFLAG .EQ. 1) GO TO 1050
    GO TO 1010
6030  FORMAT(' ',12X,F8.3,4X,F8.3,4X,F8.3,4X,F8.3,4X,F8.3)
6050  FORMAT(' ',12X,F8.3,4X,F8.3,4X,F8.3,4X,F8.3)
6040  FORMAT(' ',12X,F8.3,4X,F8.3)
6000  FORMAT('1',/)
6010  FORMAT(' ',13X,'Pressure',6X,'Rate',6X,'Pressure',6X,'Rate',
1       6X,'Pressure',6X,'Rate')
6020  FORMAT(' ',13X,'-----',6X,'----',6X,'-----',6X,'----',
1       6X,'-----',6X,'----')
1050  CONTINUE
    RETURN
    END
C*****
C***** SUBROUTINE CLEAR *****
C*****
    SUBROUTINE CLEAR
    CHARACTER ST*4
    DATA ST/' [2J'/
    WRITE (*, 6000) ST
    6000  FORMAT (1X,A4)

```

```

RETURN
END
C*****
C***** SUBROUTINE BRFFT *****
C*****
SUBROUTINE BRFFT (A1, IFLAG)
C*****
C***** FAST FOURIER TRANSFORM ROUTINE PROVIDED BY F. ROBBINS *****
C***** FFT RESULTS STORED IN THE GRAPHICS FILE IF USER SELECTED **
C*****
COMPLEX A (1024)
DIMENSION DATA (1024, 3)
DIMENSION FREQ (1024), RMAG (1024)
CHARACTER*20 FILNAM, XAXISL, YAXISL, A1 (20)
CHARACTER*40 TITLE
DIMENSION B (1024)
DIMENSION C (1024)
DIMENSION D (1024)
EQUIVALENCE (DATA (1, 1), B (1)), (DATA (1, 3), C (1), RMAG (1))
EQUIVALENCE (DATA (1, 2), D (1))
PI = 3.1415926535
C*****
C***** OBTAIN PRINTER & SCREEN CHARACTERISTICS *****
C*****
CALL CLEAR
WRITE (*, 6000)
6000 FORMAT(/,10X,'This program has the capability of generating',
1/,10X,'hard copy plots of the pressure-burn rate data. To',
2/,10X,'generate plots information concerning your printer is',
3/,10X,'required. The output port is assumed to be LPT1.',
4//,10X,'Select printer by number.',
5/,15X,'1. Epson Dot Matrix',
6/,15X,'2. HP Laser Jet ',//,10X,'Enter Choice.')
READ (*, *) NAIS
IF (NAIS .EQ. 1) THEN
MODEL = 5
ELSE
MODEL = 62
END IF
C*****
C***** READ PRESSURE & RATE FROM GRAPHICS FILE *****
C*****
OPEN (UNIT = 14, FILE = A1 (17))
J = 0
1000 CONTINUE
READ (14, *, END = 1010) P, DXDT, DPDT
J = J + 1
IF (P .LT. 7.) GO TO 1000

```

```

      JP7 = J
      PMAX = P
1020 CONTINUE
      READ (14, *, END = 1010) P, DXDT, DPDT
      J = J + 1
      IF (P .GT. PMAX) THEN
        PMAX = P
        JPMAX = J
      END IF
      GO TO 1020
1010 CONTINUE
      JTOTAL = J
      REWIND (UNIT = 14)
      NPT7PM = JPMAX - JP7 + 1
      M = INT (ALOG (REAL (NPT7PM))/ALOG (2.0) + 1.0E-10)
      N = M + 1
      M = N
      NPT = 2.0**N + 1.0E-10
      N = NPT
      IF (JTOTAL .GT. 0.1*NPT7PM + JPMAX) JTOTAL = PMAX + 0.1*NPT7PM
      IF (JTOTAL .GE. NPT) THEN
        IF (JTOTAL .EQ. NPT) GO TO 1030
        DO 1040 I = 1, JTOTAL - NPT
          READ (14, *) P, DXDT, DPDT
1040  CONTINUE
1030  CONTINUE
        DO 1050 I = 1, NPT
          READ (14, *) P, DXDT, DPDT
          DATA (I, 1) = P
          DATA (I, 2) = DXDT
          DATA (I, 3) = DPDT
          A (I) = CMPLX (DXDT, 0.0)
1050  CONTINUE
        ELSE
          J = 0
          READ (14, *) P, DXDT, DPDT
          DO 1060 I = 1, NPT - JTOTAL + 1
            A (I) = CMPLX (DXDT, 0.0)
            J = J + 1
            DATA (I, 1) = P
            DATA (I, 2) = DXDT
            DATA (I, 3) = DPDT
1060  CONTINUE
          DO 1070 I = 1, JTOTAL - 1
            READ (14, *) P, DXDT, DPDT
            DATA (J + I, 1) = P
            DATA (J + I, 2) = DXDT
            DATA (J + I, 3) = DPDT

```

