P-181

IN-37
11711

A PERFORMANCE MODEL OF A SPARK IGNITION WANKEL ENGINE: INCLUDING THE EFFECTS OF CREVICE VOLUMES, GAS LEAKAGE, AND HEAT TRANSFER

by

TIMOTHY JOHN NORMAN

B.S.M.E. University of Massachusetts, Amherst
(1981)

Submitted to the Department of
Mechanical Engineering
in Partial Fulfillment of the
Requirments of the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1983

© Massachusetts Institute of Technology

(NASA-CR-177239) A PERFORMANCE MODEL OF A SPARK IGNITION WANKEL ENGINE: INCLUDING THE EFFECTS OF CREVICE VOLUMES, GAS LEAKAGE, AND HEAT TRANSFER. VOLUME 2 M.S. Thesis (Massachusetts Inst. of Tech.) 181 p

N86-72140

11711

Unclas

00/37 43444

VOLUME 2

WANKEL CYCLE SIMULATION CODE

```
C
C
C
C                 ZERO-DIMENSIONAL SIMULATION OF THE
C
C
C                 SPARK-IGNITION WANKEL ENGINE CYCLE:
C
C
C                      A PERFORMANCE PREDICTIVE MODEL
C
C
C
C
C
C
C                                BY
C
C
C
C
C                           TIMOTHY J. NORMAN
C
C                           JOHN B. HEYWOOD
C
C                           STEPHEN G. POULOS
C
C                           S. HOSSEIN MANSOURI
C
C
C
C
C                      SLOAN AUTOMOTIVE LABORATORY
C
C
C                  MASSACHUSETTS INSTITUTE OF TECHNOLOGY
C
C
C
C
C
C                             FEBRUARY 1983
C
C
C
C
C
C
C
C      ADDRESS: TIMOTHY J. NORMAN
C               ROOM 3-339, 77 MASS. AVE.
C               MASSACHUSETTS INSTITUTE OF TECHNOLOGY
C               CAMBRIDGE, MASS  02139
C
```

```
C
C
C
C
C     PURPOSE
C
C        THE PROGRAM IS A ZERO-DIMENSIONAL SIMULATION OF THE
C
C        WANKEL ROTARY SPARK-IGNITION ENGINE OPERATING CYCLE.
C
C        THE PROGRAM CALCULATES TEMPERATURE AND PRESSURE IN THE
C
C        CHAMBER AND PREDICTS ENGINE PERFORMANCE AS A FUNCTION
C
C        OF ENGINE DESIGN AND OPERATING CONDITIONS.
C
C
C     DESCRIPTION OF PARAMETERS:
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C
C     1  GEOMETRICAL AND DESIGN PARAMETERS
C
C        ECCEN      YES    NO       ECCENTRICITY OF ROTOR (CM)
C        ROTRAD     YES    NO       RADIUS OF ROTOR (CM)
C        DEPTH      YES    NO       HEIGHT OF CHAMBER (CM)
C        VFLANK     YES    NO       VOLUME OF ROTOR POCKET (CM**3)
C        TIPO       YES    NO       INTAKE PORT OPENS (DEG)
C        TIPC       YES    NO       INTAKE PORT CLOSES (DEG)
C        TEPO       YES    NO       EXHAUST PORT OPENS (DEG)
C        TEPC       YES    NO       EXHAUST PORT CLOSES (DEG)
C        THIPO      YES    NO       INTAKE PORT OPENING TIME (DEG)
C        THEPO      YES    NO       EXHAUST PORT OPENING TIME (DEG)
C        IPA        YES    NO       INTAKE PORT OPEN AREA (CM**2)
C        EPA        YES    NO       EXHAUST PORT OPEN AREA (CM**2)
C
C
C     2  OPERATING PARAMETERS
C
C        PATM       YES    NO       ATMOSPHERIC PRESSURE (ATM)
C        TATM       YES    NO       ATMOSPHERIC TEMPERATURE (K)
C        PIM        YES    NO       INTAKE PRESSURE (ATM)
C        PEM        YES    NO       EXHAUST PRESSURE (ATM)
C        TFRESH     YES    NO       FRESH CHARGE TEMPERATURE (K)
C        TEGR       YES    NO       EGR TEMPERATURE (K)
C        EGR        YES    NO       EXHAUST GAS RECIRCULATION (%)
C        TSPARK     YES    NO       IGNITION TIMING (DEG)
C        RPM        YES    NO       ENGINE SPEED (RPM)
C
C
C     3  LEAKAGE AND CREVICE VOLUME SUB-MODEL CONSTANTS
C
C        AREALK     YES    NO       LEAK AREA PER APEX SEAL (CM**2)
C        CREVOL     YES    NO       CREVICE VOLUME PER APEX SEAL (CM**2)
```

```
C        TCREV      YES    NO       CREVICE VOLUME GAS TEMPERATURE (K)
C
C
C     4  HEAT TRANSFER CONSTANTS: NU = CONS'T * (REYNOLDS NO.)**EXP'NT
C
C        CONHT      YES    NO       CONS'T
C        EXPHT      YES    NO       EXP'NT
C        CON1       YES    NO       CONSTANT FOR NON-FIRE CHARACTERISTIC
C                   ---    --       VELOCITY
C        CON2       YES    NO       CONSTANT FOR COMBUSTION
C                   ---    --       CHARACTERISTIC VELOCITY
C        TROTOR     YES    NO       ROTOR SURFACE TEMPERATURE (K)
C        TSIDE      YES    NO       SIDE PLATE SURFACE TEMPERATURE (K)
C        THOUS      YES    NO       HOUSING SURFACE TEMPERATURE (K)
C
C
C
C     5  FUEL AND AIR SPECIFICATIONS
C
C        FUELTP     YES    NO       = 1  :   ISOOCTANE
C        ------     ---    --       = 2  :   PROPANE
C        ------     ---    --                (SEE SUBROUTINE FUELDT)
C        PHI        YES    NO       EQUIVALENCE RATIO
C        CX         NO     NO       NUMBER OF CARBON ATOMS IN THE
C        -----      --     --       FUEL (8.0 FOR C8H18)
C        DEL        NO     NO       MOLAR C:H RATIO OF THE FUEL
C        PSI        NO     NO       MOLAR N:O RATIO OF AIR
C        QLOWER     NO     NO       LOWER HEATING VALUE OF FUEL (MJ/KG)
C
C
C     6  ERROR TOLERANCES
C
C        AREROT     YES    NO       ERROR TOLERANCE FOR CALCULATING
C        ------     ---    --       ROOTS (SEE SUBROUTINE ODERT).
C        CIINTG     YES    NO       ERROR TOLERANCE FOR INTEGRATION
C        ------     ---    --       DURING INTAKE PROCESS (SEE
C        ------     ---    --       SUBROUTINE ODERT).
C        CCINTG     YES    NO       SAME, DURING COMPRESSION PROCESS
C        CBINTG     YES    NO       SAME, DURING COMBUSTION PROCESS
C        CEINTG     YES    NO       SAME, DURING EXHAUST PROCESS
C        REL        YES    NO       RELATIVE ERROR TOLERANCE FOR
C        ---        ---    --       CONTINUING INTEGRATION TO TOUT
C        ---        ---    --       (SEE MAIN PROGRAM).
C        MAXITS     YES    NO       MAXIMUM NUMBER OF ITERATIONS TO
C        ------     ---    --       COMPLETE CYCLE SIMULATION
C
C
C     7  INITIAL GUESSES AT THE START OF INTAKE PROCESS
C
C        PSTART      NO    NO       INITIAL PRESSURE IN CYLINDER (ATM)
C        TSTART      NO    NO       INITIAL TEMPERATURE IN CYLINDER (K)
C
C
C     8  TIME INCREMENTS
C
```

```
C        TPRINT     YES    NO       PRINTING INTERVAL DURING INTAKE,
C        -----      ---    --       COMPRESSION, AND EXHAUST (DEG)
C        TPRINX     YES    NO       PRINTING INTERVAL DURING COMBUSTION
C        -----      ---    --       AND EXPANSION(DEG)
C
C
C     9  OPERATING CASE
C
C        FIRE       YES    NO       = .TRUE.  FOR FIRING CASE
C        ----       ---    --       = .FALSE. FOR MOTORING CASE
C        SPBURN     YES    NO       = .TRUE.  FOR SPECIFIED BURN RATE
C        ----       ---    --       = .FALSE. NOT USED IN THIS PROGRAM
C
C
C    10  COMBUSTION MODEL INPUTS
C
C        XBZERO     YES    NO       INITIAL MASS FRACTION BURNED
C        ------     ---    --       (INITIALIZES COMBUSTION MODEL)
C        XBSTOP     YES    NO       MASS FRACTION BURNED AT END OF
C        ------     ---    --       COMBUSTION.
C        DTBRN      YES    NO       BURN DURATION FOR WIEBE
C        -----      ---    --       FUNCTION (DEG)
C        CONSPB     YES    NO       WIEBE FUNCTION CONSTANT
C        EXSPB      YES    NO       WIEBE FUNCTION EXPONENT
C
C
C
C     REMARKS
C        NONE
C
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C
C     1  WORKING SUBROUTINES
C
C        INTAKE     CMPRES     CMBSTN     EXAUST   CREVIC
C        GINT1      GINT2      GCMP       GEXH
C
C
C     2  SPECIAL UTILITY SUBROUTINES
C
C        IPACD      EPACD      CSAVDV     TABLE    BUILD
C
C
C     3  GENERAL UTILITY SUBROUTINES
C
C        UPROP      HPROD      CLDPRD     HELPHT
C        UTRANS     BTRANS     MFLRT      THERMO     PTCHEM
C        FUELDT     WRITE      ERRCHK     INTRP      ITRATE
C        ODERT      DERT1      ROOT       STEP1
C
C
C     METHOD
C        SEE REPORT
C
```

```
C     WRITTEN BY  S. G. POULOS, S. H. MANSOURI, AND T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      LOGICAL FIRE, SPBURN
      INTEGER FUELTP
      REAL*8 DT, DY(25), TOUT, RELERR, ABSERR, WORK, REROOT, AEROOT
      REAL MW, MWIM, MSTART, MASS, MAXERR, IPA
      DIMENSION Y(25), YP(25), WORK(625), IWORK(5)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, VFLANK
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/MANFP/ PIM, TIM, EGR, PEM, MSTART
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/PORTS/ IPA, EPA
      COMMON/IMTHP/ HIM, MWIM, GIM, RHOIM
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
      COMMON/FIXX/ INFLAG
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/HTRAN/THTRAN
      COMMON/UBHEAT/ UHTRCO, BHTRCO
      COMMON/YYYY1/ VIP, VEP
      COMMON/ITRLIM/ MAXTRY, MAXERR
      COMMON/RHMAS/ RHO, MASS, VOLUME, H, GAMMA, GAMAB
      COMMON/SPECB/ DTBRN, CONSPB, EXSPB
      COMMON/XSTOP/ XBSTOP
C
C
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
      COMMON/CREVIN/AREALK,CREVOL,TCREV,X1LDIN,X1LGIN
      COMMON/CREVQ/CRMASD,CRMASG,X1LDC,X1LAGC,ZMAST,ZMASS,ZLEAKD,ZLEAKG,
     &             ZCRCOD,ZCRCOG
      COMMON/TABLES/ PRES(1080), X1LD(1080), X1LG(1080)
      COMMON/HEATXG/AROTOR,ASIDE,AHOUS,ROTVEL,DCHAR
      COMMON/HTTXIN/CONHT,EXPHT,CON1,CON2
C
      NAMELIST/INPUT/ FIRE, SPBURN, FUELTP, PHI, ECCEN, ROTRAD, DEPTH,
     &         VFLANK, RPM, TIPO, TIPC, TEPO, TEPC, TSPARK, THIPO,
     & THEPO, IPA, EPA, XBZERO, XBSTOP, DTBRN, CONSPB, EXSPB,
     & PATM, TATM, PIM, TFRESH, TEGR, EGR, PEM, TROTOR, TSIDE, THOUS,
     & CONHT, EXPHT, TPRINT, TPRINX, AREROT, CIINTG,
     & CCINTG, CBINTG, CEINTG, MXTRY, REL, MAXITS, MAXERR,
     & MAXTRY, AREALK, CREVOL, TCREV,  CON1, CON2
C
      EXTERNAL INTAKE, CMPRES, CMBSTN, EXAUST, GINT1, GINT2, GCMP,
     &         GCMB, GEXH
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C        STANDARD DATA SET -- OVERRIDE BY USING NAMELIST INPUT
C
C--------------------
```

```
      FIRE   = .TRUE.
      SPBURN = .TRUE.
      FUELTP =  1
      PHI    =  1.00
C--------------------
      ECCEN  =  1.50
      ROTRAD = 10.5
      DEPTH  =  7.00
      VFLANK = 25.00
      RPM    = 3000.
C--------------------
      TIPO   = -500.0
      TIPC   = -220.0
      TEPO   = 190.0
      TEPC   = 580.0
      TSPARK = 0.00
      THIPO  = 67.0
      THEPO  = 30.0
      IPA    = 12.2
      EPA    = 6.5
C--------------------
      XBZERO = 0.005
      XBSTOP = 0.995
      DTBRN  = 40.
      CONSPB = 5.
      EXSPB  = 1.
C--------------------
      PATM   = 1.0
      TATM   = 300.0
      PIM    = 0.9959
      TFRESH = 300.0
      TEGR   = 300.0
      EGR    = 0.0
      PEM    = 0.9876
C--------------------
      TROTOR = 462.
      TSIDE  = 462.
      THOUS  = 462.
      CONHT  = 0.377
      EXPHT  = 0.8
      CON1   = 0.75
      CON2   = 0.324
C--------------------
      TPRINT = 10.0
      TPRINX = 1.0
C--------------------
      AREROT = .0002
      CIINTG = 0.0001
      CCINTG = 0.0001
      CBINTG = 0.00005
      CEINTG = 0.0001
      MXTRY  = 1
      REL    = .0002
      MAXITS = 2
      MAXERR = 0.03
```

```
      MAXTRY = 5
C--------------------
      AREALK = .01
      CREVOL = 2.00
      TCREV  = 300
C--------------------
C
C        READ NAMELIST INPUT
C
      READ (8,INPUT)
C
      ESPDI  = 1./(6. * RPM)
C
C
C         TO ALLOW A NORMALIZED OUTPUT INTO DUMMY1 DATA FILE GIVE
C         ZMAST (MASS AT TIPC) A VALUE OF UNITY UNTIL INTAKE PORT CLOSES
C
      ZMAST = 1.
C
C         CALCULATE COMPRESSION RATIO
C
      PI=3.1415926539
      ROOT3 = SQRT(3.0)
      DVOLUM = 3. * ROOT3 * ECCEN * ROTRAD * DEPTH
      ALEAN  = ASIN( 3.*ECCEN/ROTRAD)
      FH     = 1.5 * ROOT3 * ECCEN * ROTRAD
      FC     = 2. * ECCEN * ROTRAD * COS( ALEAN ) +
     &         ( 2/9.*ROTRAD*ROTRAD + 4.*ECCEN*ECCEN ) * ALEAN +
     &         PI/3. * ECCEN*ECCEN
      CMRTIO = (( FH + FC )*DEPTH + VFLANK)/(( FC- FH )*DEPTH + VFLANK)
C
C        CALCULATE ALL FUEL-RELATED PARAMETERS
C
C
C
      IF ((.NOT. FIRE) .OR. (PHI .LE. 0.0)) PHI = 0.00001
      CALL FUELDT
C
C
C        FIND THERMODYNAMIC STATE OF INTAKE CHARGE
C
      CALL THERMO (TIPO, TFRESH, PIM, 0.0, HFRESH, XXA, XXB, XXC,
     &             XXD, XXE, XXF, XXG, XXH, XXI, XXJ, XXK)
      CALL THERMO (TIPO, TEGR, PIM, 1.0, HEGR, XXA, XXB, XXC,
     &             XXD, XXE, XXF, XXG, XXH, XXI, XXJ, XXK)
      HIM = (1. - EGR/100.)*HFRESH + (EGR/100.)*HEGR
      RESFIM = EGR/100.
      TGUESS = (1. - EGR/100.)*TFRESH + (EGR/100.)*TEGR
      CALL ITRATE (TIPO, TGUESS, PIM, RESFIM, HIM, CSUBPI, CSUBTI,
     &             RHOIM, DRODTI, DRODPI, GIM, MWIM, XXA, XXB, XXC, XXD)
      TIM = TGUESS
C
C        MAKE INITIAL GUESSES FOR FIRST CYCLE ITERATION
C
      PSTART = 1.1
```

```
      TSTART = 330.
      IF (FIRE) PSTART = 1.015
      IF (FIRE) TSTART = 1000.
C
C*********************************************************************
C
C        START OF CURRENT CYCLE ITERATION
C
C*********************************************************************
C
      DO 470 ITERAS = 1, MAXITS
C
      WRITE (7,449) ITERAS, MAXITS
C
C        CALCULATE MASS IN CYLINDER
C
      RESFRK = 0.0
      IF (FIRE) RESFRK = 1.0
      IF (.NOT. FIRE) EGR = 0.0
      CALL THERMO (TIPO, TSTART, PSTART, RESFRK, ENTHLP, CSUBP, CSUBT,
     &       RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
      CALL CSAVDV (TIPO, VOLUME, DVDT)
      MSTART = RHO * VOLUME
C
C        PREPARE ALL OUTPUT FILES FOR OUTPUT FROM THIS ITERATION
C
CJ  5 REWIND  6
CJ    REWIND  9
CJ    REWIND 10
CJ  5 REWIND 11
CJ    REWIND 12
CJ    REWIND 13
CJ    REWIND 14
CJ    REWIND 15
CJ    REWIND 16
C
C        WRITE MAIN HEADINGS AND ECHO INPUT PARAMETERS
C
    5 WRITE  (6,3333)
      WRITE  (6,3333)
      WRITE  (6,2901)
      WRITE  (6,3333)
      WRITE  (6,3333)
      WRITE  (6,77)
      WRITE  (6,3333)
      WRITE  (6,2902)
      IF (FIRE) WRITE (6,2903)
      IF (.NOT. FIRE) WRITE (6,2904)
      IF (SPBURN .AND. FIRE) WRITE (6,2905)
      IF (SPBURN .AND. FIRE) WRITE (6,2906) DTBRN, CONSPB, EXSPB
      IF (.NOT. SPBURN .AND. FIRE) WRITE (6,2907)
      WRITE  (6,3333)
      WRITE  (6,2908)
      IF (FIRE .AND. FUELTP .EQ. 1) WRITE (6,2909)
      IF (FIRE .AND. FUELTP .EQ. 2) WRITE (6,2910)
```

```
      IF (FIRE) WRITE (6,2911) PHI
      IF (FIRE .AND. ((PHI .GT. 1.3) .OR. (PHI .LT. 0.7))) WRITE (6,999)
      IF (FIRE .AND. ((PHI .GT. 1.3) .OR. (PHI .LT. 0.7))) WRITE (7,999)
      IF (FIRE) WRITE  (6,2912) TSPARK
      WRITE (6,2913) RPM
      WRITE  (6,3333)
      WRITE  (6,2914)
      WRITE  (6,2915) PIM, PEM, TFRESH, EGR, TEGR, TIM, PATM, TATM
      WRITE  (6,3333)
      WRITE  (6,2916)
      WRITE  (6,2917) CONHT, EXPHT, TROTOR, TSIDE, THOUS
      WRITE  (6,3333)
      WRITE  (6,2918)
      WRITE  (6,2919) ECCEN, ROTRAD, DEPTH, CMRTIO, DVOLUM,
     &                VFLANK, TIPO, TIPC, TEPO, TEPC
      WRITE  (6,3333)
      WRITE  (6,2940)
      WRITE  (6,2941) AREALK, CREVOL, TCREV
      WRITE  (6,3333)
      WRITE  (6,2920)
      WRITE  (6,2921) MAXITS, ITERAS, TPRINT, TPRINX, XBZERO, XESTOP,
     &                XBSTOP, CIINTG, CCINTG, CBINTG, CEINTG, AREROT,
     &                REL, ERMAX, MAXERR, MAXTRY
      WRITE  (6,3333)
      WRITE  (6,3333)
      WRITE  (6,2225)
      WRITE (10,2225)
      WRITE (13,2225)
C
      WRITE  (6,4595)
      WRITE  (6,4596)
      WRITE  (6,3333)
      WRITE (10,8111)
      WRITE (10,9900)
      WRITE (10,3333)
      WRITE (13,7111)
      WRITE (13,4592)
      WRITE (13,3333)
C
C        INITIALIZE PARAMETERS FOR CALL TO SUBROUTINE ODERT
C
      Y(1)  = 0.0
      Y(2)  = 0.0
      Y(3)  = 0.0
      Y(4)  = 0.0
      Y(5)  = 1.0
      IF (FIRE) Y(5) = 0.0
CJ    Y(6) REMOVED
CJ    Y(7) REMOVED
      Y(8)  = 0.0
      Y(9)  = 0.0
      Y(10) = 0.0
      Y(11) = TSTART
      Y(12) = PSTART
      Y(13) = 0.0
```

```
      Y(14) = 0.0
      Y(15) = 0.0
      Y(16) = 0.0
      Y(17) = 0.0
      Y(18) = 0.0
      Y(19) = 0.0
      Y(20) = 0.0
      Y(21) = X1LDIN
      Y(22) = X1LGIN
      Y(23) = MSTART
      Y(24) = 0.0
      Y(25) = 0.0
C
      HEATI = 0.0
      WORKI = 0.0
      VIP = 0.0
C
      DO 10 I = 1, 25
           DY(I) = Y(I)
   10 CONTINUE
      WRITE (6,1277) TIPO, DY(12), DY(11), DY(1), DY(2), VIP,
     &               DY(5), DY(16)
      AEROOT = AREROT
      REROOT = AREROT
C
C#####################################################################
C
C        START OF INTAKE PROCESS (TIPO - TIPC)
C
C#####################################################################
C
   20 I = 0
      NEQN = 25
      IFLAG = 1
      T = TIPO
      TEND = -270.