```

      A (J + I) = CMPLX (DXDT, 0.0)
1070  CONTINUE
      END IF
      CLOSE (UNIT = 14)
C***** ATTEMPT TO BY PASS THE FILTERING TO GET PLOT *****
      IF (IFLAG .EQ. 1) GO TO 1080
C GET FILTERING WINDOW
      CALL FFT (A, M, NPT, - 1.0, 2.0/REAL (NPT))
      NX = 0
      DO 1090 I = 1, N/2
        FREQ (I) = REAL (I - 1)/REAL (N)
        IF (FREQ (I) .LE. FMAX) NX = NX + 1
1090  CONTINUE
      DO 1100 I = 1, N/2
        RMAG (I) = SQRT (REAL (A (I))**2 + AIMAG (A (I))**2)
1100  CONTINUE
      XAXISL = ' FREQ(Hz)'
      YAXISL = ' MAGNITUDE'
      TITLE = ' FFT SPECTRUM'
      CALL PL88P (FREQ, RMAG, N/2, XAXISL, YAXISL, TITLE, MODEL)
      WRITE (*, *) 'INPUT CUT OFF FREQUENCY FOR LOW PASS FILTER'
      READ (*, *) FMAX
      DO 1110 I = 1, N/2
        IF (FREQ (I) .GE. FMAX) THEN
          IM = N - I + 2
          A (I) = CMPLX (0.0, 0.0)
          A (IM) = CMPLX (0.0, 0.0)
        END IF
1110  CONTINUE
      CALL FFT (A, M, NPT, 1.0, 0.5)
1080  CONTINUE
      WRITE (*, *) 'INPUT PLOT OPTION DESIRED'
      WRITE (*, *) '1. PLOT ORIGINAL DATA'
      WRITE (*, *) '2. PLOT FILTERED DATA'
      WRITE (*, *) '3. OVERPLOT ORIGINAL AND FILTERED DATA'
      WRITE (*, *) '4. SELECT DATA SET FOR BURN RATE LAWS AND EXIT'
      READ (*, *) IOPT
      IF (IOPT .EQ. 4) GO TO 1120
      IF (IOPT .EQ. 1) GO TO 1130
      IF (IOPT .EQ. 2) GO TO 1140
      IF (IOPT .EQ. 3) GO TO 1150
1150  CONTINUE
      DO 1160 I = 1, NPT
        C (I) = REAL (A (I))
        IF (B (I) .LT. 0.0) B (I) = - B (I)
        IF (B (I) .LE. 0.1) B (I) = .1001
        IF (C (I) .LT. 0.0) C (I) = - C (I)
        IF (C (I) .LE. 0.1) C (I) = .1001

```

```

        IF (D (I) .LT. 0.0) D (I) = - D (I)
        IF (D (I) .LE. 0.1) D (I) = .1001
1160 CONTINUE
        XAXISL = 'PRESSURE (MPa)'
        YAXISL = 'BURNING RATE (cm/s)'
        TITLE = 'BURNING RATE PLOT '//A1 (10)
        CALL PL88OP (B, C, D, NPT, XAXISL, YAXISL, TITLE, MODEL)
        GO TO 1080
1130 CONTINUE
        DO 1170 I = 1, NPT
            IF (B (I) .LT. 0.0) B (I) = - B (I)
            IF (B (I) .LE. 0.1) B (I) = .1001
            IF (D (I) .LT. 0.0) D (I) = - D (I)
            IF (D (I) .LE. 0.1) D (I) = .1001
1170 CONTINUE
        XAXISL = 'PRESSURE (MPa)'
        YAXISL = 'BURNING RATE (cm/s)'
        TITLE = 'BURNING RATE PLOT '//A1 (10)
        CALL PL88LG (B, D, NPT, XAXISL, YAXISL, TITLE, MODEL)
        GO TO 1080
1140 CONTINUE
        DO 1180 I = 1, NPT
            C (I) = REAL (A (I))
            IF (B (I) .LT. 0.0) B (I) = - B (I)
            IF (B (I) .LE. 0.1) B (I) = .1001
            IF (C (I) .LT. 0.0) C (I) = - C (I)
            IF (C (I) .LE. 0.1) C (I) = .1001
1180 CONTINUE
        XAXISL = 'PRESSURE (MPa)'
        YAXISL = 'BURNING RATE (cm/s)'
        TITLE = 'BURNING RATE PLOT '//A1 (10)
        CALL PL88LG (B, C, NPT, XAXISL, YAXISL, TITLE, MODEL)
        GO TO 1080
1120 CONTINUE
        CALL CLEAR
C***** ATTEMPT TO GET GRAPH EVEN WHEN NO FILTERING IS DONE *****
        IF (IFLAG .EQ. 1) GO TO 1190
        WRITE (*, 6010)
6010 FORMAT(/,10X,'Select the data set to be utilized in computing',
1/,10X,'the burn rate laws and for the graphics file.',
2//,10X,'1. Original Data'/,10X,'2. Filtered Data'//,
310X,'Enter Selection.')
        READ (*, *) ISELET
        IF (ISELET .EQ. 2) THEN
            OPEN (UNIT = 14, FILE = A1 (17))
            REWIND (UNIT = 14)
            TEMP1 = 0.0
C*****

```

```

C****1***** FFT'S PACK EXTRA DATA POINTS AT BEGINNING TO *****
C***** FORCE THE NUMBER OF POINTS TO BE A MULTIPLE *****
C***** OF 2, NEED TO REMOVE THOSE EXTRA POINTS *****
C*****
      DO 1200 J = 2, NPT
        DELTA = ABS (DATA (J, 1) - DATA (J - 1, 1))
        IF (DELTA .GT. .1) THEN
          NSTART = J - 1
          GO TO 1210
        END IF
1200   CONTINUE
        NSTART = 1
1210   CONTINUE
        DO 1220 I = NSTART, NPT
          TEMP = REAL (A (I))
          IF (TEMP .LE. 0.1) TEMP = .1
          WRITE (14, *) DATA (I, 1), TEMP, TEMP1
1220   CONTINUE
          CLOSE (UNIT = 14)
        END IF
1190 CONTINUE
      RETURN
      END
      SUBROUTINE FFT (A, M, N, SIGN, SCALE)
      COMPLEX A (N), U, W, T
      PI = 3.1415926535
      N = 2**M
      NV2 = N/2
      NM1 = N - 1
      J = 1
      ITK = 0
      DO 1000 I = 1, N - 1
        ITK = ITK + 1
        IF (I .GE. J) GO TO 1010
        T = A (J)
        A (J) = A (I)
        A (I) = T
1010   CONTINUE
        K = NV2
1020   CONTINUE
        IF (K .GE. J) GO TO 1000
        J = J - K
        K = K/2
        GO TO 1020
        J = J + K
1000 CONTINUE
      ITK = 0
      DO 1030 L = 1, M

```