      DT = T
C
C        CHECK WHICH WAY INTAKE FLOWS AT CYCLE START
C
   30 I = I + 1
      IFLAG = 1
      IF (DY(12) .GT. PIM) GO TO 90
C
C        INTAKE FLOW INTO CHAMBER OF NEW CHARGE
C        ( FRESH AIR/FUEL MIXTURE + EGR ); SET INFLAG = 1
C
   40 INFLAG = 1
C
C
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 70
C
C        NCALL:  NO. OF TIMES INTEGRATING SUBROUTINE IS CALLED
C
```

```
      DO 60 NC = 1, NCALL
           TOUT = INT (T + 1.)
   50      ABSERR = CIINTG
           RELERR = CIINTG
C
           CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &                 IFLAG, WORK, IWORK, GINT1, REROOT, AEROOT)
           CALL HELPHT (DT, DY, 1)
C
           T = DT
C
C          SUBROUTINE BUILD CONSTRUCTS THE PRESSURE HISTORY OF
C          THE CHAMBER AND STORES THE CREVICE GAS COMPOSITIONS
C          WHEN AVAILABLE.
C
           CALL BUILD (DT,DY)
C
       TWRITE = T/TPRINT
C
C           IFLAG IS THE RETURN CODE FROM ODERT. IFLAG NOT EQUAL
C           TO 2 OR 8 IS ABNORMAL AND SHOULD BE CHECKED (REFER
C           TO SUBROUTINE ODERT).
C
       IF ( IFLAG .NE. 2 ) GO TO 55
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 56
C
   55      WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
           WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                     VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
   56 CONTINUE
           IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
           IF (IFLAG .EQ. 8) GO TO 90
C
C             I.E. REVERSE FLOW ACROSS INTAKE PORT.
C
           IF (IFLAG .NE. 2) GO TO 50
   60 CONTINUE
C
C        NO ROOT FOR GINT1; COMPLETE INTAKE PROCESS.
C
   70 TOUT = TEND
   80 ABSERR = CIINTG
      RELERR = CIINTG
C
      CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &            IFLAG, WORK, IWORK, GINT1, REROOT, AEROOT)
      CALL HELPHT (DT, DY, 1)
C
           T = DT
           CALL BUILD (DT,DY)
```

```
C
      TWRITE = T/TPRINT
      IF ( IFLAG .NE. 2 ) GO TO 85
      IF ( TWRITE .NE. INT(TWRITE) ) GO TO 86
C
   85 WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
      WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
   86 CONTINUE
      IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
      IF (IFLAG .EQ. 8) GO TO 90
      IF (IFLAG .NE. 2) GO TO 80
C
C        ROOT FOUND FOR GINT1; FLOW ACROSS INTAKE PORT
C        REVERSES AND FLOWS INTO INTAKE MANIFOLD.  FIND
C        ROOT WHEN FLOW ONCE AGAIN REVERSES DIRECTION.
C
   90 INFLAG = 0
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 120
      DO 110 NC = 1, NCALL
           TOUT = INT(T + 1.)
  100      ABSERR = CIINTG
           RELERR = CIINTG
C
           CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &                 IFLAG, WORK, IWORK, GINT1, REROOT, AEROOT)
           CALL HELPHT (DT, DY, 1)
C
           T = DT
           CALL BUILD (DT,DY)
      TWRITE = T/TPRINT
      IF ( IFLAG .NE. 2 ) GO TO 105
      IF ( TWRITE .NE. INT(TWRITE) ) GO TO 106
  105      WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
           WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                     VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
  106   CONTINUE
           IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
           IF (IFLAG .EQ. 8) GO TO 140
           IF (IFLAG .NE. 2) GO TO 100
  110 CONTINUE
  120 TOUT = TEND
  130 ABSERR = CIINTG
      RELERR = CIINTG
```

```
C
      CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &            IFLAG, WORK, IWORK, GINT1, REROOT, AEROOT)
      CALL HELPHT (DT, DY, 1)
C
           T = DT
           CALL BUILD (DT,DY)
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 135
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 136
  135 WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
      WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
  136 IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
      IF (IFLAG .EQ. 8) GO TO 140
      IF (IFLAG .NE. 2) GO TO 130
C
C        ROOT FOUND FOR GINT1; FLOW ACROSS INTAKE PORT HAS
C        REVERSED DIRECTION.  FIND ROOT WHEN ALL MASS THAT HAS
C        FLOWN INTO INTAKE MANIFOLD FLOWS BACK INTO CYLINDER.
C
  140 INFLAG = 0
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 170
      DO 160 NC = 1, NCALL
           TOUT = INT( T + 1.)
  150      ABSERR = CIINTG
           RELERR = CIINTG
C
           CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &                 IFLAG, WORK, IWORK, GINT2, REROOT, AEROOT)
           CALL HELPHT (DT, DY, 1)
C
           CALL BUILD (DT, DY)
           T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 155
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 156
  155      WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
           WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                     VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
  156      IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
           IF (IFLAG .EQ. 8) GO TO 40
           IF (IFLAG .NE. 2) GO TO 150
  160 CONTINUE
```

```
  170 TOUT = TEND
  180 ABSERR = CIINTG
      RELERR = CIINTG
C
      CALL ODERT (INTAKE, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &            IFLAG, WORK, IWORK, GINT2, REROOT, AEROOT)
      CALL HELPHT (DT, DY, 1)
C
      CALL BUILD (DT, DY)
      T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 185
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 186
  185 WRITE  (7,881)  DT, DY(12), INFLAG, IFLAG
      WRITE  (6,1210) DT, DY(12), DY(11), DY(1), DY(2), VIP,
     &                VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
  186 IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 190
      IF (IFLAG .EQ. 8) GO TO 40
      IF (IFLAG .NE. 2) GO TO 180
C
C#####################################################################
C
C        END OF INTAKE PROCESS
C
C#####################################################################
C
  190 IF (I .EQ. 2) GO TO 200
      HEATI = DY(8) + DY(9) + DY(10)

      WORKI = DY(16)
      TEND = TIPC
      GO TO 30
C
C        CALCULATE VOLUMETRIC EFFICIENCY (VOLEFI)
C
  200 VOLEFI = 100. * DY(1)/( DVOLUM * RHOIM * (1. + PHI*FASTO) )
      VOLEFA = VOLEFI * (PIM/PATM) * (TATM/TIM)
      WRITE (7,1281) VOLEFI
      WRITE  (6,1210) TIPC, DY(12), DY(11), DY(1), DY(2), VIP,
     &                    VEP, DY(5), THTRAN, DY(16), INFLAG, IFLAG
      WRITE (6,3333)
      WRITE (6,2225)
      WRITE (6,1110)
      WRITE (6,4597)
      WRITE (6,3333)
      WRITE (6,1211) TIPC, DY(12), DY(11), THTRAN,
     &               DY(16), IFLAG
C
C        CALCULATE TOTAL MASS OF FUEL INDUCTED IN THIS CYCLE (FMIN)
C
```

```
      ZMAST = DY(23)
      FMIN = DY(1) * (1. - EGR/100.) * PHI * FASTO/(PHI*FASTO + 1.)
      IF (.NOT. FIRE) FMIN = 0.0
C
C        CALCULATE RESIDUAL FRACTION AT TIPC
C
      RESIDL = 1. - DY(5)
C
C#####################################################################
C
C        START OF COMPRESSION PROCESS (FIRING CASE) (TIPC - TSPARK)
C        START OF COMPRESSION AND EXPANSION PROCESSES
C        (MOTORING CASE) (TIPC - TEPO)
C
C#####################################################################
C
      TID = TSPARK
      TBD = TSPARK
      NEQN = 25
      IFLAG = 1
      T = TIPC
      TEND = TEPO
      IF (FIRE) TEND = TSPARK
      DT = T
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 230
      DO 220 NC = 1, NCALL
           TOUT = INT( T + 1.)
  210      ABSERR = CCINTG
           RELERR = CCINTG
C
           CALL ODERT (CMPRES, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &                 IFLAG, WORK, IWORK, GCMP, REROOT, AEROOT)
           CALL HELPHT (DT, DY, 2)
C
           CALL BUILD (DT, DY)
           T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 215
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 216
  215      WRITE  (7,882)  DT, DY(12), IFLAG
           WRITE  (6,1211) DT, DY(12), DY(11), THTRAN,
     &                     DY(16), IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
  216      IF (ABS( T - TEND ) .LE. REL) GO TO 250
           IF (IFLAG .EQ. 8) GO TO 220
           IF (IFLAG .NE. 2) GO TO 210
  220 CONTINUE
  230 TOUT = TEND
  240 ABSERR = CCINTG
      RELERR = CCINTG
```

```
C
      CALL ODERT (CMPRES, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &            IFLAG, WORK, IWORK, GCMP, REROOT, AEROOT)
      CALL HELPHT (DT, DY, 2)
C
      CALL BUILD (DT, DY)
      T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 245
  245 WRITE  (7,882)  DT, DY(12), IFLAG
      WRITE  (6,1211) DT, DY(12), DY(11), THTRAN,
     &                DY(16), IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
  246 IF (ABS( T - TEND ) .LE. REL) GO TO 250
      IF (IFLAG .NE. 2) GO TO 240
C
C#####################################################################
C
C        IF FIRING CASE, GO TO START OF COMBUSTION
C        IF MOTORING CASE, BEGIN EXHAUST PROCESS (TEPO - TIPO)
C
C#####################################################################
C
       XFRESH = Y(5)
  250 WRITE (6,3333)
      IF (FIRE) GO TO 330
      WRITE (6,2225)
      WRITE (6,1111)
      WRITE (6,4599)
      WRITE (6,3333)
      VEP = 0.0
      WRITE (6,1213) TEPO, DY(12), DY(11), DY(2), VEP,
     &               THTRAN, DY(16), IFLAG
C
      I = 0
      NEQN = 25
      IFLAG = 1
      T = TEPO
      TEND = 270.
      DT = T
  260 I = I + 1
      IFLAG = 1
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 290
      DO 280 NC = 1, NCALL
           TOUT = INT( T + 1.)
  270      ABSERR = CEINTG
           RELERR = CEINTG
C
           CALL ODERT (EXAUST, NEQN, DY, DT, TOUT, RELERR, ABSERR,
     &                 IFLAG, WORK, IWORK, GEXH, REROOT, AEROOT)
```

```
          CALL HELPHT (DT, DY, 4)
C
          CALL BUILD (DT, DY)
          T = DT
      TWRITE = T/TPRINT
      IF ( IFLAG .NE. 2 ) GO TO 275
      IF ( TWRITE .NE. INT(TWRITE) ) GO TO 276
  275     WRITE  (7,882)  DT, DY(12), IFLAG
          WRITE  (6,1213) DT, DY(12), DY(11), DY(2), VEP,
     &                    THTRAN, DY(16), IFLAG
          WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                    ZCRCOD,ZCRCOG
          WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                    QFRRO, QFRSI, QFRHO
C
  276     IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 310
          IF (IFLAG .NE. 2) GO TO 270
  280 CONTINUE
  290 TOUT = TEND
  300 ABSERR = CEINTG
      RELERR = CEINTG
C
      CALL ODERT (EXAUST,NEQN,DY,DT,TOUT,RELERR,ABSERR,IFLAG,
     &           WORK,IWORK,GEXH,REROOT,AEROOT)
      CALL HELPHT (DT, DY, 4)
C
      CALL BUILD (DT, DY)
      T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 305
       IF ( I .EQ. 2  .AND.  ABS(T/TEND - 1.) .LE. REL ) GO TO 305
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 306
  305 WRITE  (7,882)  DT, DY(12), IFLAG
      WRITE  (6,1213) DT, DY(12), DY(11), DY(2), VEP,
     &                THTRAN, DY(16), IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
  306 IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 310
      IF (IFLAG .NE. 2) GO TO 300
  310 IF (I .EQ. 2) GO TO 320
C
      HEATCE = DY(8) + DY(9) + DY(10) - HEATI
      WORKCE = DY(16) - WORKI
      TEND = TIPO + 1080.
      GO TO 260
C
  320 HEATE = DY(8) + DY(9) + DY(10) - HEATCE - HEATI
      WORKE = DY(16) - WORKCE - WORKI
      WRITE (6,3333)
C
C#################################################################
C
```

```
C        END OF EXHAUST PROCESS (MOTORING CASE)
C
C#######################################################################
C
      GO TO 455
  330 CONTINUE
C
      HEATC = DY(8) + DY(9) + DY(10) - HEATI
      WORKC = DY(16) - WORKI
      WRITE (6,2225)
      WRITE (6,1112)
      WRITE (6,4598)
      WRITE (6,3333)
C
C        REINITIALIZE 'ODERT' FOR START OF COMBUSTION
C
C        CALCULATE PRESSURE RISE DUE TO INITIAL HEAT RELEASE
C
      MASS = DY(23)
      PZERO = DY(12) + 9.8692326 * MASS * XBZERO * QLOWER *
     &        (GAMMA - 1.)/VOLUME
C
C        CALCULATE INITIAL BURNED ZONE TEMPERATURE
C
      HB = H
      TGUESB = 2400.
      CALL ITRATE (TSPARK, TGUESB, PZERO, 1.0, HB, XXA, XXB,
     &             RHOB, XXC, XXD, XXE, XXF, XXG, XXH, XXI, XXJ)
C
C        CALCULATE INITIAL UNBURNED ZONE TEMPERATURE
C
      RHOUEX = (1. - XBZERO)/( VOLUME/MASS - XBZERO/RHOB )
      TGUESU = DY(11)
      RESFRK = 1. - DY(5)
      DO 335 I = 1, 30
         CALL THERMO (TSPARK, TGUESU, PZERO, RESFRK, XXA, XXB, XXC,
     &                RHOUG, DRDTUG, XXD, XXE, XXF, XXG, XXH, XXI, XXJ)
         IF ( ABS( (RHOUG - RHOUEX)/RHOUEX ) .LE. 0.00001 ) GO TO 337
         TGUESU = TGUESU + (RHOUEX - RHOUG)/DRDTUG
  335 CONTINUE
  337 CALL THERMO (TSPARK, TGUESU, PZERO, RESFRK, HU, XXB, XXC,
     &             RHOU, DRDTU, XXD, XXE, XXF, XXG, XXH, XXI, XXJ)
C
      DY(4)  = XBZERO
      DY(12) = PZERO
      DY(13) = TGUESU
      DY(14) = VOLUME - ( DY(4)*MASS/RHOB )
      DY(15) = TGUESB
C
      WRITE  (6,1212) TSPARK, DY(12), DY(15), DY(13),
     &                DY(4), THTRAN, DY(16), IFLAG
      DT = TSPARK
      CALL HELPHT (DT, DY, 3)
      WRITE (13,4210) TSPARK, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
```

```
C
C#####################################################################
C
C        START OF COMBUSTION PROCESS (TSPARK - TEPO)
C
C#####################################################################
C
      IDCNT = 0
      IBCNT = 0
      NEQN = 25
      IFLAG = 1
      T = TSPARK
      TEND = TEPO
      DT = T
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 360
      DO 350 NC = 1, NCALL
           TOUT = INT( T + 1.)
  340      ABSERR = CBINTG
           RELERR = CBINTG
           TOLDXB = T
           XBOLD = DY(4)
C
           CALL ODERT (CMBSTN,NEQN,DY,DT,TOUT,RELERR,ABSERR,IFLAG,
     &                 WORK,IWORK,GCMB,REROOT,AEROOT)
           CALL HELPHT (DT, DY, 3)
           IF (DY(4) .LE. 1.0) GO TO 344
           IF (DY(4) .GT. 1.0) XBSTOP = XBSTOP - 0.002
           GO TO 5
C
  344      CALL BUILD (DT, DY)
           T = DT
C
           IF ((IDCNT .GT. 0) .OR. (DY(4) .LT. 0.1)) GO TO 345
           TID = TOLDXB + (T - TOLDXB)*(0.1 - XBOLD)/(DY(4) - XBOLD)
           IDCNT = IDCNT + 1
  345      IF ((IBCNT .GT. 0) .OR. (DY(4) .LT. 0.9)) GO TO 347
           TBD = TOLDXB + (T - TOLDXB)*(0.9 - XBOLD)/(DY(4) - XBOLD)
           IBCNT = IBCNT + 1
  347      TWRITE = T/TPRINX
           IF (IFLAG .NE. 2 ) GO TO 348
           IF (TWRITE .NE. INT(TWRITE) ) GO TO 349
C
  348      WRITE  (7,883)  DT, DY(12), DY(4), IFLAG
           WRITE  (6,1212) DT, DY(12), DY(15), DY(13),
     &                     DY(4), THTRAN, DY(16), IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
  349      IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 380
           IF (IFLAG .NE. 2) GO TO 340
  350 CONTINUE
  360 TOUT = TEND
```

```
  370 ABSERR = CBINTG
      RELERR = CBINTG
      TOLDXB = T
      XBOLD = DY(4)
C
      CALL ODERT (CMBSTN,NEQN,DY,DT,TOUT,RELERR,ABSERR,IFLAG,
     &            WORK,IWORK,GCMB,REROOT,AEROOT)
      CALL HELPHT (DT, DY, 3)
      IF (DY(4) .LE. 1.0) GO TO 374
      IF (DY(4) .GT. 1.0) XBSTOP = XBSTOP - 0.02
      GO TO 5
C
  374 CALL BUILD (DT, DY)
      T = DT
C
      IF ((IDCNT .GT. 0) .OR. (DY(4) .LT. 0.1)) GO TO 375
      TID = TOLDXB + (T - TOLDXB)*(0.1 - XBOLD)/(DY(4) - XBOLD)
      IDCNT = IDCNT + 1
  375 IF ((IBCNT .GT. 0) .OR. (DY(4) .LT. 0.9)) GO TO 377
      TBD = TOLDXB + (T - TOLDXB)*(0.9 - XBOLD)/(DY(4) - XBOLD)
      IBCNT = IBCNT + 1
C
  377 WRITE  (7,883)  DT, DY(12), DY(4), IFLAG
      WRITE  (6,1212) DT, DY(12), DY(15), DY(13),
     &                DY(4), THTRAN, DY(16), IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                QFRRO, QFRSI, QFRHO
C
      IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 380
      IF (IFLAG .NE. 2) GO TO 370
C
C#######################################################################
C
C        START OF EXHAUST PROCESS (FIRING CASE) (TEPO - TIPO)
C
C#######################################################################
C
  380 WRITE (6,3333)
      WRITE (6,2225)
      WRITE (6,1111)
      WRITE (6,4599)
      WRITE (6,3333)
      VEP = 0.0
      WRITE (6,1213) TEPO, DY(12), DY(15), DY(2), VEP,
     &               THTRAN, DY(16), IFLAG
C
C        REINITIALIZE 'ODERT' FOR START OF EXHAUST
C
      DY(11) = DY(15)
      DY(5) = RESIDL * (1.-DY(4))
C
      I = 0
      NEQN = 25
```

```
      IFLAG = 1
      T = TEPO
      TEND = 270.
      DT = T
  390 I = I + 1
      NCALL = IFIX( ABS(TEND - T) )
      IF (NCALL .LE. 0) GO TO 420
      DO 410 NC = 1, NCALL
           TOUT = INT( T + 1.)
  400      ABSERR = CEINTG
           RELERR = CEINTG
C
           CALL ODERT (EXAUST,NEQN,DY,DT,TOUT,RELERR,ABSERR,IFLAG,
     &                 WORK,IWORK,GEXH,REROOT,AEROOT)
           CALL HELPHT (DT, DY, 4)
C
           CALL BUILD (DT, DY)
           T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 405
       IF ( I .EQ. 2   .AND.  ABS(T/TEND - 1.) .LE. REL )
     &   GO TO 435
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 406
  405      WRITE  (7,882)  DT, DY(12), IFLAG
           WRITE  (6,1213) DT, DY(12), DY(11), DY(2), VEP,
     &                     THTRAN, DY(16), IFLAG
           WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                     ZCRCOD,ZCRCOG
           WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
     &                     QFRRO, QFRSI, QFRHO
C
  406      IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 440
           IF (IFLAG .NE. 2) GO TO 400
  410 CONTINUE
  420 TOUT = TEND
  430 ABSERR = CEINTG
      RELERR = CEINTG
C
      CALL ODERT (EXAUST,NEQN,DY,DT,TOUT,RELERR,ABSERR,IFLAG,
     &            WORK,IWORK,GEXH,REROOT,AEROOT)
      CALL HELPHT (DT, DY, 4)
C
      CALL BUILD (DT, DY)
      T = DT
       TWRITE = T/TPRINT
       IF ( IFLAG .NE. 2 ) GO TO 435
       IF ( I .EQ. 2  .AND.  ABS(T/TEND - 1.) .LE. REL ) GO TO 435
       IF ( TWRITE .NE. INT(TWRITE) ) GO TO 436
C
  435 WRITE  (7,882)  DT, DY(12), IFLAG
      WRITE  (6,1213) DT, DY(12), DY(11), DY(2), VEP,
     &                THTRAN, DY(16), IFLAG
      WRITE (10,9210) DT,DY(23),CRMASD,CRMASG,DY(18),DY(19),
     &                ZCRCOD,ZCRCOG
      WRITE (13,4210) DT, VELHTX, HTPARO, HTPASI, HTPAHO,
```

```
     &                QFRRO, QFRSI, QFRHO
C
  436 IF (ABS(T/TEND - 1.0) .LE. REL) GO TO 440
      IF (IFLAG .NE. 2) GO TO 430
  440 IF (I .EQ. 2) GO TO 450
C
      HEATCE = DY(8) + DY(9) + DY(10) - HEATI
      WORKCE = DY(16) - WORKI
      TEND = TIPO + 1080.
      GO TO 390
C
C#######################################################################
C
C        END OF EXHAUST PROCESS (FIRING CASE)
C
C#######################################################################
C
  450 HEATE = DY(8) + DY(9) + DY(10) - HEATCE - HEATI
      WORKE = DY(16) - WORKCE - WORKI
      WRITE (6,3333)
C
C        CONVERGENCE CHECK
C
  455 Y(12) = DY(12)
      Y(11) = DY(11)
      Y(1) = DY(1)
      Y(2) = DY(2)
      IF (ITERAS .EQ. 1) GO TO 460
      IF (ABS( (Y(12) - PSTART)/PSTART ) .GT. 0.02) GO TO 460
      IF (ABS( (Y(11) - TSTART)/TSTART ) .GT. 0.02) GO TO 460
      IF (ABS( (Y(1) - Y(2))/MSTART )    .GT. 0.02) GO TO 460
      GO TO 480
  460 PSTART = DY(12)
      TSTART = DY(11)
C
C       INITIALIZE THE CREVICE GAS COMPOSITIONS
C
      X1LDIN = DY(22)
      X1LGIN = DY(21)
CN
C
CN470 CONTINUE
C
C***********************************************************************
C
C        END OF CURRENT CYCLE ITERATION
C
C***********************************************************************
C
C        CALCULATION RESULTS FOR THIS CYCLE
C
  480 THREFN = 0.0
      THREFG = 0.0
      HEATX  = 0.0
      IF (.NOT. FIRE) GO TO 490
```

```
      THREFN = 100. * DY(16)/(FMIN * QLOWER)
      THREFG = 100. * WORKCE/(FMIN * QLOWER)
      HEATX  = 100. * (DY(8) + DY(9) + DY(10))/(FMIN * QLOWER)
  490 ZPMEP  = 1.0E+06 * (WORKI + WORKE)/DVOLUM
      ZIMEP  = 1.0E+06 * WORKCE/DVOLUM
      ZISFC  = 3600. * FMIN/WORKCE
      RESFRK = 0.0
      IF (FIRE) RESFRK = 1.0
      AVREXH = DY(17)/DY(2)
      TGUESS = 500.
      IF (FIRE) TGUESS = 1300.
      CALL ITRATE (T, TGUESS, PEM, RESFRK, AVREXH, XXA, XXB, XXC,
     &             XXD, XXE, XXF, XXG, XXH, XXI, XXJ, XXK)
      AVREXT = TGUESS
      DTHIGD = TID - TSPARK
      DTIGD  = DTHIGD * ESPDI * 1.0E+3
      DTHBRN = TBD - TID
      DTBURN = DTHBRN * ESPDI * 1.0E+3
C
C        ENERGY BALANCE
C
      TOHIN = 1.0E-10 * HIM * DY(1)
      TOHEX = 1.0E-10 * DY(17)
      TOHEAT = HEATI + HEATCE + HEATE
      TOWORK = WORKI + WORKCE + WORKE
C
      DY(20) = DY(20) * 1.E-10
      DY(3) = DY(3) * 1.E-10
C
      DECYCL = TOHEX + TOHEAT + TOWORK - TOHIN + DY(20) + DY(3)
      DEONHI = 100.0 * DECYCL/TOHIN
      DEONQ = 0.0
      IF (FIRE) DEONQ = 100.0 * DECYCL/(FMIN * QLOWER)
      WRITE (6,2225)
      WRITE (6,5910)
      WRITE (6,3333)
      WRITE (6,5920) VOLEFI, VOLEFA, ZPMEP, ZIMEP, ZISFC, THREFG,
     &               THREFN, HEATX
      WRITE (6,5921) DTHIGD, DTIGD, DTHBRN,DTBURN, AVREXT
      WRITE (6,3333)
      WRITE (6,9876) MSTART, ZMAST, FMIN, RESIDL
      WRITE (6,3333)
      WRITE (6,1261) HEATI, WORKI
      IF (FIRE) WRITE (6,1264) HEATC, WORKC
      IF (FIRE) WRITE (6,1262) HEATCE, WORKCE
      IF (.NOT. FIRE) WRITE (6,1262) HEATCE, WORKCE
      WRITE (6,1263) HEATE, WORKE
      WRITE (6,3333)
      WRITE (6,1890) TOHIN, TOHEX, TOHEAT, TOWORK, DY(20), DY(3),
     &               DECYCL, DEONHI, DEONQ
      WRITE (6,3333)
C
C         NO CONVERGENCE CHECK IS PERFORMED AFTER THE FIRST ITERATION
C         BECAUSE THE CREVICE AND LEAKAGE MODEL IS NOT YET ACTIVATED
C
```

```
      IF (ITERAS .EQ. 1)  GO TO 470
      IF (ABS( (Y(12) - PSTART)/PSTART ) .GT. 0.02) GO TO 470
      IF (ABS( (Y(11) - TSTART)/TSTART ) .GT. 0.02) GO TO 470
      IF (ABS( (Y(1) - Y(2))/MSTART )    .GT. 0.02) GO TO 470
C
       GO TO 471
  470 CONTINUE
  471 CONTINUE
CJ
CJ    CALL WRITE
C
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C        FORMAT STATEMENTS
C
 2225 FORMAT (1H1)
C
  881 FORMAT (1H ,2X,'CA = ',F8.2,10X,'P = ',F10.5,9X,'INFLAG = ',I2,
     &         8X,'IFG = ',I2)
C
  882 FORMAT (1H ,2X,'CA = ',F6.2,10X,'P = ',F10.5,28X,'IFG = ',I2)
C
  883 FORMAT (1H ,2X,'CA = ',F6.2,10X,'P = ',F10.5,9X,'XB = ',F9.6,
     &         5X,'IFG = ',I2)
C
  449 FORMAT (////,1H ,13X,'START OF ITERATION  #',I2,2X,'OF ',I2,
     &         ' ALLOWED',//)
   77 FORMAT (//(54X,' >>>>> INPUT DATA <<<<< ' )//)
C
 4595 FORMAT (/////(1X,' >>>>> START OF INTAKE PROCESS      ')//)
C
 4596 FORMAT ((4X,'CA',7X,'P',9X,'TEMP',7X,'MIN',6X,'MEX',9X,'VIV',7X,
     &         'VEV',9X,'X1',9X,'Q DOT',9X,'WORK',8X,'IMF   IFG')/
     &         (2X,'(DEG)',4X,'(ATM)',7X,'(K)',8X,'(G)',6X,'(G)',7X,
     &         ' (CM/SEC)',2X,' (CM/SEC)',6X,'(-)',7X,'(KJ/DEG)',7X,
     &         '(KJ)'))
C
 4597 FORMAT ((4X,'CA',7X,'P',9X,'TEMP',61X,
     &         'Q DOT',9X,'WORK',14X,'IFG')/
     &         (2X,'(DEG)',4X,'(ATM)',7X,'(K)',61X,
     &         '(KJ/DEG)',7X,
     &         '(KJ)'))
C
 4598 FORMAT ((4X,'CA',7X,'P',22X,'TB ',7X,'TU ',4X,6X,4X,
     &         'XBURND',19X,'Q DOT',9X,'WORK',14X,'IFG')/
     &         (2X,'(DEG)',4X,'(ATM)',20X,'(K)',7X,'(K)',5X,
     &         3X,7X,'(-)',20X,'(KJ/DEG)',7X,
     &         '(KJ)'))
C
 4599 FORMAT ((4X,'CA',7X,'P',9X,'TEMP',16X,'MEX',19X,
     &         'VEV',20X,'Q DOT',9X,'WORK',14X,'IFG')/
     &         (2X,'(DEG)',4X,'(ATM)',7X,'(K)',17X,'(G)',17X,
     &         ' (M/SEC)',16X,'(KJ/DEG)',7X,
     &         '(KJ)'))
C
```

```
 1110 FORMAT (///(1X,' >>>>> START OF COMPRESSION PROCESS      ')//)
C
 1111 FORMAT (///(1X,' >>>>> START OF EXHAUST PROCESS      ')//)
C
 1277 FORMAT (1F7.1,2X,F9.4,2X,F9.2,2X,2F10.5,2X,F8.1,2X,11X,
     &        F9.5,16X,1F10.6)
C
 1210 FORMAT (1F7.1,2X,F9.4,2X,F9.2,2X,2F10.5,2X,F8.1,2X,F8.1,3X,
     &        F9.5,3X,1F10.6,3X,1F10.6,3X,1I4,1I6)
C
 1211 FORMAT (1F7.1,2X,F9.4,2X,F9.2,57X,
     &        1F10.6,3X,1F10.6,9X,1I4)
C
 1212 FORMAT (1F7.1,2X,F9.4,13X,2F10.2,2X,8X,2X,F8.5,15X,
     &        1F10.6,3X,1F10.6,9X,1I4)
C
 1213 FORMAT (1F7.1,2X,F9.4,2X,F9.2,12X,1F10.5,12X,F8.1,15X,
     &        1F10.6,3X,1F10.6,9X,1I4)
C
 9876 FORMAT ( /('     MASS IN CYLINDER AT TIVO = ',F8.5,' G')//
     &          ('     MASS IN CYLINDER AT TIVC = ',F8.5,' G')//
     &          ('     MASS OF FUEL INDUCTED    = ',F8.5,' G')//
     &          ('     RESIDUAL FRACTION        = ',F8.5)/)
C
 1890 FORMAT ( /('     TOTAL ENTHALPY IN / CYCLE   = ',F8.5,' KJ')//
     &          ('     TOTAL ENTHALPY OUT / CYCLE  = ',F8.5,' KJ')//
     &          ('     TOTAL HEAT LOSS / CYCLE     = ',F8.5,' KJ')//
     &          ('     TOTAL WORK OUTPUT / CYCLE   = ',F8.5,' KJ')//
     &          ('     HEAT LOSS TO CREVICE/CYCLE  = ',F8.5,' KJ')//
     &          ('     "LOST" FUEL ENERGY          = ',F8.5,' KJ')//
     &          ('     NET ENERGY GAIN / CYCLE     = ',F8.5,' KJ')//
     &          ('     (ENERGY GAIN)/(ENTHALPY IN) = ',F8.5,' %')//
     &          ('     (ENERGY GAIN)/(MFUEL*LHV)   = ',F8.5,' %')/)
C
 1261 FORMAT ( /('     HEATI   =  ',F10.6,' KJ','  (TIPO   -   -270)')/
     &          ('     WORKI   =  ',F10.6,' KJ')/ )
C
 1281 FORMAT (///('  VOLUMETRIC EFFICIENCY = ',1F5.1,' %')//)
C
 1262 FORMAT (  /('     HEATCE  =  ',F10.6,' KJ','  (TIPC   -    +270')/
     &          ('     WORKCE  =  ',F10.6,' KJ')/ )
C
 1263 FORMAT (  /('     HEATE   =  ',F10.6,' KJ','  (+270   -   TIPO)')/
     &          ('     WORKE   =  ',F10.6,' KJ')/ )
C
 1264 FORMAT (  /('     HEATC   =  ',F10.6,' KJ','  (-270   - TSPARK)')/
     &          ('     WORKC   =  ',F10.6,' KJ')/ )
C
 3210 FORMAT (5X,1F7.1,2X,2(1F12.1,2X),2(1F12.2,2X),
     &       2X,3(F10.5,4X),F9.4)
C
 4210 FORMAT (5X,1F7.1,2X,F10.1,3X,3(F11.1,3X),4X,3(F12.3,5X))
C
 9210 FORMAT (5X,1F7.1,6X,F8.4,4X,F10.6,1X,5(6X,F10.6))
 9211 FORMAT (5X,F7.1,6X,F8.4)
```

```
 1112 FORMAT (///(1X,' >>>>> START OF COMBUSTION AND EXPANSION PROCESSES
     &        ')///)
C
 4592 FORMAT (//(9X,'CA',7X,'VELHTX',7X,'HTPARO',7X,'HTPASI',7X,
     &        'HTPAHO',13X,'Q% ROTOR',10X,'Q% SIDE',8X,'Q% HOUSING')/
     &        (8X,'(DEG)',4X,'(CM/SEC)',5X,'(KW/M**2)',4X,'(KW/M**2)',
     &        4X,'(KW/M**2)',14X,'(%)',14X,'(%)',14X,'(%)'))
C
 4594 FORMAT (//(9X,'CA',7X,'MEANKE',8X,'TURBKE',8X,' VMKE ',8X,
     &        'UPRIME',10X,'MACRSC',8X,'MICRSC',9X,'SSUBL',8X,'BTIMSC')/
     &        (8X,'(DEG)',5X,'(ERG)',9X,'(ERG)',8X,'(CM/SEC)',6X,
     &       '(CM/SEC)',10X,'(CM)',10X,'(CM)',9X,'(CM/SEC)',7X,'(MS)'))
C
 9900 FORMAT (//(9X,'CA',9X,'CHAMBER',5X,'LEAD CREVICE',5X,
     &        'LAG CREVICE',5X,'LEAD LEAKAGE',4X,'LAG LEAKAGE',5X,
     &        'LEAD CREVICE',5X,'LAG CREVICE')/
     &        (20X,'MASS',8X,'MASS',13X,'MASS',12X,'MASS',12X,'MASS',
     &        12X,'COMPOSITION',6X,'COMPOSITION')/
     &        (7X,'(DEG)',10X,'(G)',12X,'(G)',13X,'(G)',14X,'( )',
     &        13X,'( )')//)
C
C3111 FORMAT (///,1H ,59X,'NOX FORMATION')
C
C5111 FORMAT (///,1H ,48X,'ADIABATIC CORE / BOUNDARY LAYER DATA')
C
C4111 FORMAT (///,1H ,54X,'FLAME PROPAGATION DATA')
C
C6111 FORMAT (///,1H ,55X,'TURBULENT FLOW MODEL')
C
 7111 FORMAT (///,1H ,56X,'HEAT TRANSFER DATA')
C
C
 8111 FORMAT (///,1H ,40X,'LEAKAGE AND CREVICE VOLUME DATA')
C
C3222 FORMAT (//,1H ,6X,'CA',7X,'YAC',7X,'YBL',7X,'YNOAC',5X,'YNOBL',5X,
C    &       'YNO',6X,'XNOAC',6X,'XNOBL',5X,'XNO',8X,'PPMAC',6X,
C    &       'PPMBL',6X,'PPMNO')
C
C4222 FORMAT (//,1H ,6X,'CA',5X,'VENONV',5X,'VBRONV',5X,'DFLONB',5X,
C    &       'AFLONB',5X,'AHUONB',5X,'AHBONB',5X,'APUONB',5X,'APBONB',
C    &       5X,'ACUONB',5X,'ACBONB')
C
C5222 FORMAT (//,1H ,6X,'CA',8X,'YAC',8X,'YBL',8X,'VACONV',6X,'VBLONV',
C    &       6X,'DBLONB',10X,'TWALLB',7X,'TB',9X,'TAC',8X,'TBLAYR')
C
 3333 FORMAT (1H ,1X,'____________________________________________',
     &        '____________________________________________',
     &        '________________________',/)
C
C3444 FORMAT (1H ,3X,F8.1,8(2X,F8.6),3(2X,F9.2))
C
C5444 FORMAT (1H ,2X,F8.1,5(4X,F8.6),2X,4(3X,F9.1))
C
C6444 FORMAT (1H ,2X,F8.1,10(3X,F8.5))
C
```

```
 5910 FORMAT (///(46X,'>--------------+--------------<')/
     &           (46X,'>                              <')/
     &           (46X,'>     CALCULATION RESULTS      <')/
     &           (46X,'>                              <')/
     &           (46X,'>--------------+--------------<')///)
C
 5920 FORMAT (/(33X,' -->  VOLUMETRIC EFFICIENCY; (%) ')/
     & (33X,'      BASED ON: INTAKE / ATM            ----> ',2(F8.1))///
     & (33X,' -->  PUMPING MEAN EFFECTIVE  ')/
     & (33X,'      PRESSURE; (KPA) : PEMP            ---->  ',1F6.0)///
     & (33X,' -->  GROSS INDICATED MEAN EFFECTIVE ')/
     & (33X,'      PRESSURE; (KPA) : IMEP            ---->  ',1F6.0)///
     & (33X,' -->  GROSS INDICATED SPECIFIC FUEL  ')/
     & (33X,'      CONSUMPTION; (G/IKW-HR) : ISFC    ---->  ',1F6.0)///
     & (33X,' -->  GROSS INDICATED THERMAL      ')/
     & (33X,'      EFFICIENCY; (%)                   ---->  ',1F7.1)///
     & (33X,' -->  NET INDICATED THERMAL        ')/
     & (33X,'      EFFICIENCY; (%)                   ---->  ',1F7.1)///
     & (33X,' -->  (HEAT TRANSFER PER CYCLE)/   ')/
     & (33X,'      (MASS OF FUEL TIMES LHV); (%)     ---->  ',1F7.1)//)
 5921 FORMAT ((33X,' -->  IGNITION DELAY (0 - 10%)     ')/
     & (33X,'      (CRANK ANGLE) / (MS)              ---->  ',2(F8.2))///
     & (33X,' -->  BURN DURATION (10 - 90%)     ')/
     & (33X,'      (CRANK ANGLE) / (MS)              ---->  ',2(F8.2))///
     & (33X,' -->  MEAN EXHAUST                 ')/
     & (33X,'      TEMPERATURE; (K)                  ---->  ',1F7.1)/)
 2901 FORMAT (/////,1H ,39X,'M.I.T. ZERO-DIMENSIONAL WANKEL ENGINE',
     &        ' CYCLE SIMULATION',/////)
 2902 FORMAT (/,1H ,10X,'>>>>> OPERATING MODE',/)
 2903 FORMAT (/,1H ,25X,'FIRING CYCLE')
 2904 FORMAT (/,1H ,25X,'MOTORED CYCLE',/)
 2905 FORMAT (/,1H ,25X,'SPECIFIED BURN RATE')
 2906 FORMAT (/,1H ,30X,'BURN DURATION    = ',F8.3,' DEG CA',
     &        /,1H ,30X,'WIEBE CONSTANT   = ',F8.3,
     &        /,1H ,30X,'WIEBE EXPONENT   = ',F8.3,/)
 2907 FORMAT (/,1H ,25X,'PREDICTED BURN RATE',/)
 2908 FORMAT (/,1H ,10X,'>>>>> OPERATING CONDITIONS',/)
 2909 FORMAT (/,1H ,25X,'FUEL USED IS ISOOCTANE')
 2910 FORMAT (/,1H ,25X,'FUEL USED IS PROPANE')
 2911 FORMAT (/,1H ,25X,'F/A EQUIVALENCE RATIO = ',F9.3)
 2912 FORMAT (/,1H ,25X,'SPARK TIMING          = ',F8.2,' DEG CA')
 2913 FORMAT (/,1H ,25X,'ENGINE SPEED          = ',F7.1,' RPM',/)
 2914 FORMAT (/,1H ,10X,'>>>>> MANIFOLD CONDITIONS',/)
 2915 FORMAT (/,1H ,25X,'INTAKE MANIFOLD PRESSURE  = ',F10.4,' ATM',/
     &        /,1H ,25X,'EXHAUST MANIFOLD PRESSURE = ',F10.4,' ATM',/
     &        /,1H ,25X,'FRESH CHARGE TEMPERATURE  = ',F8.2,'   K',/
     &        /,1H ,25X,'EXHAUST GAS RECIRCULATION = ',F8.2,'   %',/
     &        /,1H ,25X,'EGR TEMPERATURE           = ',F8.2,'   K',/
     &        /,1H ,25X,'INTAKE CHARGE TEMPERATURE = ',F8.2,'   K',/
     &        /,1H ,25X,'ATMOSPHERIC PRESSURE      = ',F10.4,' ATM',/
     &        /,1H ,25X,'ATMOSPHERIC TEMPERATURE   = ',F8.2,'   K',/)
 2916 FORMAT (/,1H ,10X,'>>>>> HEAT TRANSFER AND TURBULENCE',
     &                 ' PARAMETERS',/)
 2917 FORMAT (/,1H ,25X,'HEAT TRANSFER CONSTANT         = ',F10.4,/
     &        /,1H ,25X,'HEAT TRANSFER EXPONENT         = ',F10.4,/
```

```
     &        /,1H ,25X,'ROTOR TEMPERATURE              =',F9.2,'   K',/
     &        /,1H ,25X,'SIDE WALL TEMPERATURE          =',F9.2,'   K',/
     &        /,1H ,25X,'HOUSING WALL TEMPERATURE       =',F9.2,'   K',/
     &        )
 2918 FORMAT (/,1H ,10X,'>>>>> ENGINE DESIGN PARAMETERS',/)
 2919 FORMAT (/,1H ,25X,'ECCENTRICITY OF ROTOR   = ',F9.3,' CM',/
     &        /,1H ,25X,'RADIUS OF ROTOR         = ',F9.3,' CM',/
     &        /,1H ,25X,'DEPTH OF CHAMBER        = ',F9.3,' CM',/
     &        /,1H ,25X,'COMPRESSION RATIO       = ',F9.3,/
     &        /,1H ,25X,'DISPLACED VOLUME        = ',F9.3,' CC',/
     &        /,1H ,25X,'VOLUME OF ROTOR POCKET  = ',F9.3,' CC',/
     &        /,1H ,25X,'INTAKE PORT OPENS      = ',F7.1,'   DEG CA',/
     &        /,1H ,25X,'INTAKE PORT CLOSES     = ',F7.1,'   DEG CA',/
     &        /,1H ,25X,'EXHAUST PORT OPENS     = ',F7.1,'   DEG CA',/
     &        /,1H ,25X,'EXHAUST PORT CLOSES    = ',F7.1,'   DEG CA',/)
 2920 FORMAT (/,1H ,10X,'>>>>> COMPUTATIONAL PARAMETERS',/)
 2921 FORMAT (/,1H ,25X,'MAXIMUM # OF ITERATIONS =   ',I4,/
     &        /,1H ,25X,'OUTPUT AT ITERATION #   =   ',I4,/
     &        /,1H ,25X,'TPRINT                  = ',F9.2,
     &        /,1H ,25X,'TPRINX                  = ',F9.2,
     &        /,1H ,25X,'XBZERO                  = ',F12.5,
     &        /,1H ,25X,'XESTOP                  = ',F12.5,
     &        /,1H ,25X,'XBSTOP                  = ',F12.5,
     &        /,1H ,25X,'CIINTG                  = ',F13.6,
     &        /,1H ,25X,'CCINTG                  = ',F13.6,
     &        /,1H ,25X,'CBINTG                  = ',F13.6,
     &        /,1H ,25X,'CEINTG                  = ',F13.6,
     &        /,1H ,25X,'AREROT                  = ',F13.6,
     &        /,1H ,25X,'REL                     = ',F13.6,
     &        /,1H ,25X,'ERMAX                   = ',F13.6,
     &        /,1H ,25X,'MAXERR                  = ',F13.6,
     &        /,1H ,25X,'MAXTRY                  = ',I6,/)
 2940 FORMAT (/,1H , 10X,'>>>>> LEAKAGE AND CREVICE VOLUME PARAMETERS'/)
 2941 FORMAT (/,1H , 25X,'LEAK AREA PER APEX     = ',F12.6,'  CM*CM'
     &        //,1H , 25X,'CREVICE VOLUME PER APEX= ',F12.6,'  CC   '
     &        //,1H , 25X,'CREVICE GAS TEMPERATURE= ',F12.6,'  K'/)
  999 FORMAT (///,1H ,15X,
     &        'WARNING!!  RESULTS FROM THIS CYCLE SIMULATION',/,16X,
     &        'MAY NOT BE ACCURATE FOR PHI > 1.3 OR PHI < 0.7',///)
      STOP
      END
C-----------------------------------------------------------------------
C_______________________________________________________________________
C
C     SUBROUTINE INTAKE
C
C     PURPOSE
C        CALCULATES OR ASSISTS THE CREVICE AND HEAT TRANSFER SUBROUTINES
C        CALCULATE THE TIME RATE OF CHANGE OF PRESSURE, TEMPERATURE,
C        MASS, HEAT TRANSFER,AND WORK TRANSFER IN THE CHAMBER DURING
C        INTAKE.