```

      LE = 2**L
      LE1 = LE/2
      U = CMPLX (1.0, 0.0)
      W = CMPLX (cos (PI/LE1), SIGN*sin (PI/LE1))
      DO 1030 J = 1, LE1
        DO 1040 I = J, N, LE
          ITK = ITK + 1
          IP = I + LE1
          T = A (IP)*U
          A (IP) = A (I) - T
          A (I) = A (I) + T
1040    CONTINUE
        U = U*W
1030 CONTINUE
      ITK = 0
      DO 1050 I = 1, N
        A (I) = A (I)*SCALE
        ITK = ITK + 1
1050 CONTINUE
      RETURN
      END
      SUBROUTINE PL88P (X, Y, NPTS, XAXISH, YAXISH, TITLEH, MODEL)
      DIMENSION X (1), Y (1)
      CHARACTER*20 XAXISH, YAXISH
      CHARACTER*40 TITLEH
      CHARACTER*20 YAXISL
      CHARACTER*40 TITLE
      CHARACTER*20 XAXISL
      CHARACTER*1 TITA (40), XAXISA (20), YAXISA (20)
      EQUIVALENCE (TITA (1), TITLE), (XAXISA (1),
1    XAXISL), (YAXISA (1), YAXISL)
      XAXISL = XAXISH
      YAXISL = YAXISH
      TITLE = TITLEH
      DO 1000 I = 1, 20
        KI = 21 - I
        IF (XAXISA (KI) .NE. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
        NXA = KI
        DO 1020 I = 1, 20
          KI = 21 - I
          IF (YAXISA (KI) .NE. ' ') GO TO 1030
1020 CONTINUE
1030 CONTINUE
          NYA = KI
          DO 1040 I = 1, 40
            KI = 41 - I

```

```

        IF (TTTA (KI) .NE. ' ') GO TO 1050
1040 CONTINUE
1050 CONTINUE
        NTA = KI
        WRITE (*, 6000)
        WRITE (*, 6010)
6010 FORMAT('O DO YOU WISH TO HAVE A HARDCOPY? (1 FOR YES, 0 FOR NO)')
        READ (*, 5000) ICYN
5000 FORMAT(I1)
6000 FORMAT('O STRIKE ENTER WHEN FINISHED WITH PLOT')
        CALL PLOTS (0, 97, 97)
        CALL FACTOR (1.0)
        HT = .25
        CALL PLOT (2.50, 1.50, - 3)
        XT = 2.5 - HT*5./8.*.5*NTA
        CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
        CALL SCALE (X, 5.00, NPTS, 1)
        CALL SCALE (Y, 5.00, NPTS, 1)
        CALL STAXIS (.25, .25, .111, .112, 2)
        CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X1
            (NPTS + 1), X (NPTS + 2))
        CALL STAXIS (.25, .25, .111, .112, - 1)
        CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y1
            (NPTS + 1), Y (NPTS + 2))
        CALL LINE (X, Y, NPTS, 1, 0, 0)
        CALL PLOT (0.0, 0.0, 999)
C      READ(*,310)CH
C 310  FORMAT(A1)
        IF (ICYN .EQ. 0) GO TO 1060
        CALL PLOTS (0, 1, MODEL)
        CALL FACTOR (1.0)
        HT = .25
        CALL PLOT (2.50, 1.50, - 3)
        XT = 2.5 - HT*5./8.*.5*NTA
        CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
        CALL SCALE (X, 5.00, NPTS, 1)
        CALL SCALE (Y, 5.00, NPTS, 1)
        CALL STAXIS (.25, .25, .111, .112, 2)
        CALL AXIS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
        CALL STAXIS (.25, .25, .111, .112, - 1)
        CALL AXIS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
        CALL LINE (X, Y, NPTS, 1, 0, 0)
        CALL PLOT (0.0, 0.0, 999)
1060 CONTINUE
        RETURN
        END

```

```

SUBROUTINE PL88LG (X, Y, NPTS, XAXISH, YAXISH, TITLEH, MODEL)
DIMENSION X (1), Y (1)
CHARACTER*20 XAXISH, YAXISH
CHARACTER*40 TITLEH
CHARACTER*20 YAXISL
CHARACTER*40 TITLE
CHARACTER*20 XAXISL
CHARACTER*1 TITA (40), XAXISA (20), YAXISA (20)
EQUIVALENCE (TITA (1), TITLE), (XAXISA (1),
1  XAXISL), (YAXISA (1), YAXISL)
XAXISL = XAXISH
YAXISL = YAXISH
TITLE = TITLEH
DO 1000 I = 1, 20
    KI = 21 - I
    IF (XAXISA (KI) .NE. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
    NXA = KI
    DO 1020 I = 1, 20
        KI = 21 - I
        IF (YAXISA (KI) .NE. ' ') GO TO 1030
1020 CONTINUE
1030 CONTINUE
    NYA = KI
    DO 1040 I = 1, 40
        KI = 41 - I
        IF (TITA (KI) .NE. ' ') GO TO 1050
1040 CONTINUE
1050 CONTINUE
    NTA = KI
    WRITE (*, 6000)
    WRITE (*, 6010)
6010 FORMAT('O DO YOU WISH TO HAVE A HARDCOPY? ( 1 FOR YES,0 FOR NO)')
    READ (*, 5000) ICYN
5000 FORMAT(I1)
6000 FORMAT('O STRIKE ENTER WHEN FINISHED WITH PLOT')
    CALL PLOTS (0, 97, 97)
    CALL FACTOR (1.0)
    HT = .25
    CALL PLOT (2.50, 1.50, - 3)
    XT = 2.5 - HT*5./8.*.5*NTA
    CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
    CALL SCALG (X, 5.00, NPTS, 1)
    CALL SCALG (Y, 5.00, NPTS, 1)
    CALL STAXIS (.25, .25, .111, .112, 2)
    CALL LGAXS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))

```