C
C     USAGE
C        CALL INTAKE (DT, DY, DYP)
C
```

```
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(1)      NO     NO       MASS INDUCTED INTO CHAMBER THROUGH
C        -----      --     --       INTAKE PORT (G)
C        DY(2)      NO     NO       MASS EXHAUSTED FROM CHAMBER THROUGH
C        -----      --     --       EXHAUST PORT (G)
C        DY(5)      YES    NO       MASS FRACTION OF FRESH CHARGE (-)
C        DY(8)      NO     NO       HEAT TRANSFER TO ROTOR (KJ)
C        DY(9)      NO     NO       HEAT TRANSFER TO SIDE PLATES (KJ)
C        DY(10)     NO     NO       HEAT TRANSFER TO HOUSING (KJ)
C        DY(11)     YES    NO       CHAMBER TEMPERATURE (K)
C        DY(12)     YES    NO       CHAMBER PRESSURE (ATM)
C        DY(16)     NO     NO       TOTAL WORK TRANSFER (KJ)
C        DY(17)     NO     NO       TOTAL ENTHALPY EXHAUSTED (KJ)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN LEAD CREVICE  (-)
C        DY(22)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
C        DY(24)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       LEADING CHAMBER OR CREVICE (G)
C        DY(25)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       TRAILING CHAMBER OR CREVICE (G)
C
C        ------------------------------------------------------------
C
C        DYP(1)     NO     YES      RATE AT WHICH MASS IS INDUCTED
C        ------     --     ---      THROUGH THE INTAKE PORT  (G/DEG)
C        DYP(2)     NO     YES      RATE AT WHICH MASS IS EXHAUSTED
C        ------     --     ---      THROUGH THE EXHAUST PORT (G/DEG)
C        DYP(5)     NO     YES      RATE OF CHANGE OF MASS FRACTION OF
C        ------     --     ---      FRESH CHARGE (1/DEG)
C        DYP(8)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      ROTOR WALL  (KJ/DEG)
C        DYP(9)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      SIDE PLATES (KJ/DEG)
C        DYP(10)    NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      HOUSING  (KJ/DEG)
C        DYP(11)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      TEMPERATURE (K/DEG)
C        DYP(12)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      PRESSURE (ATM/DEG)
C        DYP(16)    NO     YES      RATE OF TOTAL WORK TRANSFER (KJ/DEG)
C        DYP(17)    NO     YES      RATE AT WHICH TOTAL ENTHALPY IS
C        -------    --     ---      EXHAUSTED (KJ/DEG)
C        DYP(18)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
```

```
C        -------    --     ---      LEAD APEX SEAL (G/DEG)
C        DYP(19)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---      TRAILING APEX SEAL (G/DEG)
C        DYP(20)    NO     YES      RATE OF HEAT TRANSFER TO CREVICE
C        -------    --     ---      VOLUME WALLS (KJ/DEG)
C        DYP(21)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN LEAD CREVICE  ( )
C        DYP(22)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN TRAILING CREVICE ( )
C        DYP(23)    NO     YES      RATE OF CHANGE OF TOTAL CHAMBER
C        -------    --     ---      MASS (G/DEG)
C        DYP(24)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER LEAD CREVICE AND
C        -------    --     ---      LEAD CHAMBER (G/DEG)
C        DYP(25)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER TRAILING CREVICE
C        -------    --     ---      AND TRAILING CHAMBER (G/DEG)
C
C
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        UTHRMO     UTRANS     IPACD      MFLRT
C        CSAVDV     EPACD      CREVIC     HEATTX
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY S. H. MANSOURI, S. G. POULOS, AND T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE INTAKE (DT, DY, DYP)
C
      REAL*8 DT, DY(25), DYP(25)
      REAL MW, MWIM, MWIMM, MASS, MDOT, MDOTFR, MSTART,
     &     MAXERR, MWBL, MWAC, IPA
      DIMENSION Y(25), YP(25)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, AFLANK, VFLANK
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/MANFP/ PIM, TIM, EGR, PEM, MSTART
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/IMTHP/ HIM, MWIM, GIM, RHOIM
      COMMON/FIXX/ INFLAG
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/HTRAN/THTRAN
      COMMON/YYYY1/ VIP, VEP
      COMMON/RHMAS/ RHO, MASS, VOLUME, H, GAMMA
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
      COMMON/BURN/SPBURN,FIRE,FIREFL
```

```
C
      VIP = 0.0
      VEP = 0.0
C
      DO 10  I = 1, 25
           Y(I) = DY(I)
   10 CONTINUE
      T = DT
      DO 20 I = 1, 25
           YP(I) = 0.0
   20 CONTINUE
C
C        FIND THERMODYNAMIC AND TRANSPORT PROPERTIES IN CHAMBER
C
      RESFRK = 1. - Y(5)
      CALL THERMO (T, Y(11), Y(12), RESFRK, H, CSUBP, CSUBT,
     & RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
      MASS = Y(23)
C
C        FIND OUT IF INTAKE PORT IS OPEN.
C
      IF (T .GE. TIPC) GO TO 50
C
C        YES IT IS.
C        FIND OUT IF ANY MASS FLOWS ACROSS INTAKE PORT.
C
      IF (PIM - Y(12)) 30, 50, 40
C
C        REVERSE FLOW PAST PORT.
C        CALCULATE CD AND EFFECTIVE AREA.
C
   30 PR = Y(12)/PIM
      CALL IPACD (T, AREA, CD)
C
C        CALCULATE MASS FLOW RATE FROM CHAMBER TO INTAKE MANIFOLD.
C
      CALL MFLRT (CD, AREA, Y(12), MW, Y(11), PIM, GAMMA, FRAIV)
C
C        CALCULATE RATES DUE TO THIS FLOW.
C
      YP(1) = -FRAIV
      IF (AREA .LE. 0.0) GO TO 35
      VIP = -FRAIV/(RHO * AREA)
   35 HIMM = H
      GO TO 50
C
C        FLOW INTO CHAMBER.
C        CALCULATE CD AND AREA.
C
   40 PR = PIM/Y(12)
      CALL IPACD (T, AREA, CD)
C
C        CALCULATE THERMODYNAMIC STATE OF MATERIAL FLOWING
C        INTO CHAMBER.
C        INFLAG = 0; CHAMBER GASES IN INTAKE MANIFOLD FLOWING BACK
```

```
C        INFLAG = 1; FRESH CHARGE (I.E. AIR, FUEL, AND EGR).
C
      TIMM = TIM * INFLAG + Y(11) * (1 - INFLAG)
      HIMM = HIM * INFLAG + H * (1 - INFLAG)
      MWIMM = MWIM * INFLAG + MW * (1 - INFLAG)
      GIMM = GIM * INFLAG + GAMMA * (1 - INFLAG)
      RHOIMM = RHOIM * INFLAG + RHO * (1 - INFLAG)
C
C        CALCULATE MASS FLOW RATE
C
      CALL MFLRT (CD, AREA, PIM, MWIMM, TIMM, Y(12), GIMM, FRAIV)
C
C        CALCULATE RATES DUE TO THIS FLOW
C
      YP(1) = FRAIV
      IF (AREA .LE. 0.0) GO TO 50
      VIP = FRAIV/(RHOIMM * AREA)
C
C        IS EXHAUST PORT STILL OPEN ?
C
   50 IF ((T + 1080.) .GE. TEPC) GO TO 80
C
C        YES IT IS.
C        ANY FLOW ACROSS IT ?
C
      IF (Y(12) - PEM) 60, 80, 70
C
C        YES, FLOW INTO CHAMBER.
C        FIND CD AND AREA FOR EXHAUST PORT.
C
   60 PR = PEM/Y(12)
CT
CT        FOR A CORRECT CALCULATION OF THE EXHAUST PORT OPEN
CT        AREA AN ADJUSTED TIME MUST BE USED.
      TEP = T + 1080.
CT
      CALL EPACD (TEP, AREA, CD)
C
C        FIND MASS FLOW RATE.
C
      CALL MFLRT (CD, AREA, PEM, MW, Y(11), Y(12), GAMMA, FRAEV)
C
C        CALCULATE RATES DUE TO THIS FLOW.
C
      YP(2) = -FRAEV
      IF (AREA .LE. 0.0) GO TO 80
      VEP = -FRAEV/(RHO * AREA)
      GO TO 80
C
C        FLOW FROM CHAMBER INTO EXHAUST MANIFOLD.
C        FIND AREA AND CD FOR EXHAUST PORT.
C
   70 PR = Y(12)/PEM
C
C         FOR A CORRECT CALCULATION OF THE EXHAUST PORT OPEN AREA
```

```
C         AN ADJUSTED TIME MUST BE USED.
C
      TEP = T + 1080.
      CALL EPACD (TEP, AREA, CD)
C
C        FIND MASS FLOW RATE.
C
      CALL MFLRT (CD, AREA, Y(12), MW, Y(11), PEM, GAMMA, FRAEV)
C
C        CALCULATE RATES DUE TO THIS FLOW.
C
      YP(2) = FRAEV
      IF (AREA .LE. 0.0) GO TO 75
      VEP = FRAEV/(RHO * AREA)
   75 CONTINUE
C
C        FIND SURFACE AREAS AND VOLUME OF CHAMBER
C
   80 CALL CSAVDV (T, VOLUME, DVDT)
      MDOT = YP(1) - YP(2)
      MDOTFR = YP(1) * (1. - EGR/100.) * INFLAG - YP(2) * Y(5)
     &         + YP(1) * Y(5) * (1 - INFLAG)
C
C              CALCULATE HEAT TRANSFER RATES
C
C
         CALL HEATTX (T,Y,YP,THTRAN,XX1,XX2)
C
C
C
C        CALCULATE RATES OF CHANGE OF TEMPERATURE AND PRESSURE IN
C        THE CHAMBER.  THEN CALCULATE RATE OF DOING WORK.
C
   90 CALL CREVIC (T,Y,YP)
      YP(16) = Y(12) * DVDT * .101325E-3
C
      YP(8)  = YP(8) * 1.E-10
      YP(9)  = YP(9) * 1.E-10
      YP(10) = YP(10) * 1.E-10
      THTRAN = THTRAN * 1.E-10 * ESPDI
C
      YP(17) = YP(2) * H
C
C        CONVERT ALL TIME DERIVATIVES TO RATE PER CRANK
C        ANGLE DEGREE.
C
      DO 100 I = 1, 25
           DYP(I) = YP(I) * ESPDI
  100 CONTINUE
C
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE CMPRES
```

```
C
C     PURPOSE
C        CALCULATES OR ASSISTS THE CREVICE AND HEAT TRANSFER SUBROUTINES
C        CALCULATE THE TIME RATE OF CHANGE OF PRESSURE, TEMPERATURE,
C        MASS, HEAT TRANSFER, AND WORK TRANSFER IN THE CHAMBER DURING
C        COMPRESSION.
C
C     USAGE
C        CALL CMPRES (DT, DY, DYP)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(5)      YES    NO       MASS FRACTION OF FRESH CHARGE (-)
C        DY(8)      NO     NO       HEAT TRANSFER TO ROTOR (KJ)
C        DY(9)      NO     NO       HEAT TRANSFER TO SIDE PLATES (KJ)
C        DY(10)     NO     NO       HEAT TRANSFER TO HOUSING (KJ)
C        DY(11)     YES    NO       CHAMBER TEMPERATURE (K)
C        DY(12)     YES    NO       CHAMBER PRESSURE (ATM)
C        DY(16)     NO     NO       TOTAL WORK TRANSFER (KJ)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN LEAD CREVICE  (-)
C        DY(22)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
C        DY(24)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       LEADING CHAMBER OR CREVICE (G)
C        DY(25)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       TRAILING CHAMBER OR CREVICE (G)
C
C        ------------------------------------------------------------------
C        DYP(5)     NO     YES      RATE OF CHANGE OF MASS FRACTION OF
C        ------     --     ---      FRESH CHARGE (1/DEG)
C        DYP(8)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      ROTOR WALL  (KJ/DEG)
C        DYP(9)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      SIDE PLATES (KJ/DEG)
C        DYP(10)    NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      HOUSING  (KJ/DEG)
C        DYP(11)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      TEMPERATURE (K/DEG)
C        DYP(12)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      PRESSURE (ATM/DEG)
C        DYP(16)    NO     YES      RATE OF TOTAL WORK TRANSFER (KJ/DEG)
C        DYP(18)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---      LEAD APEX SEAL  (G/DEG)
C        DYP(19)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
```

```
C        -------    --     ---      TRAILING APEX SEAL (G/DEG)
C        DYP(20)    NO     YES      RATE OF HEAT TRANSFER TO CREVICE
C        -------    --     ---      VOLUME WALLS  (KJ/DEG)
C        DYP(21)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN LEAD CREVICE  (/DEG)
C        DYP(22)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN TRAILING CREVICE (/DEG)
C        DYP(23)    NO     YES      RATE OF CHANGE OF TOTAL CHAMBER
C        -------    --     ---      MASS (G/DEG)
C        DYP(24)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER LEAD CREVICE AND
C        -------    --     ---      LEAD CHAMBER (G/DEG)
C        DYP(25)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER TRAILING CREVICE
C        -------    --     ---      AND TRAILING CHAMBER (G/DEG)
C
C
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        UTHRMO     UTRANS     CSAVDV    CREVIC    HEATTX
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY S. H. MANSOURI, S. G. POULOS, AND T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE CMPRES (DT,DY,DYP)
C
      REAL*8 DT, DY(25), DYP(25)
      REAL MW, MWIM, MWIMM, MASS, MDOT, MDOTFR, MSTART
      DIMENSION Y(25), YP(25)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, VFLANK
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/MANFP/ PIM, TIM, EGR, PEM, MSTART
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/IMTHP/ HIM, MWIM, GIM, RHOIM
      COMMON/FIXX/ INFLAG
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/HTRAN/THTRAN
      COMMON/YYYY1/ VIP, VEP
      COMMON/RHMAS/ RHO, MASS, VOLUME, H, GAMMA
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
      COMMON/BURN/SPBURN,FIRE,FIREFL
C
C
      DO 10  I = 1, 25
           Y(I) = DY(I)
```

```
   10 CONTINUE
      T = DT
      DO 20 I = 1, 25
           YP(I) = 0.0
   20 CONTINUE
C
C        FIND THERMODYNAMIC AND TRANSPORT PROPERTIES IN CHAMBER
C
      RESFRK = 1. - Y(5)
      CALL THERMO (T, Y(11), Y(12), RESFRK, H, CSUBP, CSUBT,
     & RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
      MASS = Y(23)
C
C        FIND SURFACE AREAS AND VOLUME OF CHAMBER
C
      CALL CSAVDV (T, VOLUME, DVDT)
C
C                CALCULATE HEAT TRANSFER RATES
C
C
         CALL HEATTX (T,Y,YP,THTRAN,XX1,XX2)
C
C
C
C        CALCULATE RATES OF CHANGE OF TEMPERATURE AND PRESSURE IN
C        THE CHAMBER.  THEN CALCULATE RATE OF DOING WORK.
C
   30 CALL CREVIC (T,Y,YP)
      YP(16) = Y(12) * DVDT * .101325E-3
C
C         CONVERT THE HEAT TRANSFER RATES TO KILO JOULES
C
      YP(8)  = YP(8) * 1.E-10
      YP(9)  = YP(9) * 1.E-10
      YP(10) = YP(10) * 1.E-10
      THTRAN = THTRAN * 1.E-10 * ESPDI
C
C        CONVERT ALL TIME DERIVATIVES TO RATE PER CRANK
C        ANGLE DEGREE.
C
      DO 40 I = 1, 25
           DYP(I) = YP(I) * ESPDI
   40 CONTINUE
C
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE CMBSTN
C
C     PURPOSE
C        CALCULATES OR ASSISTS THE CREVICE AND HEAT TRANSFER SUBROUTINES
C        CALCULATE THE TIME RATE OF CHANGE OF PRESSURE, TEMPERATURE,
C        MASS, HEAT TRANSFER,AND WORK TRANSFERIN THE CHAMBER DURING
C        COMBUSTION.