```

CALL STAXIS (.25, .25, .111, .112, - 1)
CALL LGAXS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
CALL LGLIN (X, Y, NPTS, 1, 0, 0, 0)
CALL PLOT (0.0, 0.0, 999)
C   READ(*,310)CH
C 310  FORMAT(A1)
      IF (ICYN .EQ. 0) GO TO 1060
      CALL PLOTS (0, 1, MODEL)
      CALL FACTOR (1.0)
      HT = .25
      CALL PLOT (2.50, 1.50, - 3)
      XT = 2.5 - HT*5./8.*.5*NTA
      CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
      CALL SCALG (X, 5.00, NPTS, 1)
      CALL SCALG (Y, 5.00, NPTS, 1)
      CALL STAXIS (.25, .25, .111, .112, 2)
      CALL LGAXS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
      CALL STAXIS (.25, .25, .111, .112, - 1)
      CALL LGAXS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
      CALL LGLIN (X, Y, NPTS, 1, 0, 0, 0)
      CALL PLOT (0.0, 0.0, 999)
1060 CONTINUE
      RETURN
      END
      SUBROUTINE PL88OP (X, Y, Z, NPTS, XAXISH, YAXISH, TITLEH, MODEL)
      DIMENSION X (1), Y (1), Z (1)
      CHARACTER*20 XAXISH, YAXISH
      CHARACTER*40 TITLEH
      CHARACTER*20 YAXISL
      CHARACTER*40 TITLE
      CHARACTER*20 XAXISL
      CHARACTER*1 TITA (40), XAXISA (20), YAXISA (20)
      EQUIVALENCE (TITA (1), TITLE), (XAXISA (1), XAXISL), (YAXISA (1), YAXISL)
      XAXISL = XAXISH
      YAXISL = YAXISH
      TITLE = TITLEH
      DO 1000 I = 1, 20
        KI = 21 - I
        IF (XAXISA (KI) .NE. ' ') GO TO 1010
1000 CONTINUE
1010 CONTINUE
      NXA = KI
      DO 1020 I = 1, 20
        KI = 21 - I

```

```

        IF (YAXISA (KI) .NE. ' ') GO TO 1030
1020 CONTINUE
1030 CONTINUE
        NYA = KI
        DO 1040 I = 1, 40
            KI = 41 - I
            IF (TITA (KI) .NE. ' ') GO TO 1050
1040 CONTINUE
1050 CONTINUE
        NTA = KI
        WRITE (*, 6000)
        WRITE (*, 6010)
6010 FORMAT('O DO YOU WISH TO HAVE A HARDCOPY? (1 FOR YES, 0 FOR NO)')
        READ (*, 5000) ICYN
5000 FORMAT(I1)
6000 FORMAT('O STRIKE ENTER WHEN FINISHED WITH PLOT')
        CALL PLOTS (0, 97, 97)
        CALL FACTOR (1.0)
        HT = .25
        CALL PLOT (2.50, 1.50, - 3)
        XT = 2.5 - HT*5./8.*.5*NTA
        CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
        CALL SCALG (X, 5.00, NPTS, 1)
        CALL SCALG (Y, 5.00, NPTS, 1)
        CALL SCALG (Z, 5.00, NPTS, 1)
C*****
C***** MAKE SCALE FOR Y & Z IDENTICAL *****
C*****
        START = MIN (Y (NPTS + 1), Z (NPTS + 1))
        Y (NPTS + 1) = START
        Z (NPTS + 1) = START
        START = MAX (Y (NPTS + 2), Z (NPTS + 2))
        Y (NPTS + 2) = START
        Z (NPTS + 2) = START
        CALL STAXIS (.25, .25, .111, .112, 2)
        CALL LGAXS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
        CALL STAXIS (.25, .25, .111, .112, - 1)
        CALL LGAXS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
        CALL LGLIN (X, Y, NPTS, 1, 0, 0, 0)
        CALL LGLIN (X, Z, NPTS, 1, 0, 0, 0)
        CALL PLOT (0.0, 0.0, 999)
C    READ(*,310)CH
C 310  FORMAT(A1)
        IF (ICYN .EQ. 0) GO TO 1060
        CALL PLOTS (0, 1, MODEL)
        CALL FACTOR (1.0)

```

```

HT = .25
CALL PLOT (2.50, 1.50, - 3)
XT = 2.5 - HT*5./8.*5*NTA
CALL SYMBOL (XT, 5.6, HT, TITLE, 0., NTA)
CALL SCALG (X, 5.00, NPTS, 1)
CALL SCALG (Y, 5.00, NPTS, 1)
CALL SCALG (Z, 5.00, NPTS, 1)
CALL STAXIS (.25, .25, .111, .112, 2)
CALL LGAXS (0., 0., XAXISL, - NXA, 5.00, 0., X (NPTS + 1), X (NPTS + 2))
CALL STAXIS (.25, .25, .111, .112, - 1)
CALL LGAXS (0., 0., YAXISL, NYA, 5.0, 90., Y (NPTS + 1), Y (NPTS + 2))
CALL LGLIN (X, Y, NPTS, 1, 0, 0, 0)
CALL LGLIN (X, Z, NPTS, 1, 0, 0, 0)
CALL PLOT (0.0, 0.0, 999)
1060 CONTINUE
RETURN
END
C*****
C***** SUBROUTINE BRNRT *****
C*****
SUBROUTINE BRNRT (A1, IOUTCH)
C*****
C PROVIDED BY F. ROBBINS; 12/28/91
C MODIFIED BY W. OBERLE; 12/28/91
C NOW CALCULATE LEAST SQUARES FIT OF DXDT=A*P^N
C OVER PRESSURE RANGE PLOW TO PHIGH FOR AS MANY
C RANGES AS REQUIRED
C*****
CHARACTER*20 A1 (20)
DIMENSION XRATES (100, 5), PRE (1024), RT (1024)
C*****
C***** FIRST THE PREDETERMINED RATES ARE LISTED *****
C*****
OPEN (UNIT = 5, FILE = A1 (18))
READ (5, *) NN
DO 1000 I = 1, NN
READ (5, *) XRATES (I, 1)
READ (5, *) XRATES (I, 2)
READ (5, *) DUMMY
READ (5, *) DUMMY
READ (5, *) DUMMY
1000 CONTINUE
CLOSE (UNIT = 5)
IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
WRITE (7, 6000)
END IF

```

```

      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6000)
      END IF
6000 FORMAT('1',10X,'Table of Burn Rate Laws'//)
1010 CONTINUE
      CALL CLEAR
      WRITE (*, 6010)
6010 FORMAT(/,10X,'The current pressure regions over which the',
1/,10X,'burn rate laws will be computed are given below.',
2/,10X,'Pressure',10X,'Pressure'./)
      DO 1020 I = 1, NN
        WRITE (*, 6020) XRATES (I, 1), XRATES (I, 2)
1020 CONTINUE
6020 FORMAT(' ',2F18.4)
1030 CONTINUE
      WRITE (*, 6030)
6030 FORMAT(/,' DO YOU WANT TO ADD ANOTHER PRESSURE RANGE',
1/, ' FOR BURN RATE CALCULATION? (Yes=1, No=2)')
      READ (*, 5000) ICYN
5000 FORMAT(I1)
      IF (ICYN .EQ. 2) GO TO 1040
      CALL CLEAR
      WRITE (*, 6040)
6040 FORMAT(' ENTER LOW AND HIGH PRESSURES: (MPA) ')
      READ (*, *) PLOW, PHIGH
      NN = NN + 1
      XRATES (NN, 1) = PLOW
      XRATES (NN, 2) = PHIGH
      GO TO 1010
1040 CONTINUE
C*****
C***** HEADINGS ARE WRITTEN *****
C*****
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6050)
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6050)
      END IF
6050 FORMAT(' ',8X,'RANGE'/7X,'PLOW  PHIGH',9X,'COEF',12X,
1 'EXP',8X,'CORR COEF'/8X,'MPA',4X,'MPA',7X,'CM/S-MPA^EXP',
2 9X,'-',13X,'-'/)
C*****
C***** GRAPHICS FILE OPENED & READ *****
C*****
      OPEN (UNIT = 4, FILE = A1 (17), STATUS = 'OLD')
      REWIND (UNIT = 4)
      JJ = 1

```

```

1050 CONTINUE
  READ (4, *, END = 1060) PRE (JJ), RT (JJ), XXX
  JJ = JJ + 1
  GO TO 1050
1060 CONTINUE
  CLOSE (UNIT = 4)
  JJ = JJ - 1
C*****
C***** NOW LAWS ARE COMPUTED *****
C*****
  DO 1070 I = 1, NN
    NLOW = 1
    NHIGH = JJ
    PLOW = XRATES (I, 1)
    PHIGH = XRATES (I, 2)
    DO 1080 K = 1, JJ
      IF (PRE (K) .GE. PLOW) THEN
        NLOW = K
        GO TO 1090
      END IF
1080  CONTINUE
      WRITE (*, *) ' NO PRESSURE ABOVE: ', PLOW
      WRITE (*, *) ' WILL GO TO NEXT STEP'
      PAUSE
      GO TO 1070
1090  CONTINUE
      DO 1100 K = NLOW + 1, JJ
        IF (PRE (K) .GT. PHIGH) THEN
          NHIGH = K - 1
          GO TO 1110
        END IF
1100  CONTINUE
        NHIGH = JJ
1110  CONTINUE
        A7 = 0.
        A8 = 0.
        A9 = 0.
        H5 = 0.
        H8 = 0.
        H9 = 0.
        DO 1120 K = NLOW, NHIGH
          P = PRE (K)
          DXDT = RT (K)
          A7 = A7 + ALOG10 (P)
          A8 = A8 + ALOG10 (P)*ALOG10 (P)
          A9 = A9 + ALOG10 (P)*ALOG10 (DXDT)
          H5 = H5 + ALOG10 (DXDT)
          H8 = H8 + ALOG10 (DXDT)*ALOG10 (DXDT)

```