```

```
C
C     USAGE
C        CALL CMBSTN (DT, DY, DYP)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(3)      NO     NO       FUEL ENERGY THAT ENTERS EXAUST
C        -----      --     --       CHAMBER (KJ)
C        DY(4)      YES    NO       MASS FRACTION BURNED (-)
C        DY(8)      NO     NO       HEAT TRANSFER TO ROTOR (KJ)
C        DY(9)      NO     NO       HEAT TRANSFER TO SIDE PLATES (KJ)
C        DY(10)     NO     NO       HEAT TRANSFER TO HOUSING (KJ)
C        DY(12)     YES    NO       CYLINDER PRESSURE (ATM)
C        DY(13)     YES    NO       TEMPERATURE OF UNBURNED MIXTURE
C        ------     ---    --       DURING COMBUSTION (K)
C        DY(14)     YES    NO       VOLUME OF UNBURNED MIXTURE
C        ------     ---    --       DURING COMBUSTION (CM**3)
C        DY(15)     YES    NO       TEMPERATURE OF BURNED PRODUCTS
C        ------     ---    --       DURING COMBUSTION (K)
C        DY(16)     NO     NO       TOTAL WORK TRANSFER (KJ)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN LEAD CREVICE  (-)
C        DY(22)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
C        DY(24)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       LEADING CHAMBER OR CREVICE (G)
C        DY(25)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       TRAILING CHAMBER OR CREVICE (G)
C
C
C        ------------------------------------------------------------
C
C        DY(3)      NO     NO       FUEL ENERGY THAT ENTERS EXAUST
C        -----      --     --       CHAMBER (KJ)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     YES    NO       MASS FRACTION OF FRESH CHARGE
C        ------     ---    --       IN LEAD CREVICE  (-)
C        DY(22)     YES    NO       MASS FRACTION OF FRESH CHARGE
C        ------     ---    --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
```

```
C        DY(24)      YES    NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------      ---    --       LEADING CHAMBER OR CREVICE (G)
C        DY(25)      YES    NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------      ---    --       TRAILING CHAMBER OR CREVICE (G)
C
C        ------------------------------------------------------------------
C
C        DYP(3)      NO     YES      RATE OF FUEL ENTERING EXHAUST
C        ------      --     ---      CHAMBER.
C        DYP(4)      NO     YES      RATE OF CHANGE OF MASS FRACTION
C        ------      --     ---      BURNED (1/DEG)
C        DYP(8)      NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------     --     ---      ROTOR WALL  (KJ/DEG)
C        DYP(9)      NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------     --     ---      SIDE PLATES (KJ/DEG)
C        DYP(10)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------     --     ---      HOUSING  (KJ/DEG)
C        DYP(12)     NO     YES      RATE OF CHANGE OF CYLINDER
C        ------      --     ---      PRESSURE (ATM/DEG)
C        DYP(13)     NO     YES      RATE OF CHANGE OF UNBURNED MIXTURE
C        ------      --     ---      TEMPERATURE DURING COMBUSTION
C        ------      --     ---      (K/DEG)
C        DYP(14)     NO     YES      RATE OF CHANGE OF UNBURNED MIXTURE
C        -------     --     ---      VOLUME DURING COMBUSTION (CM**3/DEG)
C        DYP(15)     NO     YES      RATE OF CHANGE OF BURNED PRODUCTS
C        ------      --     ---      TEMPERATURE DURING COMBUSTION
C        ------      --     ---      (K/DEG)
C        DYP(16)     NO     YES      RATE OF TOTAL WORK TRANSFER (KJ/DEG)
C        DYP(18)     NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------     --     ---      LEAD APEX SEAL (G/DEG)
C        DYP(19)     NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------     --     ---      TRAILING APEX SEAL  (G/DEG)
C        DYP(20)     NO     YES      RATE OF HEAT TRANSFER TO CREVICE
C        -------     --     ---      VOLUME WALLS  (KJ/DEG)
C        DYP(21)     NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------     --     ---      FRACTION IN LEAD CREVICE  (/DEG)
C        DYP(22)     NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------     --     ---      FRACTION IN TRAILING CREVICE (/DEG)
C        DYP(23)     NO     YES      RATE OF CHANGE OF TOTAL CHAMBER
C        -------     --     ---      MASS (G/DEG)
C        DYP(24)     NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------     --     ---      CHAMBER TO ENTER LEAD CREVICE AND
C        -------     --     ---      LEAD CHAMBER (G/DEG)
C        DYP(25)     NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------     --     ---      CHAMBER TO ENTER TRAILING CREVICE
C        -------     --     ---      AND TRAILING CHAMBER (G/DEG)
C
C
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        UTHRMO     UTRANS     CSAVDV     CREVIC     HEATTX
C
```

```
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY S. H. MANSOURI, K. RADHAKRISHNAN, S. G. POULOS, AND
C                T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE CMBSTN (DT,DY,DYP)
C
      LOGICAL SPBURN,BURN,FIREFL
      REAL*8 DT,DY(25),DYP(25), D(4), XMOFR(14), TMP, PRS
      REAL MW,MWIM,MWIMM,MASS,MDOT,MDOTFR,MSTART,
     &     MASSU
      DIMENSION Y(25),YP(25)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, VFLANK
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/MANFP/ PIM,TIM,EGR,PEM, MSTART
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/IMTHP/ HIM,MWIM,GIM,RHOIM
      COMMON/FIXX/ INFLAG
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/HTRAN/THTRAN
      COMMON/UBHEAT/ UHTRCO, BHTRCO
      COMMON/YYYY1/ VIP, VEP
      COMMON/CHEM/ D
      COMMON/SPECB/ DTBRN, CONSPB, EXSPB
      COMMON/XSTOP/ XBSTOP
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
      COMMON/RHMAS/RHO,MASS,VOLUME,H,GAMMA,GAMAB
      COMMON/CREVCB/HU,HB,DRODPU,DRODPB,RHOU,RHOB,DRODTU,DRODTB,
     &              CSUBPU,CSUBPB,CSUBTU,CSUBTB,UHTRAN,BHTRAN
C
C
CJ    DATA CONSAM,CONSEM/1.0,1.0/
C
C
      FIREFL = .TRUE.
C
      DO 10  I = 1, 25
           Y(I) = DY(I)
   10 CONTINUE
      T = DT
      DO 20 I = 1, 25
           YP(I) = 0.0
   20 CONTINUE
C
C        FIND THERMODYNAMIC AND TRANSPORT PROPERTIES IN CYLINDER
C
      RESFRK = 1. - Y(5)
      CALL THERMO (T, Y(13), Y(12), RESFRK, HU, CSUBPU, CSUBTU,
```

```
     & RHOU, DRODTU, DRODPU, XXA, XXB, XXXA, XXXB, XXC, XXD)
      CALL THERMO (T, Y(15), Y(12), 1., HB, CSUBPB, CSUBTB,
     & RHOB, DRODTB, DRODPB, GAMAB, XXB, XXXA, XXXB, XXC, XXD)
C
C
      MASS  = Y(23)
      MASSU = MASS*(1.- Y(4) )
C
C
      CALL CSAVDV (T, VOLUME, DVDT)
      VBURND = VOLUME - Y(14)
C
C        SPECIFIED BURN RATE COMBUSTION MODEL
C
   30 TONDTB = (T - TSPARK)/DTBRN
      YP(4) = CONSPB*(EXSPB + 1.)*(TONDTB**EXSPB)*EXP( -CONSPB*
     &        TONDTB**(EXSPB + 1.) )/(DTBRN*ESPDI)
      IF (Y(4) .GE. XBSTOP) YP(4) = YP(4)/1.5
      IF (Y(4) .GE. 0.998)  YP(4) = YP(4)/1.5
      IF (Y(4) .GE. 0.999)  YP(4) = YP(4)/1.5
      IF (Y(4) .GE. 0.9999) YP(4) = 0.0
C
   40 CONTINUE
C
      CALL HEATTX (T, Y, YP, THTRAN, UHTRAN, BHTRAN)
C
C        CALCULATE RATES OF CHANGE OF TEMPERATURE AND PRESSURE IN
C        THE CYLINDER.  THEN CALCULATE RATE OF DOING WORK.
C
      CALL CREVIC (T,Y,YP)
C
      YP(16) = Y(12) * DVDT * .101325E-3
C
C        CONVERT TTHE HEAT TRANSFER RATES TO KILO JOULES
C
      YP(8)  = YP(8) * 1.E-10
      YP(9)  = YP(9) * 1.E-10
      YP(10) = YP(10) * 1.E-10
      THTRAN = THTRAN * 1.E-10 * ESPDI
C
C        CONVERT ALL TIME DERIVATIVES TO RATE PER CRANK
C        ANGLE DEGREE.
C
   70 DO 80 I = 1, 25
         DYP(I) = YP(I) * ESPDI
   80 CONTINUE
C
      FIREFL = .FALSE.
C
      RETURN
      END
C-----------------------------------------------------------------------
C
C     SUBROUTINE EXAUST
C
```

```
C     PURPOSE
C        CALCULATES OR ASSISTS THE CREVICE AND HEAT TRANSFER SUBROUTINES
C        CALCULATE THE TIME RATE OF CHANGE OF PRESSURE, TEMPERATURE,
C        MASS, HEAT TRANSFER, AND WORK TRANSFERIN THE CHAMBER DURING
C        EXAUST.
C
C     USAGE
C        CALL EXAUST (DT, DY, DYP)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(2)      NO     NO       MASS EXHAUSTED FROM CHAMBER THROUGH
C        -----      ---    --       EXHAUST PORT (G)
C        DY(3)      NO     NO       FUEL ENERGY THAT ENTERS EXAUST (KJ)
C        DY(8)      NO     NO       HEAT TRANSFER TO ROTOR (KJ)
C        DY(9)      NO     NO       HEAT TRANSFER TO SIDE PLATES (KJ)
C        DY(10)     NO     NO       HEAT TRANSFER TO HOUSING (KJ)
C        DY(11)     YES    NO       CHAMBER TEMPERATURE (K)
C        DY(12)     YES    NO       CHAMBER PRESSURE (ATM)
C        DY(16)     NO     NO       TOTAL WORK TRANSFER (KJ)
C        DY(17)     NO     NO       TOTAL ENTHALPY EXHAUSTED (KJ)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN LEAD CREVICE  (-)
C        DY(22)     NO     NO       MASS FRACTION OF FRESH CHARGE
C        ------     --     --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
C        DY(24)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       LEADING CHAMBER OR CREVICE (G)
C        DY(25)     NO     NO       TOTAL MASS THAT HAS LEFT AND ENTERED
C        ------     --     --       TRAILING CHAMBER OR CREVICE (G)
C
C        ------------------------------------------------------------
C
C        DYP(2)     NO     YES      RATE AT WHICH MASS IS EXHAUSTED
C        ------     --     ---      THROUGH THE EXHAUST PORT (G/DEG)
C        DYP(3)     NO     YES      RATE OF FUEL ENERGY ENTERING
C        ------     ---    ---      EXHAUST CHAMBER (KJ/DEG)
C        DYP(8)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      ROTOR WALL  (KJ/DEG)
C        DYP(9)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      SIDE PLATES (KJ/DEG)
C        DYP(10)    NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      HOUSING  (KJ/DEG)
C        DYP(11)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      TEMPERATURE (K/DEG)
C        DYP(12)    NO     YES      RATE OF CHANGE OF CHAMBER
```

```
C        ------     --     ---     PRESSURE (ATM/DEG)
C        DYP(16)    NO     YES     RATE OF TOTAL WORK TRANSFER (KJ/DEG)
C        DYP(17)    NO     YES     RATE AT WHICH TOTAL ENTHALPY IS
C        -------    --     ---     EXHAUSTED (KJ/DEG)
C        DYP(18)    NO     YES     RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---     LEAD APEX SEAL (G/DEG)
C        DYP(19)    NO     YES     RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---     TRAILING APEX SEAL (G/DEG)
C        DYP(20)    NO     YES     RATE OF HEAT TRANSFER TO CREVICE
C        -------    --     ---     VOLUME WALLS (KJ/DEG)
C        DYP(21)    NO     YES     RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---     FRACTION IN LEAD CREVICE (/DEG)
C        DYP(22)    NO     YES     RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---     FRACTION IN TRAILING CREVICE (/DEG)
C        DYP(23)    NO     YES     RATE OF CHANGE OF TOTAL CHAMBER
C        -------    --     ---     MASS (G/DEG)
C        DYP(24)    NO     YES     NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---     CHAMBER TO ENTER LEAD CREVICE AND
C        -------    --     ---     LEAD CHAMBER (G/DEG)
C        DYP(25)    NO     YES     NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---     CHAMBER TO ENTER TRAILING CREVICE
C        -------    --     ---     AND TRAILING CHAMBER  (G/DEG)
C
C
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        UTHRMO     BTRANS     MFLRT     CREVIC
C        CSAVDV     EPACD      HEATTX
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY S. H. MANSOURI, S. G. POULOS, AND T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE EXAUST (DT, DY, DYP)
C
      LOGICAL FIRE
      REAL*8 DT, DY(25), DYP(25)
      REAL MW, MWIM, MWIMM, MASS, MDOT, MDOTFR, MSTART
      DIMENSION Y(25), YP(25)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, VFLANK
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/MANFP/ PIM, TIM, EGR, PEM, MSTART
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/IMTHP/ HIM, MWIM, GIM, RHOIM
      COMMON/FIXX/ INFLAG
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
```

```
      COMMON/HTRAN/THTRAN
      COMMON/YYYY1/ VIP, VEP
      COMMON/RHMAS/ RHO, MASS, VOLUME, H, GAMMA
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
C
      VEP = 0.0
C
      DO 10  I = 1, 25
           Y(I) = DY(I)
   10 CONTINUE
      T = DT
      DO 20 I = 1, 25
           YP(I) = 0.0
   20 CONTINUE
C
C        FIND THERMODYNAMIC AND TRANSPORT PROPERTIES IN CHAMBER
C
      RESFRK = 1.
      IF (.NOT. FIRE) RESFRK = 0.0
      CALL THERMO (T, Y(11), Y(12), RESFRK, H, CSUBP, CSUBT,
     & RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
      MASS = Y(23)
C
C        IS EXHAUST PORT STILL OPEN ?
C
      IF (T .GE. TEPC) GO TO 50
C
C        YES IT IS.
C        ANY FLOW ACROSS IT ?
C
      IF (Y(12) - PEM) 30, 50, 40
C
C        YES, FLOW INTO CHAMBER.
C        FIND CD AND AREA FOR EXHAUST PORT.
C
   30 PR = PEM/Y(12)
      CALL EPACD (T, AREA, CD)
C
C        FIND MASS FLOW RATE.
C
      CALL MFLRT (CD, AREA, PEM, MW, Y(11), Y(12), GAMMA, FRAEV)
C
C        CALCULATE RATES DUE TO THIS FLOW.
C
      YP(2) = -FRAEV
      IF (AREA .LE. 0.0) GO TO 35
      VEP = -FRAEV/(RHO * AREA)
   35 CONTINUE
      GO TO 50
C
C        FLOW FROM CHAMBER INTO EXHAUST MANIFOLD.
C        FIND AREA AND CD FOR EXHAUST PORT.
C
   40 PR = Y(12)/PEM
```

```
      CALL EPACD (T, AREA, CD)
C
C        FIND MASS FLOW RATE.
C
      CALL MFLRT (CD, AREA, Y(12), MW, Y(11), PEM, GAMMA, FRAEV)
C
C        CALCULATE RATES DUE TO THIS FLOW
C
      YP(2) = FRAEV
      IF (AREA .LE. 0.0) GO TO 45
      VEP = FRAEV/(RHO * AREA)
   45 CONTINUE
C
C        FIND SURFACE AREAS AND VOLUME OF CHAMBER
C
   50 CALL CSAVDV (T, VOLUME, DVDT)
C
C               CALCULATE HEAT TRANSFER RATES
C
C
C
      CALL HEATTX (T, Y, YP, THTRAN, XX, XXX)
C
C        CALCULATE RATES OF CHANGE OF TEMPERATURE AND PRESSURE IN
C        THE CHAMBER.  THEN CALCULATE RATE OF DOING WORK.
C
   60 CALL CREVIC(T,Y,YP)
      YP(16) = Y(12) * DVDT * .101325E-3
C
      YP(8)  = YP(8) * 1.E-10
      YP(9)  = YP(9) * 1.E-10
      YP(10) = YP(10) * 1.E-10
      THTRAN = THTRAN * 1.E-10 * ESPDI
C
      YP(17) = YP(2) * H
C
C        CONVERT ALL TIME DERIVATIVES TO RATE PER CRANK
C        ANGLE DEGREE.
C
      DO 70 I = 1, 25
         DYP(I) = YP(I) * ESPDI
   70 CONTINUE
C
CJ
CJ    DY(21) = Y(21)
CJ    DY(22) = Y(22)
CJ
      RETURN
      END
C-----------------------------------------------------------------------
C
C
C     SUBROUTINE CREVICE
C
C     PURPOSE
```

```
C        TO CALCULATE THE LEAKAGE AND CREVICE VOLUME MASS FLOW RATES
C        AND COMPOSITIONS. BECAUSE AN ASSUMPTION FOR NET FLOW DIRECTION
C        AT EACH APEX MUST BE MADE, AND THEN CHECKED, SEVERAL OF THE
C        INTEGRATION VARIABLES ARE EVALUATED IN THIS SUBROUTINE
C        RATHER THAN THE PROCESS SUBROUTINES.
C
C     USAGE
C        CALL CREVIC (T,Y,YP)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(3)      NO     NO       FUEL ENERGY THAT ENTERS EXAUST
C        -----      --     --       CHAMBER (KJ)
C        DY(4)      NO     NO       MASS FRACTION BURNED (-)
C        DY(5)      YES    NO       MASS FRACTION OF FRESH CHARGE (-)
C        DY(11)     YES    NO       CHAMBER TEMPERATURE (K)
C        DY(12)     YES    NO       CHAMBER PRESSURE (ATM)
C        DY(13)     YES    NO       TEMPERATURE OF UNBURNED MIXTURE
C        ------     ---    --       DURING COMBUSTION (K)
C        DY(14)     YES    NO       VOLUME OF UNBURNED MIXTURE
C        ------     ---    --       DURING COMBUSTION (CM**3)
C        DY(15)     YES    NO       TEMPERATURE OF BURNED PRODUCTS
C        ------     ---    --       DURING COMBUSTION (K)
C        ------     --     ---      (K/DEG)
C        DY(18)     NO     NO       TOTAL MASS LEAKED PAST LEAD
C        ------     --     --       APEX SEAL  (G)
C        DY(19)     NO     NO       TOTAL MASS LEAKED PAST TRAILING
C        ------     --     --       APEX SEAL  (G)
C        DY(20)     NO     NO       TOTAL HEAT LOSS TO CREVICE VOLUME
C        ------     --     --       WALLS (KJ)
C        DY(21)     YES    NO       MASS FRACTION OF FRESH CHARGE
C        ------     ---    --       IN LEAD CREVICE  (-)
C        DY(22)     YES    NO       MASS FRACTION OF FRESH CHARGE
C        ------     ---    --       IN TRAILING CREVICE  (-)
C        DY(23)     YES    NO       TOTAL MASS IN CHAMBER  (G)
C        ------------------------------------------------------------
C
C        DYP(1)     YES    NO       RATE AT WHICH MASS IS INDUCTED
C        ------     --     --       THROUGH THE INTAKE PORT
C        DYP(2)     YES    NO       RATE AT WHICH MASS IS EXHAUSTED
C        ------     --     --       THROUGH THE EXHAUST PORT
C        DYP(3)     NO     YES      RATE OF FUEL ENTERING EXHAUST
C        ------     --     ---      CHAMBER.