```

      H9 = H9 + 1
1120  CONTINUE
      B5 = (A8*H5 - A7*A9)/(H9*A8 - A7*A7)
      B6 = (H9*A9 - A7*H5)/(H9*A8 - A7*A7)
      B5 = 10.**B5
      R6 = (H9*A9 - H5*A7)/SQRT ((H9*A8 - A7*A7)*(H9*H8 - H5*H5))
      XRATES (I, 3) = B5
      XRATES (I, 4) = B6
      XRATES (I, 5) = R6
      IF ((IOUTCH .EQ. 1) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (7, 6060) PLOW, PHIGH, B5, B6, R6
      END IF
      IF ((IOUTCH .EQ. 2) .OR. (IOUTCH .EQ. 3)) THEN
        WRITE (2, 6060) PLOW, PHIGH, B5, B6, R6
      END IF
C 460  CONTINUE
1070  CONTINUE
6060  FORMAT(4X,F7.0,1X,F7.0,5X,E12.6,5X,F9.6,5X,F9.6)
C*****
C***** WRITING BURN RATE FILE *****
C*****
      OPEN (UNIT = 4, FILE = A1 (18))
      REWIND (UNIT = 4)
      WRITE (4, *) NN
      DO 1130 I = 1, NN
        WRITE (4, *) XRATES (I, 1)
        WRITE (4, *) XRATES (I, 2)
        WRITE (4, *) XRATES (I, 3)
        WRITE (4, *) XRATES (I, 4)
        WRITE (4, *) XRATES (I, 5)
1130  CONTINUE
      CLOSE (UNIT = 4)
      RETURN
      END

```

INTENTIONALLY LEFT BLANK.

APPENDIX M:
SMOOTHING AND DIFFERENTIATION COEFFICIENTS

INTENTIONALLY LEFT BLANK.

SMOOTHING COEFFICIENTS

BRIDGE LENGTH: N = 5

-8.571428571428E-02
3.428571428571E-01
4.857142857143E-01
3.428571428571E-01
-8.571428571428E-02

BRIDGE LENGTH: N = 7

-9.523809523809E-02
1.428571428571E-01
2.857142857143E-01
3.333333333333E-01
2.857142857143E-01
1.428571428571E-01
-9.523809523809E-02

BRIDGE LENGTH: N = 9

-9.090909090909E-02
6.060606060606E-02
1.688311688312E-01
2.337662337662E-01
2.554112554113E-01
2.337662337662E-01
1.688311688312E-01
6.060606060606E-02
-9.090909090909E-02

BRIDGE LENGTH: N = 11

-8.391608391608E-02
2.097902097902E-02
1.025641025641E-01
1.608391608392E-01
1.958041958042E-01
2.074592074592E-01
1.958041958042E-01
1.608391608392E-01
1.025641025641E-01
2.097902097902E-02
-8.391608391608E-02

BRIDGE LENGTH: N = 13

-7.692307692307E-02
8.881784197001E-16
6.293706293706E-02
1.118881118881E-01
1.468531468531E-01
1.678321678322E-01
1.748251748252E-01
1.678321678322E-01
1.468531468531E-01
1.118881118881E-01
6.293706293706E-02
8.881784197001E-16
-7.692307692307E-02

BRIDGE LENGTH: N = 15

-7.058823529411E-02
-1.176470588235E-02
3.800904977376E-02
7.873303167421E-02
1.104072398190E-01
1.330316742081E-01
1.466063348416E-01
1.511312217195E-01
1.466063348416E-01
1.330316742081E-01
1.104072398190E-01
7.873303167421E-02
3.800904977376E-02
-1.176470588235E-02
-7.058823529411E-02

BRIDGE LENGTH: N = 17

-6.501547987616E-02
-1.857585139319E-02
2.167182662539E-02
5.572755417957E-02
8.359133126935E-02
1.052631578947E-01
1.207430340557E-01
1.300309597523E-01
1.331269349845E-01
1.300309597523E-01
1.207430340557E-01

1.052631578947E-01
8.359133126935E-02
5.572755417957E-02
2.167182662539E-02
-1.857585139319E-02
-6.501547987616E-02

BRIDGE LENGTH: N = 19

-6.015037593985E-02
-2.255639097744E-02
1.061477222468E-02
3.936311366652E-02
6.368863334808E-02
8.359133126935E-02
9.907120743034E-02
1.101282618310E-01
1.167624944715E-01
1.189739053516E-01
1.167624944715E-01
1.101282618310E-01
9.907120743034E-02
8.359133126935E-02
6.368863334808E-02
3.936311366652E-02
1.061477222468E-02
-2.255639097744E-02
-6.015037593985E-02

BRIDGE LENGTH: N = 21

-5.590062111801E-02
-2.484472049689E-02
2.942137953580E-03
2.745995423341E-02
4.870872834260E-02
6.668846028114E-02
8.139915004904E-02
9.284079764629E-02
1.010134030729E-01
1.059169663289E-01
1.075514874142E-01
1.059169663289E-01
1.010134030729E-01
9.284079764629E-02
8.139915004904E-02
6.668846028114E-02
4.870872834260E-02

2.745995423341E-02
2.942137953580E-03
-2.484472049689E-02
-5.590062111801E-02

BRIDGE LENGTH: N = 23

-5.217391304348E-02
-2.608695652174E-02
-2.484472049689E-03
1.863354037267E-02
3.726708074534E-02
5.341614906832E-02
6.708074534161E-02
7.826086956522E-02
8.695652173913E-02
9.316770186335E-02
9.689440993789E-02
9.813664596273E-02
9.689440993789E-02
9.316770186335E-02
8.695652173913E-02
7.826086956522E-02
6.708074534161E-02
5.341614906832E-02
3.726708074534E-02
1.863354037267E-02
-2.484472049689E-03
-2.608695652174E-02
-5.217391304348E-02

BRIDGE LENGTH: N = 25

-4.888888888889E-02
-2.666666666667E-02
-6.376811594202E-03
1.198067632850E-02
2.840579710145E-02
4.289855072464E-02
5.545893719807E-02
6.608695652174E-02
7.478260869565E-02
8.154589371981E-02
8.637681159420E-02
8.927536231884E-02
9.024154589372E-02
8.927536231884E-02
8.637681159420E-02

8.154589371981E-02
7.478260869565E-02
6.608695652174E-02
5.545893719807E-02
4.289855072464E-02
2.840579710145E-02
1.198067632850E-02
-6.376811594202E-03
-2.666666666667E-02
-4.888888888889E-02

BRIDGE LENGTH: N = 27

-4.597701149425E-02
-2.681992337165E-02
-9.195402298849E-03
6.896551724139E-03
2.145593869732E-02
3.448275862069E-02
4.597701149425E-02
5.593869731801E-02
6.436781609195E-02
7.126436781609E-02
7.662835249042E-02
8.045977011494E-02
8.275862068965E-02
8.352490421456E-02
8.275862068965E-02
8.045977011494E-02
7.662835249042E-02
7.126436781609E-02
6.436781609195E-02
5.593869731801E-02
4.597701149425E-02
3.448275862069E-02
2.145593869732E-02
6.896551724139E-03
-9.195402298849E-03
-2.681992337165E-02
-4.597701149425E-02

BRIDGE LENGTH: N = 29

-4.338153503893E-02
-2.669632925473E-02
-1.124706463972E-02
2.966258806081E-03
1.594364108268E-02

2.768508219009E-02
3.819058212829E-02
4.746014089729E-02
5.549375849709E-02
6.229143492770E-02
6.785317018910E-02
7.217896428130E-02
7.526881720430E-02
7.712272895810E-02
7.774069954270E-02
7.712272895810E-02
7.526881720430E-02
7.217896428130E-02
6.785317018910E-02
6.229143492770E-02
5.549375849709E-02
4.746014089729E-02
3.819058212829E-02
2.768508219009E-02
1.594364108268E-02
2.966258806081E-03
-1.124706463972E-02
-2.669632925473E-02
-4.338153503893E-02

BRIDGE LENGTH: N = 31

-4.105571847507E-02
-2.639296187683E-02
-1.274142987157E-02
-1.011224592977E-04
1.152796036000E-02
2.214581858631E-02
3.175245221964E-02
4.034786125999E-02
4.793204570735E-02
5.450500556173E-02
6.006674082314E-02
6.461725149156E-02
6.815653756699E-02
7.068459904945E-02
7.220143593892E-02
7.270704823541E-02
7.220143593892E-02
7.068459904945E-02
6.815653756699E-02
6.461725149156E-02
6.006674082314E-02

5.450500556173E-02
4.793204570735E-02
4.034786125999E-02
3.175245221964E-02
2.214581858631E-02
1.152796036000E-02
-1.011224592977E-04
-1.274142987157E-02
-2.639296187683E-02
-4.105571847507E-02

BRIDGE LENGTH: N = 33

-3.896103896104E-02
-2.597402597403E-02
-1.382488479263E-02
-2.513615416841E-03
7.959782153331E-03
1.759530791789E-02
2.639296187683E-02
3.435274403016E-02
4.147465437788E-02
4.775869291998E-02
5.320485965647E-02
5.781315458735E-02
6.158357771261E-02
6.451612903226E-02
6.661080854629E-02
6.786761625471E-02
6.828655215752E-02
6.786761625471E-02
6.661080854629E-02
6.451612903226E-02
6.158357771261E-02
5.781315458735E-02
5.320485965647E-02
4.775869291998E-02
4.147465437788E-02
3.435274403016E-02
2.639296187683E-02
1.759530791789E-02
7.959782153331E-03
-2.513615416841E-03
-1.382488479263E-02
-2.597402597403E-02
-3.896103896104E-02

BRIDGE LENGTH: N = 35

-3.706563706564E-02
-2.548262548263E-02
-1.460161460161E-02
-4.422604422604E-03
5.054405054405E-03
1.382941382941E-02
2.190242190242E-02
2.927342927343E-02
3.594243594244E-02
4.190944190944E-02
4.717444717445E-02
5.173745173745E-02
5.559845559846E-02
5.875745875746E-02
6.121446121446E-02
6.296946296946E-02
6.402246402246E-02
6.437346437346E-02
6.402246402246E-02
6.296946296946E-02
6.121446121446E-02
5.875745875746E-02
5.559845559846E-02
5.173745173745E-02
4.717444717445E-02
4.190944190944E-02
3.594243594244E-02
2.927342927343E-02
2.190242190242E-02
1.382941382941E-02
5.054405054405E-03
-4.422604422604E-03
-1.460161460161E-02
-2.548262548263E-02
-3.706563706564E-02

DIFFERENTIATION COEFFICIENTS

BRIDGE LENGTH: N = 5

-2.000000000000E-01
-1.000000000000E-01
0.0
1.000000000000E-01
2.000000000000E-01

BRIDGE LENGTH: N = 7

-1.071428571429E-01
-7.142857142857E-02
-3.571428571429E-02
0.0
3.571428571429E-02
7.142857142857E-02
1.071428571429E-01

BRIDGE LENGTH: N = 9

-6.66666666667E-02
-5.00000000000E-02
-3.33333333333E-02
-1.66666666667E-02
0.0
1.66666666667E-02
3.33333333333E-02
5.00000000000E-02
6.66666666667E-02

BRIDGE LENGTH: N = 11

-4.54545454545E-02
-3.63636363636E-02
-2.72727272727E-02
-1.81818181818E-02
-9.09090909090E-03
0.0
9.09090909090E-03
1.81818181818E-02
2.72727272727E-02
3.63636363636E-02
4.54545454545E-02

BRIDGE LENGTH: N = 13

-3.29670329670E-02
-2.74725274725E-02
-2.19780219780E-02
-1.64835164835E-02
-1.09890109890E-02
-5.49450549450E-03
0.0
5.49450549450E-03

1.098901098901E-02
1.648351648352E-02
2.197802197802E-02
2.747252747253E-02
3.296703296703E-02

BRIDGE LENGTH: N = 15

-2.500000000000E-02
-2.142857142857E-02
-1.785714285714E-02
-1.428571428571E-02
-1.071428571429E-02
-7.142857142857E-03
-3.571428571429E-03
0.0
3.571428571429E-03
7.142857142857E-03
1.071428571429E-02
1.428571428571E-02
1.785714285714E-02
2.142857142857E-02
2.500000000000E-02

BRIDGE LENGTH: N = 17

-1.960784313725E-02
-1.715686274510E-02
-1.470588235294E-02
-1.225490196078E-02
-9.803921568627E-03
-7.352941176471E-03
-4.901960784314E-03
-2.450980392157E-03
0.0
2.450980392157E-03
4.901960784314E-03
7.352941176471E-03
9.803921568627E-03
1.225490196078E-02
1.470588235294E-02
1.715686274510E-02
1.960784313725E-02

BRIDGE LENGTH: N = 19

-1.578947368421E-02
-1.403508771930E-02

-1.228070175439E-02
-1.052631578947E-02
-8.771929824561E-03
-7.017543859649E-03
-5.263157894737E-03
-3.508771929825E-03
-1.754385964912E-03

0.0

1.754385964912E-03
3.508771929825E-03
5.263157894737E-03
7.017543859649E-03
8.771929824561E-03
1.052631578947E-02
1.228070175439E-02
1.403508771930E-02
1.578947363421E-02

BRIDGE LENGTH: N = 21

-1.298701298701E-02
-1.158831168831E-02
-1.038961038961E-02
-9.090909090909E-03
-7.792207792208E-03
-6.493506493506E-03
-5.194805194805E-03
-3.896103896104E-03
-2.597402597403E-03
-1.298701298701E-03

0.0

1.298701298701E-03
2.597402597403E-03
3.896103896104E-03
5.194805194805E-03
6.493506493506E-03
7.792207792208E-03
9.090909090909E-03
1.038961038961E-02
1.168831168831E-02
1.298701298701E-02

BRIDGE LENGTH: N = 23

-1.086956521739E-02
-9.881422924901E-03
-8.893280632411E-03
-7.905138339921E-03

-6.916996047431E-03
-5.928853754941E-03
-4.940711462451E-03
-3.952569169960E-03
-2.964426877470E-03
-1.976284584980E-03