C        DYP(4)     YES    NO       RATE OF CHANGE OF MASS FRACTION
C        ------     --     ---      BURNED
C        DYP(5)     NO     YES      RATE OF CHANGE OF MASS FRACTION OF
C        ------     --     ---      FRESH CHARGE IN CHAMBER
C        DYP(11)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      TEMPERATURE
C        DYP(12)    NO     YES      RATE OF CHANGE OF CHAMBER
C        ------     --     ---      PRESSURE
C        DYP(13)    NO     YES      RATE OF CHANGE OF UNBURNED MIXTURE
```

```
C        ------     --     ---      TEMPERATURE DURING COMBUSTION
C        DYP(14)    NO     YES      RATE OF CHANGE OF UNBURNED MIXTURE
C        -------    --     ---      VOLUME DURING COMBUSTION
C        DYP(15)    NO     YES      RATE OF CHANGE OF BURNED PRODUCTS
C        ------     --     ---      TEMPERATURE DURING COMBUSTION
C        DYP(18)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---      LEAD APEX SEAL
C        DYP(19)    NO     YES      RATE OF CHANGE OF MASS LEAKED PAST
C        -------    --     ---      TRAILING APEX SEAL
C        DYP(20)    NO     YES      RATE OF HEAT TRANSFER TO CREVICE
C        -------    --     ---      VOLUME WALLS
C        DYP(21)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN LEAD CREVICE
C        DYP(22)    NO     YES      RATE OF CHANGE OF FRESH CHARGE MASS
C        -------    --     ---      FRACTION IN TRAILING CREVICE
C        DYP(23)    NO     YES      RATE OF CHANGE OF TOTAL CHAMBER
C        -------    --     ---      MASS
C        DYP(24)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER LEAD CREVICE AND
C        -------    --     ---      LEAD CHAMBER
C        DYP(25)    NO     YES      NET RATE OF CHANGE OF MASS LEAVING
C        -------    --     ---      CHAMBER TO ENTER TRAILING CREVICE
C        -------    --     ---      AND TRAILING CHAMBER
C
C
C     REMARKS
C       - LEAD CHAMBER REFERS TO THE CHAMBER AHEAD (IN THE DIRECTION
C         OF ROTATION) OF THE THERMODYNAMIC SYSTEM.
C
C       - LAG (OR TRAILING) CHAMBER REFERS TO THE CHAMBER BEHIND.
C
C       - REFER TO PROCESS ROUTINES FOR UNITS OF INTEGRATION VARIABLES.
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        THERMO     MFLRT      TABLE
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C
      SUBROUTINE CREVIC (T,Y,YP)
C
      LOGICAL FIRE,FIREFL
      REAL MW, MW1, MW2, MWLDC, MWLAGC, MASS, MASLDC, MDOTDC, MASLGC,
     &     MDOTGC, LDLEAK, LGLEAK,MASSB,MASSU,MDOTLD,MDOTLG,MDOTCB
      DIMENSION Y(25), YP(25)
      COMMON/RHMAS/RHO,MASS,VOLUME,H,GAMMA
      COMMON/TEMPS/TPSTON,THEAD,TCW
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/HTRAN/THTRAN
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
```

```
      COMMON/CREV2/ MASLDC, MASLGC
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/FIXX/ INFLAG
      COMMON/MANFP/ PIM, TIM, EGR, PEM, MSTART
      COMMON/CREVCB/HU,HB,DRODPU,DRODPB,RHOU,RHOB,DRODTU,DRODTB,
     &              CSUBPU,CSUBPB,CSUBTU,CSUBTB,UHTRAN,BHTRAN
      COMMON/CREVQ/CRMASD,CRMASG,X1LDC,X1LAGC,ZMAST,ZMASS,ZLEAKD,ZLEAKG,
     &             ZCRCOD,ZCRCOG
      COMMON/CREVIN/AREALK,CREVOL,TCREV,X1LDIN,X1LGIN
C
C       FIND THE THERMODYNAMIC PROPERTIES OF THE FRESH CHARGE AND THE
C       BURNED GASES IN THE CHAMBER.
C
      TCR  = TCREV
      AREA = AREALK
      MASS = Y(23)
      VOLLDC = CREVOL
      VOLLGC = CREVOL
C
      RES = RESIDL
      IF ( T .LE. TIPC) RES = RESFIM
      CALL THERMO (T,Y(11),Y(12),RES,H1,XXA,XXB,XXC,XXD,XXE,XXF,
     &             MW1,XXG,XXH,XXI,XXJ)
      CALL THERMO (T,Y(11),Y(12),1.0,H2,XXA,XXB,XXC,XXD,XXE,XXF,MW2,
     &             XXG,XXH,XXI,XXJ)
C
C       DURING THE FIRST ITERATION THE CREVICE AND LEAKAGE MODELS
C       ARE INACTIVE BECAUSE NO PRESSURE HISTORY IS AVAILABLE.
C
            IF (ITERAS .NE. 1) GO TO 300
                   MASLDC = 0.0
                   MDOTDC = 0.0
                   MASLGC = 0.0
                   MDOTGC = 0.0
                   LDLEAK = 0.0
                   LGLEAK = 0.0
C
            IF ( FIREFL ) GO TO 150
            GO TO 90
C
C       FIND THE PRESSURES IN THE ADJACENT CHAMBERS AND THE COMPOSITION
C       OF THE TWO CREVICE VOLUMES FROM THE STORED VALUES OF THE LAST
C       ITERATION
C
  300 CALL TABLE (T,Y,PLEAD,PLAG,X1LDC,X1LAGC)
C
C       BECAUSE OF CHAMBER PRESSURE DIFFERENCES BETWEEN ITERATIONS
C       IT IS POSSIBLE THAT THE TABLE DOES NOT HOLD THE NECESSARY
C       CREVICE COMPOSITION INFORMATION NEAR THE 'SWITCHOVER' POINT
C       IF THIS OCCURS THE PROGRAM ASSIGNS THE LAST KNOWN VALUE OF
C       THE COMPOSITION FROM THE PREVIOUS ITERATION.
C
            IF (X1LDC .LE. 0.0) X1LDC = X1LDIN
            IF (X1LDC .GT. 1.0) X1LDC = X1LDIN
            IF (X1LAGC .LE. 0.0) X1LAGC = X1LGIN
```

```
            IF (X1LAGC .GT. 1.0) X1LAGC = X1LGIN
C
C
C       DETERMINE THE PRESSURE DIFFERENCE ACROSS EACH APEX SEAL
C
C
      IF (PLEAD .GE. Y(12)) GO TO 10
C
C         THE CHAMBER HAS A HIGHER PRESSURE THAN THE LEAD CHAMBER
C         SO THE LEAD CREVICE IS ASSOCIATED WITH THE CHAMBER AND
C         LEAKAGE FLOWS FROM THE LEAD CREVICE TO THE LEAD CHAMBER.
C
      PLEADC = Y(12)
      IF ( MASLDC .GT. 0.0   .AND.  Y(21) .NE. 0.0 )  X1LDC = Y(21)
           IF (CREVOL .GT. 0.0  .OR.  AREALK .LE. 0.0) GO TO 15
              X1LDC = Y(5)
              IF (FIREFL) X1LDC = 1.- Y(4)
              Y(21) = X1LDC
   15      CONTINUE
C
      CALL THERMO (T,TCR,PLEADC,1.-X1LDC,HLEADC,XXA,XXB,RHOLDC,XXD,XXE,
     &             GAMLDC,MWLDC,XXF,XXG,XXH,XXI)
      CALL MFLRT (1.0,AREA,PLEADC,MWLDC,TCR,PLEAD,GAMLDC,LDLEAK)
      MASLDC = RHOLDC * VOLLDC
      GO TO 20
C
C
C         THE LEAD CHAMBER HAS A HIGHER PRESSURE THAN THE  CHAMBER
C         CONSEQUENTLY THE LEAD CREVICE IS ASSOCIATED WITH THE LEAD
C         CHAMBER. THERE IS NO CREVICE FLOW AND LEAKAGE FLOWS FROM THE
C         LEAD CREVICE TO THE CHAMBER.
C
C
   10 PLEADC = PLEAD
      CALL THERMO (T,TCR,PLEADC,1.-X1LDC,HLEADC,XXA,XXB,XXC,XXD,XXE,
     &             GAMLDC,MWLDC,XXF,XXG,XXH,XXI)
      CALL MFLRT (1.0,AREA,PLEADC,MWLDC,TCR,Y(12),GAMLDC,LDLEAK)
      LDLEAK = -LDLEAK
      MASLDC = 0.0
      MDOTDC = 0.0
C
C
   20 IF (PLAG .GE. Y(12)) GO TO 30
C
C         THE CHAMBER HAS A HIGHER PRESSURE THAN THE LAG CHAMBER
C         SO THE LAG CREVICE IS ASSOCIATED WITH THE CHAMBER AND
C         LEAKAGE FLOWS FROM THE LAG CREVICE TO THE LAG CHAMBER.
C
      PLAGC = Y(12)
      IF ( MASLGC .GT. 0.0   .AND.  Y(22) .NE. 0.0 ) X1LAGC = Y(22)
           IF (CREVOL .GT. 0.0  .OR.  AREALK .LE. 0.0) GO TO 25
              X1LAGC = Y(5)
              IF (FIREFL) X1LAGC = 1.- Y(4)
              Y(22) = X1LAGC
   25      CONTINUE
```

```
C
      CALL THERMO (T,TCR,PLAGC,1.-X1LAGC,HLAGC,XXA,XXB,RHOLGC,XXD,XXE,
     &             GAMLGC,MWLAGC,XXF,XXG,XXH,XXI)
      CALL MFLRT (1.0,AREA,PLAGC,MWLAGC,TCR,PLAG,GAMLGC,LGLEAK)
      MASLGC = RHOLGC * VOLLGC
      GO TO 40


C
C         THE LAG CHAMBER HAS A HIGHER PRESSURE THAN THE  CHAMBER
C         CONSEQUENTLY THE LAG CREVICE IS ASSOCIATED WITH THE LAG
C         CHAMBER. THERE IS NO CREVICE FLOW, AND LEAKAGE FLOWS
C         FROM THE LAG CREVICE TO THE CHAMBER.
C
   30 PLAGC = PLAG
      CALL THERMO (T,TCR,PLAGC,1.-X1LAGC,HLAGC,XXA,XXB,XXC,XXD,XXE,
     &             GAMLGC,MWLAGC,XXF,XXG,XXH,XXI)
      CALL MFLRT (1.0,AREA,PLAGC,MWLAGC,TCR,Y(12),GAMLGC,LGLEAK)
      LGLEAK = -LGLEAK
      MASLGC = 0.0
      MDOTGC = 0.0
C
C          DURING COMBUSTION A DIFFERENT SET OF EQUATIONS
C          ARE USED.
C
   40         IF ( FIREFL ) GO TO 100
C
C
C          ASSUME THAT LEAD AND LAG MASS FLOWS (ALGEBRAIC SUM OF
C          LEAKAGE AND CREVICE VOLUME MASS FLOW) HAVE THE SAME SENSE
C          AS AT THE LAST TIME STEP.
C
      IF (MDOTLD .LE. 0.0) GO TO 50
C
C          LEAD FLOW IS ASSUMED TO BE OUT OF THE CHAMBER
C
              LDFLAG = +1
              X1LEAD = Y(5)
              HLEAD  = H
              GO TO 60
C
C          LEAD FLOW IS ASSUMED TO BE FROM THE LEAD CREVICE TO
C          THE CHAMBER.
C
   50         LDFLAG = -1
              X1LEAD = X1LDC
              HLEAD  = HLEADC
C
   60 IF (MDOTLG .LE. 0.0) GO TO 70
C
C          LAG FLOW IS ASSUMED TO BE OUT OF THE CHAMBER.
C
              LGFLAG = +1
              X1LAG = Y(5)
              HLAG  = H
```

```
              GO TO 90
C
C          LAG FLOW IS ASSUMED TO BE FROM THE LAG CREVICE TO
C          THE CHAMBER
C
   70         LGFLAG = -1
              X1LAG = X1LAGC
              HLAG  = HLAGC
C
C
   90 Y(12) = Y(12) * 1.01325E+6
C
C
      X1INT = (1.- RESFIM) * INFLAG + Y(5) * (1 - INFLAG)
      ASTAR = (1./MASS) * ( YP(1)*( X1INT-Y(5) ) - LDLEAK*( X1LEAD-Y(5))
     &        - LGLEAK*( X1LAG-Y(5) ) )
      BSTAR = 1./(MASS*Y(12)) * ( MASLDC*( X1LEAD-Y(5) ) + MASLGC*
     &        (X1LAG-Y(5) ) )
      CSTAR = (1./Y(12)) * ( MASLDC + MASLGC )
      DSTAR = YP(1) - YP(2) - LDLEAK - LGLEAK
      ESTAR = YP(1)*HIMM - YP(2)*H - LDLEAK*HLEAD - LGLEAK*HLAG
      FSTAR = (1./Y(12)) * ( MASLDC*HLEAD + MASLGC*HLAG )
      ZSTAR = 1./( DRHODP/RHO + BSTAR*MW*(1./MW1 - 1./MW2) +
     &        CSTAR/MASS)
      WSTAR = VOLUME - FSTAR + H*CSTAR + MASS*BSTAR*(H1-H2) - CSUBT*MASS
      A1STAR = ASTAR * MW * (1./MW1 - 1./MW2)
C
C
      YP(5)  = ASTAR - BSTAR
      YP(11) = ( 1./(MASS*CSUBP + DRHODT*WSTAR*ZSTAR/RHO) ) *
     &         ( WSTAR*ZSTAR* ( A1STAR - DVDT/VOLUME + DSTAR/MASS ) -
     &         MASS*ASTAR*(H1-H2) - H*DSTAR + ESTAR - THTRAN )
      YP(12) = ZSTAR * ( A1STAR - DVDT/VOLUME - YP(11)*DRHODT/RHO +
     &         DSTAR/MASS )
C
C
      Y(12) = Y(12) / 1.01325E+6
      YP(12) = YP(12)/1.01325E+6
C
      MDOTDC = YP(12) * MASLDC/Y(12)
      MDOTGC = YP(12) * MASLGC/Y(12)
C
C         CHECK THAT THE ASSUMPTIONS MADE ABOUT THE NET FLOW
C         DIRECTIONS AT EACH APEX ARE CORRECT.  IF EITHER OF THE
C         FLOWS HAVE REVERSED THEN FLOW COMPOSITION AND ENTHALPY
C         MUST BE REASSIGNED.
C
      MDOTLD = MDOTDC + LDLEAK
      ICHECD = -1
      IF (MDOTLD .GT. 0.0 ) ICHECD = +1
      MDOTLG = MDOTGC + LGLEAK
      ICHECG = -1
      IF (MDOTLG .GT. 0.0 ) ICHECG = +1
      IF ( ICHECD .NE. LDFLAG .OR. ICHECG .NE. LGFLAG ) GO TO 40
C
```

```
C        THE FLOW ASSUMPTIONS HAVE BEEN CONFIRMED AS CORRECT.  THE
C        RATE OF CHANGE IN CREVICE COMPOSITIONS CAN NOW BE EVALUATED.
C
      IF (MASLDC .LE. 0.0 ) GO TO 93
           YP(21) = (X1LEAD - X1LDC) * (MDOTLD/MASLDC)
           IF ( MDOTLD .LE. 0.0 ) YP(21) = 0.0
           GO TO 94
C
   93 YP(21) = 0.0
   94 IF ( Y(21) .LE. 0.0  .AND. MASLDC .GT. 0.0 )   Y(21) = X1LDC
      IF ( Y(21) .LE. 0.0  .AND. CREVOL .EQ. 0.0 )   Y(21) = X1LDC
C
C
C
      IF (MASLGC .LE. 0.0) GO TO 95
           YP(22) = (X1LAG - X1LAGC) * (MDOTLG/MASLGC)
           IF (MDOTLG .LE. 0.0) YP(22) = 0.0
           GO TO 96
C
   95 YP(22) = 0.0
   96 IF ( Y(22) .LE. 0.0  .AND.  MASLGC .GT. 0.0 )   Y(22) = X1LAGC
      IF ( Y(22) .LE. 0.0  .AND.  CREVOL .EQ. 0.0 )   Y(22) = X1LAGC
C
C
      YP(23) = YP(1) - YP(2) - MDOTLD - MDOTLG
      YP(18) = LDLEAK
      YP(19) = LGLEAK
      YP(24) = MDOTLD
      YP(25) = MDOTLG
      YP(20) = MDOTLD*HLEAD + MDOTLG*HLAG
C
C
C         CALCULATE THE FUEL ENERGY THAT ENTERS THE EXHAUST CHAMBER
C
C
      IF ( T .LT. TEPO  .OR.  .NOT. FIRE ) GO TO 500
      IF ( MDOTLD .LE. 0.0 ) YP(3) = MDOTLD * X1LDC
      IF ( MDOTLG .LE. 0.0 ) YP(3) = YP(3) + MDOTLG * X1LAGC
      YP(3) = ABS( YP(3) * (H1 - H2))
C
      GO TO 500
C#######################################################################
C
C         THE GOVERNING EQUATIONS DURING THE COMBUSTION PROCESS
C         REQUIRE DIFFERENT TREATMENT AND ARE SOLVED IN THE
C         FOLLOWING SECTION.
C
C#######################################################################
C
C         FIND THE THERMODYNAMIC PROPERTIES OF THE BURNED AND THE
C         UNBURNED GASSES IN THE CREVICES
C
C
  100 CALL THERMO (T,TCR,Y(12),1.-Y(5),HUC,XXA,XXB,XXC,XXD,XXE,XXF,XXG,
     &             XXH,XXI,XXJ,XXK)
```

```
      CALL THERMO (T,TCR,Y(12),1.0,HBC,XXA,XXB,XXC,XXD,XXE,XXF,XXG,XXH,
     &             XXI,XXJ,XXK)
C
C
C
C         ASSUME THAT NET FLOWS HAVE THE SAME SENSE AS THE LAST
C         TIME STEP.
C
C           LEAD AND LAG FLOWS ARE FROM THE CHAMBER TO THE CREVICES
C
      IF (MDOTLD .LE. 0.0  .OR.  MDOTLG .LE. 0.0) GO TO 210
              ALPHLD = 1.- Y(4)
              ALPHLG = 1.- Y(4)
              HLEADU = HU
              HLEADB = HB
              HLAGU  = HU
              HLAGB  = HB
              GO TO 150
C
C           LEAD FLOW IS FROM THE CREVICE TO THE CHAMBER
C           LAG FLOW IS FROM THE CHAMBER TO THE CREVICE
C
  210 IF ( MDOTLG .LE. 0.0 ) GO TO 220
              ALPHLG = 1.- Y(4)
              ALPHLD =  Y(21)/Y(5)
              HLEADU = HU
              HLEADB = HB
              HLAGU  = HUC
              HLAGB  = HBC
              GO TO 150
C
C           LEAD AND LAG FLOWS ARE FROM THE CREVICE TO THE CHAMBER.
C
C
  220         ALPHLD = Y(21)/Y(5)
              ALPHLG = Y(22)/Y(5)
              HLEADU = HUC
              HLEADB = HBC
              HLAGU  = HUC
              HLAGB  = HBC
C
C
C          CALCULATE THE NECESSARY QUANTITIES TO SOLVE THE GOVERNING
C          EQUATIONS.
C
  150 Y(12) = Y(12) * 1.01325E+6
      MASSB = Y(4) * MASS
      MASSU = MASS - MASSB
      VOLB  = VOLUME - Y(14)
C
C
C              THE MASS BURNING RATE IS NOT EXACTLY YP(4)*MASS BUT
C              IS ALSO AFFECTED BY THE OTHER MASS FLOW RATES.
C
      MDOTCB = YP(4)*MASS + Y(4)*(-MDOTLD-MDOTLG) + (1.- ALPHLD)*
```

```
     &          MDOTLD + (1. - ALPHLG)*MDOTLG
C
C
C
      CSTARU = ( ALPHLD*MASLDC + ALPHLG*MASLGC ) * (1./Y(12))
      CSTARB = ((1.- ALPHLD)*MASLDC + (1.- ALPHLG)*MASLGC) * (1./Y(12))
      DSTARU = -ALPHLD*LDLEAK - ALPHLG*LGLEAK - MDOTCB
      DSTARB = (ALPHLD-1.)*LDLEAK + (ALPHLG-1.)*LGLEAK + MDOTCB
      ESTARU = -ALPHLD*LDLEAK*HLEADU - ALPHLG* LGLEAK*HLAGU -
     &         MDOTCB*HU
      ESTARB = (ALPHLD-1.)*LDLEAK*HLEADB + (ALPHLG-1.)*LGLEAK*HLAGB +
     &         MDOTCB*HU
      FSTARU = (1./Y(12)) * (ALPHLD*MASLDC*HLEADU + ALPHLG*MASLGC*HLAGU)
      FSTARB = (1./Y(12)) * ((1.-ALPHLD)*MASLDC*HLEADB + (1.-ALPHLG)*
     &         MASLGC*HLAGB)
      ZSTARU = 1./( DRODPU/RHOU + CSTARU/MASSU )
      ZSTARB = 1./( DRODPB/RHOB + CSTARB/MASSB )
      WSTARU = Y(14) - FSTARU + HU*CSTARU - CSUBTU*MASSU
C
C           IF THE UNBURNED GAS TEMPERATURE DROPS BELOW THE CREVICE
C           TEMPERATURE WSTARU CAN GO NEGATIVE AND CONSEQUENTLY CAN
C           TAKE ON A VALUE OF ZERO.  THIS LEADS TO AN INFINITE
C           VALUE OF YP(12).
C
          IF (WSTARU .LE. Y(14)) WSTARU = Y(14) - CSUBTU*MASSU
C
      WSTARB = VOLB - FSTARB + HB*CSTARB - CSUBTB*MASSB
      GSTARU = 1.- (WSTARU*ZSTARU*DRODTU)/(RHOU*MASSU*CSUBPU +
     &         WSTARU*ZSTARU*DRODTU)
      GSTARB = 1. - (WSTARB*ZSTARB*DRODTB)/(RHOB*MASSB*CSUBPB +
     &         WSTARB*ZSTARB*DRODTB)
C
C         CALCULATE THE TIME RATE OF CHANGE IN CHAMBER PRESSURE
C
      YP(12) = ( DSTARU/MASSU + (1.-1./GSTARU)*(ESTARU - HU*DSTARU -
     &          UHTRAN)/(ZSTARU*WSTARU) + (VOLB/Y(14))*(DSTARB/MASSB -
     &          DVDT/VOLB + (1.-1./GSTARB)*(ESTARB - HB*DSTARB -BHTRAN)/
     &          (ZSTARB*WSTARB)) ) / ( (VOLB/( Y(14)*GSTARB)) *
     &          (CSTARB/MASSB + DRODPB/RHOB) + (1./GSTARU)*
     &          (CSTARU/MASSU + DRODPU/RHOU) )
C
C
C
      MDOTDC = YP(12)*MASLDC/Y(12)
      MDOTGC = YP(12)*MASLGC/Y(12)
      MDOTLD = MDOTDC + LDLEAK
      MDOTLG = MDOTGC + LGLEAK
C
C
C
      YP(13) = ( YP(12) * (Y(14) - MASSU*CSUBTU + CSTARU*HU - FSTARU)
     &          - HU*DSTARU + ESTARU - UHTRAN )/ (MASSU*CSUBPU)
C
      YP(15) = ( YP(12) * (VOLB - MASSB*CSUBTB + CSTARB*HB - FSTARB)
     &          - HB*DSTARB + ESTARB - BHTRAN )/ (MASSB*CSUBPB)
```

```
C
      YP(14) = Y(14) * ((DSTARU-CSTARU*YP(12))/MASSU - (DRODTU*YP(13)+
     &         DRODPU*YP(12))/RHOU)
C
C
      IF ( ITERAS .EQ. 1 ) GO TO 400
C
C
      IF (MASLDC .LE. 0.0) GO TO 103
           YP(21) = (MDOTLD/MASLDC) * (ALPHLD*Y(5) - Y(21))
           IF (MDOTLD .LE. 0.0) YP(21) = 0.0
           GO TO 104
  103 YP(21) = 0.0
C
C
  104 IF (MASLGC .LE. 0.0) GO TO 105
           YP(22) = (MDOTLG/MASLGC) * (ALPHLG*Y(5) - Y(21))
           IF (MDOTLG .LE. 0.0) YP(22) = 0.0
           GO TO 400
  105 YP(22) = 0.0
C
C
  400 YP(23) = -MDOTLD - MDOTLG
      YP(18) = LDLEAK
      YP(19) = LGLEAK
      YP(20) = MDOTLD*ALPHLD*HLEADU + MDOTLG*ALPHLG*HLAGU  +
     &         MDOTLD*(1.- ALPHLD)*HLEADB + MDOTLG*(1.- ALPHLG)*HLAGB
      YP(24) = MDOTLD
      YP(25) = MDOTLG
      YP(12) = YP(12)/1.01325E+6
      Y(12)  = Y(12)/1.01325E+6
C
C
  500 CRMASD = MASLDC
      CRMASG = MASLGC
      ZMASS  = MASS
      ZLEAKD = Y(18)
      ZLEAKG = Y(19)
      ZCRCOD = Y(21)
      ZCRCOG = Y(22)
      IF (MASLDC .LE. 0.0) ZCRCOD = 0.0
      IF (MASLGC .LE. 0.0) ZCRCOG = 0.0
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE TABLE
C
C     PURPOSE
C        TO INTERPOLATE BETWEEN THE STORED VALUES OF CHAMBER PRESSURE
C        AND CREVICE COMPOSITION AND THEN TO RETURN THE INTERPOLATED
C        VALUES TO SUBROUTINE CREVICE.
C
C     USAGE
C        CALL TABLE (T,Y,PLEAD,PLAG,X1LDC,X1LAGC)
```

```
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        PLEAD      NO     YES      LEADING CHAMBER PRESSURE (ATM)
C        PLAG       NO     YES      TRAILING CHAMBER PRESSURE (ATM)
C        X1LDC      NO     YES      LEAD CREVICE COMPOSITION ( )
C        X1LAGC     NO     YES      LAG CREVICE COMPOSITION ( )
C
C
C     REMARKS
C        NONE
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C
      SUBROUTINE TABLE (T,Y,PLEAD,PLAG,X1LDC,X1LAGC)
C
C
      LOGICAL FIREFL, FIRE
      DIMENSION Y(25)
      COMMON/TABLES/ PRES(1080), X1LD(1080), X1LG(1080)
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/CREVIN/AREALK,CREVOL,TCREV,X1LDIN,X1LGIN
C
C         IN THIS SUBROUTINE THE CYCLE BEGINS AT TIPO = 0.0 AND
C         ENDS AT TIPO = 1080.0 DEG. THE LEADING CHAMBER IS 360 CRANK
C         ANGLES AHEAD (TLEAD),AND THE TRAILING CHAMBER IS
C         360 DEGREES BEHIND (TLAG).
C
C
      ABST = T + ABS(TIPO)
      TLEAD = ABST + 360.
      IF ( TLEAD .GE. 1080.) TLEAD = TLEAD - 1080.
      TLAG = ABST - 360.
      IF ( TLAG .LT. 0.0 ) TLAG = TLAG + 1080.
      ITLEAD = INT(TLEAD)
      ITLAG = INT(TLAG)
C
      PLEAD = PRES(ITLEAD) + (TLEAD - ITLEAD)*( PRES(ITLEAD+1) -
     &        PRES(ITLEAD) )
      PLAG = PRES(ITLAG) + (TLAG - ITLAG)*( PRES(ITLAG+1) -
     &       PRES(ITLAG) )
      IABST = INT(ABST)
      X1LDC = X1LD(IABST) + (ABST - IABST)*( X1LD(IABST+1) -
     &        X1LD(IABST) )
      X1LAGC = X1LG(IABST) + (ABST - IABST)*( X1LG(IABST+1) -
     &        X1LG(IABST) )
```

```
C
C            DURING THE FIRST ITERATION NO  VALUES OF CREVICE
C            COMPOSITION ARE STORED TO INTERPOLATE BETWEEN.SO
C            FOR THE SECOND ITERATION ASSUME THE CREVICE GAS HAS THE
C            ARBITRARY COMPOSITION OF 0.75 .
C
      IF ( ITERAS .NE. 2 ) GO TO 200
C
      IF ( CREVOL .LE. 0.0  .AND.  AREALK .GT. 0.0 ) GO TO 200
C
              X1LDC = 0.75
              X1LAGC = 0.75
C
C
C        FOR A MOTORING RUN THE INITIAL VALUE OF CREVICE COMPOSITION
C        SHOULD BE 1.0
C
      IF ( FIRE )  GO TO 250
              X1LDC  = 1.0
              X1LAGC = 1.0
  250 CONTINUE
C
C       FOR A FIRING CASE WITH ZERO CREVICE VOLUME THE LEAKAGE
C       COMPOSITION IS ASSUMED TO BE RELATED TO THE MASS FRACTION
C       BURNED.
C
      IF ( FIREFL ) GO TO 300
           GO TO 200
C
  300         X1LDC = 1.- Y(4)
              X1LAGC = 1.- Y(4)
  200 RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE BUILD
C
C     PURPOSE
C        TO STORE THE CHAMBER PRESSURE AND CREVICE COMPOSITIONS
C        FROM ONE ITERATION TO THE NEXT.
C
C     USAGE
C        CALL BUILD  (DT, DY)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C
C     REMARKS
C        IT IS ASSUMED IN BUILD THAT THE STEP SIZE FOR ODERT IS
C        ONE (1.0) DEGREE.  IF THE MAIN PROGRAM IS CHANGED SO THAT
C        THE STEP SIZE IS ALTERED THEN SUBROUTINE BUILD MUST ALSO
C        BE ALTERED.
```

```
C
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C
      SUBROUTINE BUILD (DT,DY)
C
      LOGICAL FIREFL, FIRE
      REAL*8 DT,DY(25)
      REAL MASLDC, MASLGC
      DIMENSION Y(25)
      COMMON/TABLES/ PRES(1080), X1LD(1080), X1LG(1080)
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/CREV2/ MASLDC, MASLGC
      COMMON/BURN/ SPBURN, FIRE, FIREFL
      COMMON/CREVIN/AREALK,CREVOL,TCREV,X1LDIN,X1LGIN
C
      T = DT
      DO 10 I = 1,25
            Y(I) = DY(I)
   10 CONTINUE
      ABST = T + ABS(TIPO)
      IABST = INT(ABST)
      IF ( IABST .NE. ABST ) GO TO 30
      PRES(IABST) = Y(12)
C
C        IF THE CREVICE BELONGS TO THE CHAMBER THEN ITS MASS, AND
C        COMPOSITION ARE KNOWN. THIS CREVICE ALSO CORRESPONDS TO THE
C        ADJACENT CHAMBER AT TLAG OR TLEAD.
C
      ITLEAD = INT( ABST+360.)
      IF (ITLEAD .GE. 1080.)ITLEAD = ITLEAD-1080
      ITLAG = INT( ABST-360.)
      IF (ITLAG .LT. 0.0) ITLAG = ITLAG+1080
C
C       CHECK TO SEE IF LEAD CREVICE BELONGS TO CHAMBER
C
      IF (MASLDC .LE. 0.0  .OR. Y(21) .LE. 0.1 ) GO TO 20
      X1LD(IABST) = Y(21)
      X1LG(ITLEAD) = Y(21)
C
C        CHECK TO SEE IF LAG CREVICE BELONGS TO CHAMBER
C
   20 IF (MASLGC .LE. 0.0  .OR. Y(22) .LE. 0.1 ) GO TO 30
      X1LG(IABST) = Y(22)
      X1LD(ITLAG) = Y(22)
C
C         IF A LEAKAGE AREA IS SPECIFIED BUT NOT A CREVICE
C         VOLUME THEN STORE THE CHAMBER COMPOSITION.  DURING
C         COMBUSTION THIS IS DEFINED TO BE THE MASS FRACTION
C         UNBURNED.
C
```

```
   30 IF (CREVOL .GT. 0.0  .OR.  AREALK .LE. 0.0) GO TO 40
      COMP = Y(5)
      IF (FIRE    .AND.  T .GT. TSPARK )  COMP = 1. - Y(4)
      X1LD(IABST)  = COMP
      X1LG(ITLEAD) = COMP
      X1LG(IABST)  = COMP
      X1LD(ITLAG)  = COMP
C
C
   40 RETURN
      END
C-----------------------------------------------------------------------
C
C     SUBROUTINE IPACD
C
C     PURPOSE
C        CALCULATES AREA AND DISCHARGE COEFFICIENT
C        OF THE INTAKE PORTS.
C
C     USAGE
C        CALL IPACD  (T, AREA, CD)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        T          YES    NO       TIME (DEG)
C        AREA       NO     YES      EFFECTIVE AREA OF INTAKE
C        ----       --     ---      PORT (CM**2)
C        CD         NO     YES      DISCHARGE COEFFICIENT
C
C
C     REMARKS
C
C        THIPO =   NUMBER OF CRANK ANGLES REQUIRED TO FULLY OPEN
C                  OR CLOSE THE PORT.
C
C        - IT IS ASSUMED THAT THE PORTS OPEN AND CLOSE LINEARLY AND
C          THE DISCHARGE COEFFICIENT IS CONSTANT
C
C        - SEE WARNING ABOUT PORT AREA CHANGES 1/21/83
C
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE IPACD (T, AREA, CD)
      REAL IPA
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
```

```
      COMMON/PORTS/ IPA, EPA
CJ
CJ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
CJ        THIS ROUTINE HAS BEEN CHANGED SUBSTANTIALLY IN ORDER TO MODEL
CJ        THE ENLARGED INTAKE PORT INSTALLED IN THE NASA TEST ENGINE.
CJ        ANY CHANGES WHATSOEVER TO THE PORT CONFIGURATION MUST BE
CJ        REFLECTED IN THIS ROUTINE
CJ
CJ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      IF ( T .GT. (TIPO + 120.0)) GO TO 20
      AREA = IPA *(T - TIPO)/120.0
      GO TO 40
   20 IF ( T .LT. (TIPC - 180.0)) GO TO 30
      IF ( T .LT. -240.0)   AREA  =   (-12.2/120.0)*(T+360.) + 13.8
      IF (T .GT. -240.) AREA = (1.600073*EXP(-10./60.*(T+240.))-.000073)
      GO TO 40
   30 AREA = IPA
   40 CD= 0.75
C
C     IN ORDER TO AVOID ANY DIVISION BY ZERO THE PORT AREA SHALL BE
C     ASSIGNED AN ARBITRARILY SMALL VALUE
C
      IF (AREA .EQ. 0.0) AREA=1.E-6
C
      RETURN
      END
C______________________________________________________________________
C
C     SUBROUTINE EPACD
C
C     PURPOSE
C        CALCULATES AREA AND DISCHARGE COEFFICIENT
C        OF EXHAUST VALVE
C
C     USAGE
C        CALL EPACD  (T, AREA, CD)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        T          YES    NO       TIME (DEG)
C        AREA       NO     YES      EFFECTIVE AREA OF EXHAUST
C        ----       --     ---      PORT (CM**2)
C        CD         NO     YES      DISCHARGE COEFFICIENT
C
C     REMARKS
C
C        THEVO   = NUMBER OF CRANK ANGLES REQUIRED TO OPEN OR CLOSE
C                  THE EXHAUST PORT
C
C        -  IT IS ASSUMED THAT THE PORT OPENS AND CLOSES LINEARLY AND
C           THE DISCHARGE COEFFICIENT IS CONSTANT.
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
```

```
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE EPACD (T, AREA, CD)
      REAL IPA
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/PORTS/ IPA, EPA
      IF (T .GT. (TEPO + THEPO)) GO TO 20
      AREA = EPA *(T - TEPO)/THEPO
      GO TO 40
   20 IF (T .LT. (TEPC - THEPO)) GO TO 30
      AREA = EPA *(TEPC - T)/THEPO
      GO TO 40
   30 AREA = EPA
   40 CD = 0.65
C
C     IN ORDER TO AVOID ANY DIVISION BY ZERO THE PORT AREA SHALL BE
C     ASSIGNED AN ARBITRARILY SMALL VALUE
C
      IF(AREA .EQ. 0.0) AREA = 1.E-6
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C     SUBROUTINE HEATTX
C
C     PURPOSE
C        CALCULATES THE RATE OF HEAT TRANSFER FROM THE CHAMBER
C        THROUGH THE WALLS OF THE ROTOR, SIDE PLATES, AND HOUSING.
C
C     USAGE
C        CALL HEATTX (T,Y,YP,THTRAN,UHTRAN,BHTRAN)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        DT         YES    NO       TIME (DEG)
C        DY(11)     YES    NO       CYLINDER TEMPERATURE (K)
C        DY(12)     YES    NO       CYLINDER PRESSURE (ATM)
C        DY(13)     YES    NO       UNBURNED MIXTURE TEMPERATURE
C        ------     ---    --       DURING COMBUSTION  (K)
C        DY(14)     YES    NO       VOLUME OF UNBURNED MIXTURE
C        ------     ---    --       DURING COMBUSTION (CM**3)
C        DY(15)     YES    NO       BURNED PRODUCTS TEMPERATURE
C        ------     ---    --       DURING COMBUSTION  (K)
C -----------------------------------------------------------------------
C        DYP(8)     NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      ROTOR WALL
C        DYP(9)     NO     YES      RATE OF HEAT TRANSFER THROUGH
```

```
C        -------    --     ---      SIDE PLATES
C        DYP(10)    NO     YES      RATE OF HEAT TRANSFER THROUGH
C        -------    --     ---      HOUSING
C
C     REMARKS
C        SEE THE PROCESS SUBROUTINE FOR THE UNITS OF THE INTEGRATION
C        VARIABLES.
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        UTRANS     BTRANS
C
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
C     EDITED BY  T. J. NORMAN
C
C
C
C
      SUBROUTINE HEATTX (T,Y,YP,THTRAN,UHTRAN,BHTRAN)
C
C
      LOGICAL FIRE, FIREFL
      REAL MW, MASS, KINVIS
      DIMENSION Y(25),YP(25)
C
      COMMON/HEATXG/AROTOR,ASIDE,AHOUS,ROTVEL,DCHAR
      COMMON/BURN/SPBURN,FIRE,FIREFL
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM
      COMMON/HTTXIN/CONHT,EXPHT,CON1,CON2
      COMMON/RHMAS/RHO, MASS, VOLUME, H, GAMMA, GAMAB
      COMMON/TEMPS/TROTOR,TSIDE,THOUS
      COMMON/CREVCB/HU,HB,DRODPU,DRODPB,RHOU,RHOB,DRODTU,DRODTB,
     &              CSUBPU,CSUBPB,CSUBTU,CSUBTB
      COMMON/HEATS/VELHTX, HTRCOE, HTPARO, HTPASI, HTPAHO, HTXRO,
     &             HTXSI, HTXHO,  QFRRO, QFRSI, QFRHO
      COMMON/UBHEAT/ UHTRCO, BHTRCO
C
C         CALCULATE THE CONSTANTS OF THE POLYTROPIC COMPRESSION
C         ( PV**CONST2 = CONST1)
C
      IF ( T .NE. TIPC ) GO TO 5
           PIPC = Y(12)
           VIPC = VOLUME
    5 IF ( T .NE. INT(TSPARK) ) GO TO 7
           PSPARK = Y(12)
           VSPARK = VOLUME
           TEMPSP = Y(11)
C
      CONST2 = (LOG(PSPARK/PIPC)) / (LOG(VIPC/VSPARK))
      CONST1 = PIPC*VIPC**CONST2
    7 CONTINUE
```

```
C
C
      IF ( T .GE. TSPARK .AND. FIREFL ) GO TO 9
           VELHTX = CON1*ROTVEL
           GO TO 11
C
C
   9  PMOTOR = CONST1/(VOLUME**CONST2)
      VELHTX = CON1*ROTVEL + CON2*( Y(12) - PMOTOR ) * (VOLUME*TEMPSP/
     &         (PSPARK*VSPARK))
C
C
   11 IF ( FIREFL ) GO TO 30
      IF ( FIRE   .AND.  T .GE. TEPO ) GO TO 10
C
C         INTAKE, COMPRESS, AND EXPANSION-EXHAUST (MOTORING CASE)
C
         CALL UTRANS (Y(11),DYNVIS,THRCON)
         GO TO 20
C
C         EXHAUST (FIRING CASE)
C
   10 CALL BTRANS (Y(11),GAMAB,CSUBPB,DYNVIS,THRCON)
C
   20 KINVIS = DYNVIS/RHO
      HTRCOE = (CONHT*((VELHTX*DCHAR)/KINVIS)**EXPHT) * (THRCON/DCHAR)
C
C         CALCULATE THE HEAT TRANSFER RATES PER UNIT AREA
C
              HTPARO = HTRCOE * (Y(11) - TROTOR)
              HTPASI = HTRCOE * (Y(11) - TSIDE)
              HTPAHO = HTRCOE * (Y(11) - THOUS)
C
              HTXRO = HTPARO * AROTOR
              HTXSI = HTPASI * ASIDE
              HTXHO = HTPAHO * AHOUS
C
C
      YP(8)  =  HTXRO
      YP(9)  =  HTXSI
      YP(10) =  HTXHO
C
C     FIND THE TOTAL HEAT TRANSFER FROM THE CHAMBER
C
      THTRAN = HTXRO + HTXSI + HTXHO
C
              IF (ABS(THTRAN) .LE. .0002) GO TO 40
              QFRRO = 100.* HTXRO/THTRAN
              QFRSI = 100.* HTXSI/THTRAN
              QFRHO = 100.* HTXHO/THTRAN
C
      GO TO 40
C
C
C     CALCULATE THE HEAT TRANSFER RATE FROM BOTH ZONES DURING
```

```
C     COMBUSTION.
C
CJ
   30 VOLU = Y(14)
      VOLB = VOLUME - VOLU
C
C       CALCULATE THE BURNED AND UNBURNED WALL AREAS
C
      UAROT = AROTOR/( (VOLB/VOLU)**(2./3.) + 1.)
      UARSI = ASIDE/( (VOLB/VOLU)**(2./3.) + 1.)
      UARHO = AHOUS/( (VOLB/VOLU)**(2./3.) + 1.)