-9.881422924901E-04
0.0
9.881422924901E-04
1.976284584980E-03
2.964426877470E-03
3.952569169960E-03
4.940711462451E-03
5.928853754941E-03
6.916996047431E-03
7.905138339921E-03
8.893280632411E-03
9.881422924901E-03
1.086956521739E-02

BRIDGE LENGTH: N = 25

-9.230769230769E-03
-8.461538461538E-03
-7.692307692308E-03
-6.923076923077E-03
-6.153846153846E-03
-5.384615384615E-03
-4.615384615385E-03
-3.846153846154E-03
-3.076923076923E-03
-2.307692307692E-03
-1.538461538462E-03
-7.692307692308E-04
0.0
7.692307692308E-04
1.538461538462E-03
2.307692307692E-03
3.076923076923E-03
3.846153846154E-03
4.615384615385E-03
5.384615384615E-03
6.153846153846E-03
6.923076923077E-03
7.692307692308E-03
8.461538461538E-03
9.230769230769E-03

BRIDGE LENGTH: N = 27

-7.936507936508E-03
-7.326007325007E-03
-6.715506715507E-03
-6.105006105006E-03
-5.494505494505E-03
-4.884004884005E-03
-4.273504273504E-03
-3.663003663004E-03
-3.052503052503E-03
-2.442002442002E-03
-1.831501831502E-03
-1.221001221001E-03
-6.105006105006E-04
0.0
6.105006105006E-04
1.221001221001E-03
1.831501831502E-03
2.442002442002E-03
3.052503052503E-03
3.663003663004E-03
4.273504273504E-03
4.884004884005E-03
5.494505494505E-03
6.105006105006E-03
6.715506715507E-03
7.326007326007E-03
7.936507936508E-03

BRIDGE LENGTH: N = 29

-6.896551724138E-03
-6.403940886699E-03
-5.911330049261E-03
-5.418719211823E-03
-4.926108374384E-03
-4.433497536946E-03
-3.940886699507E-03
-3.448275862069E-03
-2.955665024631E-03
-2.463054187192E-03
-1.970443349754E-03
-1.477832512315E-03
-9.852216748768E-04
-4.926108374384E-04
0.0
4.926108374384E-04

9.852216748768E-04
1.477832512315E-03
1.970443349754E-03
2.463054187192E-03
2.955665024631E-03
3.448275862069E-03
3.940886699507E-03
4.433497536946E-03
4.926108374384E-03
5.418719211823E-03
5.911330049261E-03
6.403940886699E-03
6.896551724138E-03

BRIDGE LENGTH: N = 31

-6.048387096774E-03
-5.645161290323E-03
-5.241935483871E-03
-4.838709677419E-03
-4.435483870968E-03
-4.032258064516E-03
-3.629032258065E-03
-3.225806451613E-03
-2.822580645161E-03
-2.419354838710E-03
-2.016129032258E-03
-1.612903225806E-03
-1.209677419355E-03
-8.064516129032E-04
-4.032258064516E-04
0.0
4.032258064516E-04
8.064516129032E-04
1.209677419355E-03
1.612903225806E-03
2.016129032258E-03
2.419354838710E-03
2.822580645161E-03
3.225806451613E-03
3.629032258065E-03
4.032258064516E-03
4.435483870968E-03
4.838709677419E-03
5.241935483871E-03
5.645161290323E-03
6.048387096774E-03

BRIDGE LENGTH: N = 33

-5.34759358288E-03
-5.013368983957E-03
-4.679144385027E-03
-4.344919786096E-03
-4.010695187166E-03
-3.676470588235E-03
-3.342245989305E-03
-3.008021390374E-03
-2.673796791444E-03
-2.339572192513E-03
-2.005347593583E-03
-1.671122994652E-03
-1.336898395722E-03
-1.002673796791E-03
-6.684491978610E-04
-3.342245989305E-04
0.0
3.342245989305E-04
6.684491978610E-04
1.002673796791E-03
1.336898395722E-03
1.671122994652E-03
2.005347593583E-03
2.339572192513E-03
2.673796791444E-03
3.008021390374E-03
3.342245989305E-03
3.676470588235E-03
4.010695187166E-03
4.344919786096E-03
4.679144385027E-03
5.013368983957E-03
5.347593582888E-03

BRIDGE LENGTH: N = 35

-4.761904761905E-03
-4.481792717087E-03
-4.201680672269E-03
-3.921568627451E-03
-3.641456582633E-03
-3.361344537815E-03
-3.081232492997E-03
-2.801120448179E-03
-2.521008403361E-03
-2.240896358543E-03

-1.960784313725E-03
-1.680672268908E-03
-1.400560224090E-03
-1.120448179272E-03
-8.403361344538E-04
-5.602240896359E-04
-2.801120448179E-04
0.0
2.801120448179E-04
5.602240896359E-04
8.403361344538E-04
1.120448179272E-03
1.400560224090E-03
1.680672268908E-03
1.960784313725E-03
2.240896358543E-03
2.521008403361E-03
2.801120448179E-03
3.081232492997E-03
3.361344537815E-03
3.641456582633E-03
3.921568627451E-03
4.201680672269E-03
4.481792717087E-03
4.761904761905E-03

<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: ASQNC-TAC-DIT (Technical Information Center) Warren, MI 48397-5000
1	Director U.S. Army Research Laboratory ATTN: AMSRL-D 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, Tech Publishing 2800 Powder Mill Rd. Adelphi, MD 20783-1145	1	Commandant U.S. Army Field Artillery School ATTN: ATSF-CSI Ft. Sill, OK 73503-5000
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-IMI-I Picatinny Arsenal, NJ 07806-5000	(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-TDC Picatinny Arsenal, NJ 07806-5000	(Unclass. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905-5660
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050	1	WL/MNOI Eglin AFB, FL 32542-5000 <u>Aberdeen Proving Ground</u>
(Unclass. only) 1	Commander U.S. Army Rock Island Arsenal ATTN: SMCRI-IMC-RT/Technical Library Rock Island, IL 61299-5000	2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Director U.S. Army Aviation Research and Technology Activity ATTN: SAVRT-R (Library) M/S 219-3 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-CI
		1	Dir, USARL ATTN: AMSRL-SL-I
		10	Dir, USARL ATTN: AMSRL-OP-CI-B (Tech Lib)

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-36 (Part II) Date of Report January 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, staple or tape closed, and mail.)