C
      BAROT = AROTOR - UAROT
      BARSI = ASIDE - UARSI
      BARHO = AHOUS - UARHO
C
      CALL UTRANS(Y(13),UDYVIS,UTHCON)
      CALL BTRANS(Y(15),GAMAB,CSUBPB,BDYVIS,BTHCON)
      UKNVIS = UDYVIS/RHOU
      BKNVIS = BDYVIS/RHOB
C
      UHTRCO = (CONHT*((VELHTX*DCHAR)/UKNVIS)**EXPHT) * (UTHCON/DCHAR)
      BHTRCO = (CONHT*((VELHTX*DCHAR)/BKNVIS)**EXPHT) * (BTHCON/DCHAR)
C
C        CALCULATE THE BURNED AND UNBURNED HEAT TRANSFER RATES PER
C        UNIT AREA
C
             UHPARO = UHTRCO * (Y(13) - TROTOR)
             UHPASI = UHTRCO * (Y(13) - TSIDE)
             UHPAHO = UHTRCO * (Y(13) - THOUS)
C
             BHPARO = BHTRCO * (Y(15) - TROTOR)
             BHPASI = BHTRCO * (Y(15) - TSIDE)
             BHPAHO = BHTRCO * (Y(15) - THOUS)
C
             UHTRRO = UHPARO * UAROT
             UHTRSI = UHPASI * UARSI
             UHTRHO = UHPAHO * UARHO
C
             BHTRRO = BHPARO * BAROT
             BHTRSI = BHPASI * BARSI
             BHTRHO = BHPAHO * BARHO
C
C
      YP(8)  = UHTRRO + BHTRRO
      YP(9)  = UHTRSI + BHTRSI
      YP(10) = UHTRHO + BHTRHO
C
C     FIND THE TOTAL HEAT TRANSFER RATE FROM EACH ZONE
C
      UHTRAN = UHTRRO + UHTRSI + UHTRHO
      BHTRAN = BHTRRO + BHTRSI + BHTRHO
C
C     THE TOTAL HEAT TRANSFER RATE FROM THE CHAMBER IS:
C
```

```
      THTRAN = UHTRAN + BHTRAN
C
   40 CONTINUE
C
CJ    QTOTAL = THTRAN
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE MFLRT
C
C     PURPOSE
C        CALCULATES MASS FLOW RATE THROUGH AN ORIFICE.
C
C     USAGE
C        CALL MFLRT  (CD, AREA, PO, MW, TO, PS, GAMMA, FLRT)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        CD         YES    NO       DISCHARGE COEFFICIENT
C        AREA       YES    NO       AREA OF RESTRICTION (CM**2)
C        PO         YES    NO       UPSTREAM PRESSURE (ATM)
C        PS         YES    NO       DOWNSTREAM PRESSURE (ATM)
C        MW         YES    NO       MOLECULAR WEIGHT (G/MOLE)
C        TO         YES    NO       UPSTREAM TEMPERATURE (K)
C        GAMMA      YES    NO       RATIO OF SPECIFIC HEATS, CP/CV
C        FLRT       NO     YES      MASS FLOW RATE (G/S)
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAM REQUIRED
C        NONE
C
C     METHOD
C        FLOW THROUGH THE ORIFICE IS TREATED AS ONE-DIMENSIONAL,
C        QUASI-STEADY, AND ISENTROPIC (MODIFIED BY A DISCHARGE
C        COEFFICIENT)
C
C     WRITTEN BY S. H. MANSOURI AND K. RADHAKRISHNAN
C     EDITED BY S. H. MANSOURI AND S. G. POULOS
C
      SUBROUTINE MFLRT (CD, AREA, PO, MW, TO, PS, GAMMA, FLRT)
C
      REAL MW
C
      FLRT  = 0.0
      IF (PO .EQ. PS) GO TO 20
      GI    = 1.0/GAMMA
      SUM   = GAMMA * MW/TO
      CONST = 111.12272 * CD * AREA * PO * SQRT(SUM)
C
      RATIO = PS/PO
```

```
      CRIT  = ( 2./(GAMMA + 1.) )**( GAMMA/(GAMMA - 1.) )
C
C        CHECK IF FLOW IS CHOKED
C
      IF (RATIO .LT. CRIT) GO TO 10
C
C        SUBSONIC FLOW
C
      SUN  = 2./(GAMMA - 1.) * ( RATIO**(GI + GI) - RATIO**(GI + 1.) )
      FLRT = CONST * SQRT(SUN)
      GO TO 20
C
C        CHOKED FLOW
C
   10 FLRT = CONST * CRIT**( 0.5 * (1.0 + GI) )
C
   20 RETURN
      END
C
C
C
C     SUBROUTINE CSAVDV
C
C     PURPOSE
C        CALCULATES SURFACE AREA, VOLUME, AND TIME RATE OF CHANGE OF
C        VOLUME OF COMBUSTION CHAMBER.
C
C     USAGE
C        CALL CSAVDV (T, VOLUME, DVDT)
C
C     DESCRIPTION OF PARAMETERS
C
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        T          YES    NO       TIME (DEG)
C        ASIDE      NO     YES      SIDE SURFACE AREA (CM**2)
C        AROTOR     NO     YES      ROTOR SURFACE AREA (CM**2)
C        AHOUS      NO     YES      HOUSING SURFACE AREA (CM**2)
C        VOLUME     NO     YES      CHAMBER VOLUME (CM**3)
C        DVDT       NO     YES      TIME RATE OF CHANGE OF VOLUME OF
C        -----      --     ---      CHAMBER (CM**3/SEC)
C        ROTVEL     NO     YES      AVERAGE ROTOR SPEED (CM/SEC)
C        DCHAR      NO     YES      CHARACTERISTIC DIMENSION FOR
C        ------     --     ---      HEAT TRANSFER CALCULATIONS (CM)
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        SEE REPORT
C
C     WRITTEN BY T. J. NORMAN
```

```
C     EDITED BY T. J. NORMAN
C
      SUBROUTINE CSAVDV (T, VOLUME, DVDT)
      COMMON/EPARAM/ ECCEN, ROTRAD, DEPTH, VFLANK
      COMMON/DTDTH/ ESPDI, RPM
      COMMON/HEATXG/AROTOR, ASIDE, AHOUS, ROTVEL, DCHAR
C
      PI=3.14159265358979323
      ROOT3 = SQRT(3.)
C
C     THETA IS THE CRANK ANGLE (OFFSET BY 90 DEGREES) IN RADIANS
C     ALPHA IS THE "FOLLOWING" APEX SEAL ANGLE IN RADIANS
C
      THETA =(T+90.) * PI/180.
      ALPHA = THETA/3.
      BETA  = 2.* ALPHA
C
C          ALEAN IS THE MAXIMUM ANGLE OF INCLINATION OF THE APEX SEAL
C          FROM THE NORMAL TO THE HOUSING.
C
      ALEAN = ASIN(3.* ECCEN/ROTRAD )
      FT    = (ECCEN*ECCEN + ROTRAD*ROTRAD/3.) * PI -
     &        ROOT3 * ECCEN * ROTRAD * SIN( BETA+PI/6.)
      DFTDTH = -2*ROOT3/3.* ECCEN * ROTRAD * COS( BETA + PI/6.)
C
      FR     = PI/3.* (ROTRAD*ROTRAD + 2.*ECCEN*ECCEN)
     &         -2. * ECCEN * ROTRAD * COS(ALEAN)
     &         -ALEAN * (ROTRAD*ROTRAD*2./9. + 4.*ECCEN*ECCEN )
      DFRDTH = 0.0
C
      FR1     = ECCEN * ROTRAD * SIN(BETA)/2.
      DFR1DT  = (1/3.) * ECCEN * ROTRAD * COS( BETA )
C
      FR2     = ECCEN * ROTRAD * SIN(BETA+PI/3.)/2.
      DFR2DT  = (1/3.) * ECCEN * ROTRAD * COS( BETA + PI/3.)
C
      ASIDE   = FT-FR-FR1-FR2
      DASDTH  = DFTDTH - DFRDTH - DFR1DT  - DFR2DT
C
      VOLUME  = ASIDE * DEPTH + VFLANK
      DVDTH   = DASDTH * DEPTH
      DVDT    = DVDTH * RPM * PI/30.
C
C             FIND THE SURFACE AREAS OF THE HOUSING, ROTOR AND SIDES.
C
      ASIDE  = 2.* ASIDE
      RPRIME = ROTRAD - ECCEN + 3.*ECCEN*ROTRAD/(ROTRAD - 4.*ECCEN)
      BETA2  = ROOT3*ROTRAD/((6.*ECCEN*ROTRAD)/(ROTRAD-4.*ECCEN) +
     &         ROTRAD + 2.*ECCEN)
      ROTORL = RPRIME * 2.* BETA2
      AROTOR = ROTORL * DEPTH
C
C            AN APPROXIMATION IS USED TO FIND THE TOTAL SURFACE AREA
C            THE CORRECTION FACTOR SHOULD BE CHECKED IF THE ENGINE
C            GEOMETRY DIFFERS GREATLY FROM THE TEST CONDITIONS:
```

```
C            (ROTRAD = 10.5 , ECCEN = 1.5 , DEPTH = 7.0).
C
C
      AREA  = 2.* (VOLUME/ROTORL + VOLUME/DEPTH + ROTORL*DEPTH )
     &        + .151 * VOLUME
      AHOUS = AREA - AROTOR - ASIDE
C
C          DEFINE A CHARACTERISTIC ROTOR VELOCITY AND DIMENSION
C          FOR HEAT TRANSFER PURPOSES.
C
      ROTVEL = RPM*PI*ROTRAD/90.
      DCHAR  = DEPTH
C
C          COMPONENT AREAS ARE PASSED IN THE CALL STATEMENT AND
C          IN THE COMMON STATEMENT HEATXG.
C
      RETURN
      END
C_____________________________________________________________________
      FUNCTION GINT1 (DT, DY, DYP)
      REAL*8 DT,DY(25),DYP(25),GINT1
C
      GINT1 = DYP(1)
C
      RETURN
      END
C_____________________________________________________________________
C
      FUNCTION GINT2 (DT, DY, DYP)
      REAL*8 DT,DY(25),DYP(25),GINT2
C
      GINT2 = DY(1)
C
      RETURN
      END
C_____________________________________________________________________
C
      FUNCTION GCMP (DT, DY, DYP)
      REAL*8 DT,DY(25),DYP(25),GCMP
C
      GCMP =DYP(12)
C
      RETURN
      END
C_____________________________________________________________________
C
      FUNCTION GEXH (DT, DY, DYP)
      REAL*8 DT,DY(25),DYP(25),GEXH
C
      GEXH =10.00
C
      RETURN
      END
C_____________________________________________________________________
      FUNCTION GCMB (DT, DY, DYP)
```

```
      REAL*8 DT,DY(25),DYP(25),GCMB1
      COMMON/CREV/DRHODP,CSUBT,MW,ITERAS,DVDT,CSUBP,DRHODT,HIMM,RESIDL,
     &            RESFIM
C
      GCMB = DYP(24)
      IF (ITERAS .EQ. 1 ) GCMB = 10.0
C
      RETURN
      END
C
C
C     SUBROUTINE HELPHT
C
C     PURPOSE
C        CALLS EITHER 'INTAKE', 'CMPRES', 'CMBSTN', OR 'EXAUST',
C        AFTER EVERY CALL TO 'ODERT'.  'ODERT' MAY OVERSHOOT IT'S
C        STOPPING POINT (TOUT) FOR IMPROVED ACCURACY.  IF THIS
C        OCCURS, 'ODERT' WILL RETURN WITH THE CORRECT VECTOR (DY),
C        BUT THE HEAT TRANSFER DATA CALCULATED BY THE EXTERNAL
C        FUNCTIONS WILL CORRESPOND TO THE LAST OVERSHOT VALUE
C        OF T AND NOT TO TOUT.  IT IS THEN NECESSARY TO CALL
C        THE EXTERNAL FUNCTIONS ONCE AFTER EACH CALL TO ODERT IF
C        THE CORRECT HEAT TRANSFER DATA IS TO BE AVAILABLE FOR
C        PRINTING OUT.  THIS PROCEDURE DOES NOT AFFECT THE NOR-
C        MAL CALCULATION PROCESS PERFORMED BY 'ODERT'.
C
C     USAGE
C        CALL HELPHT (DT, DY, IWHERE)
C
C     DESCRIPTION OF PARAMETERS
C        PARAMETER  INPUT  OUTPUT  DESCRIPTION
C
C        DT         YES    NO      CURRENT TIME (DEG)
C        DY         YES    NO      CURRENT SOLUTION VECTOR
C        IWHERE     YES    NO      1 = INTAKE; 2 = CMPRES;
C        ------     ---    --      3 = CMBSTN; 4 = EXAUST
C        XXX        NO     NO      DUMMY VARIABLE
C
C     REMARKS
C        NONE
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        INTAKE     CMPRES     CMBSTN     EXAUST
C
C     METHOD
C        SEE PURPOSE, ABOVE
C
C     WRITTEN BY S. G. POULOS
C     EDITED BY S. G. POULOS
C
      SUBROUTINE HELPHT (DT, DY, IWHERE)
      REAL*8 DT, DY(25), XXX(25)
C
      IF (IWHERE.EQ.1) CALL INTAKE (DT, DY, XXX)
      IF (IWHERE.EQ.2) CALL CMPRES (DT, DY, XXX)
```

```
      IF (IWHERE.EQ.3) CALL CMBSTN (DT, DY, XXX)
      IF (IWHERE.EQ.4) CALL EXAUST (DT, DY, XXX)
C
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE UTRANS
C
C     PURPOSE
C        CALCULATES DYNAMIC VISCOSITY AND THERMAL CONDUCTIVITY
C        OF UNBURNED MIXTURE
C
C     USAGE
C        CALL UTRANS (TEMP, DYNVIS, THRCON)
C
C     DESCRIPTION OF PARAMETERS
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        TEMP       YES    NO       TEMPERATURE (K)
C        DYNVIS     NO     YES      DYNAMIC VISCOSITY (G/SEC CM)
C        THRCON     NO     YES      THERMAL CONDUCTIVITY (ERG/SEC CM K)
C
C     REMARKS
C        UNBURNED MIXTURE IS ASSUMED TO BE THE SAME AS AIR.
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        SEE MANSOURI PHD THESIS
C
C     WRITTEN BY S. H. MANSOURI
C     EDITED BY S. G. POULOS
C
      SUBROUTINE UTRANS (TEMP, DYNVIS, THRCON)
C
      DYNVIS = 14.58E-6 * (TEMP** 1.5)/(TEMP + 110.4)
      THRCON = 2.6464E+2 * ( TEMP**0.5 )/( 1. + 245.4 *
     &         (10.**(-12./TEMP))/TEMP )
C
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE BTRANS
C
C     PURPOSE
C        CALCULATES DYNAMIC VISCOSITY AND THERMAL CONDUCTIVITY
C        OF BURNED PRODUCTS
C
C     USAGE
C        CALL BTRANS (TEMP, GAMMA, CP, DYNVIS, THRCON)
C
C     DESCRIPTION OF PARAMETERS
```

```
C        PARAMETER  INPUT  OUTPUT   DESCRIPTION
C
C        TEMP       YES    NO       TEMPERATURE (K)
C        CP         YES    NO       HEAT CAPACITY AT CONSTANT PRESSURE
C        --         ---    --       OF BURNED PRODUCTS (ERG/G K)
C        DYNVIS     NO     YES      DYNAMIC VISCOSITY OF
C        ------     --     ---      BURNED PRODUCTS (G/SEC CM)
C        THRCON     NO     YES      THERMAL CONDUCTIVITY OF
C        ------     --     ---      BURNED PRODUCTS (ERG/SEC CM K)
C
C     REMARKS
C        NONE
C
C     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C
C     METHOD
C        SEE S. H. MANSOURI AND J. B. HEYWOOD, " CORRELATIONS FOR THE
C        VISCOSITY AND PRANDTL NUMBER OF HYDROCARBON-AIR COMBUSTION
C        PRODUCTS," COMBUSTION SCIENCE AND TECHNOLOGY, 1980, VOL. 23,
C        PP. 251-256
C
C     WRITTEN BY S. H. MANSOURI
C     EDITED BY S. G. POULOS
C
      SUBROUTINE BTRANS (TEMP, GAMMA, CP, DYNVIS, THRCON)
C
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
C
      DYNVIS = 3.3E-6 * (TEMP**.7)/(1.0 + .027 * PHI)
      PRNDTL = 0.05 + 4.2 * (GAMMA - 1.0) - 6.7 * (GAMMA - 1.0) *
     &         (GAMMA - 1.0)
      THRCON = DYNVIS * CP/PRNDTL
      IF ((PHI .LE. 1.0) .OR. (TEMP .LE. 1500.)) RETURN
      PRNDTL = PRNDTL/(1.0 + 1.5E-8 * PHI * PHI * TEMP * TEMP)
      THRCON = DYNVIS * CP/PRNDTL
C
      RETURN
      END
C
C
C     SUBROUTINE THERMO
C
C     PURPOSE
C        'THERMO' IS CALLED BY THE THE 4 PROCESS ROUTINES AND BY
C        'MAIN' AND RETURNS WITH THE REQUIRED THERMODYNAMIC PROPER-
C        TIES IN EACH CASE.  IT CALLS 'UPROP' AND OR 'BPROP' AS
C        REQUIRED FOR EACH PROCESS, AND THEN CALCULATES FROM THE
C        RETURNED DATA ANY ADDITIONAL PROPERTIES OR COMBINATIONS
C        OF PROPERTIES OF INTEREST.  'THERMO' ALSO CONVERTS ALL
C        VALUES TO UNITS THAT ARE CONSISTENT WITH THOSE USED IN
C        THE REST OF THE PROGRAM.
C
C     USAGE
```

```
C        CALL THERMO ( T, TEMP, P, RESFRK, ENTHLP, CSUBP, CSUBT, RHO,
C              DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
C
C     DESCRIPTION OF PARAMETERS
C        PARAMETER  INPUT  OUTPUT  DESCRIPTION
C
C        T          YES    NO      CRANK ANGLE (DEG)
C        TEMP       YES    NO      TEMPERATURE (K)
C        P          YES    NO      PRESSURE (ATM)
C        RESFRK     YES    NO      MASS BURNED / TOTAL MASS
C        ------     ---    --      ( =1. FOR BURNED ZONE  )
C        ENTHLP     NO     YES     ENTHALPY  ------
C        CSUBP      NO     YES     --------
C        CSUBT      NO     YES     --------
C        RHO        NO     YES     DENSITY
C        DRHODT     NO     YES     --------
C        DRHODP     NO     YES     --------
C        MW         NO     YES     MOLECULAR WEIGHT
C        GAMMA      NO     YES     RATIO OF SPECIFIC HEATS
C        ADUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        BDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        GDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        HDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C
C     REMARKS
C        THE DUMMY VARIABLES ARE NOT USED IN THE WANKEL PROGRAM
C        BUT HAVE BEEN LEFT INTACT FOR FUTURE USE, IF NECESSARY.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        UPROP      HPROD     CLDPRD
C
C     METHOD
C        SEE PURPOSE, ABOVE
C
C     WRITTEN BY S. G. POULOS
C     EDITED BY S. G. POULOS, AND T. J. NORMAN
C
C
      SUBROUTINE THERMO (T, TEMP, P, RESFRK, ENTHLP, CSUBP, CSUBT, RHO,
     &            DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
C
      REAL MW
      LOGICAL XXC, XXG
      COMMON/TIMES/ TIPO, TIPC, TEPO, TEPC, THIPO, THEPO, TSPARK
C
      IF (RESFRK .GT. .99) GO TO 10
C
      CALL UPROP (P, TEMP, RESFRK, ENTHLP, CSUBP, CSUBT, RHO,
     &                 DRHODT, DRHODP, XXA)
C
      DRHODP = DRHODP/1.01325E+6
CJ    BDUMY  = (1. - RHO*CSUBT)/DRHODP
      R      = 1.01325E+6 * P/( RHO * TEMP )
C
C        IF COMPRESSION OR COMBUSTION PROCESS, PREPARE TO RETURN
```

```
C
      IF (T .GE. TIPC) GO TO 20
C
      CALL UPROP (P, TEMP, 0.0, H1, XXA, XXB, RHO1,
     &            XXC, XXD, XXE)
      H1 = H1 * 4.184E+10
      CALL UPROP (P, TEMP, 1.0, H2, XXA, XXB, RHO2,
     &            XXC, XXD, XXE)
      H2 = H2 * 4.184E+10
      R1 = 1.01325E+6 * P/( RHO1 * TEMP )
      R2 = 1.01325E+6 * P/( RHO2 * TEMP )
CJ    GDUMY = (R1 - R2)/R
CJ    HDUMY = GDUMY + (H2 - H1)/BDUMY
      GO TO 20
C
   10 CALL HPROD (P, TEMP, ENTHLP, CSUBP, CSUBT, RHO, DRHODT, DRHODP)
      R = 1.01325E+6 * P/( RHO * TEMP )
      DRHODP = DRHODP/1.01325E+6
CJ    BDUMY = (1. - RHO*CSUBT)/DRHODP
C
C        CONVERT TO UNITS NEEDED IN MAIN PROGRAM
C
   20 CONTINUE
      CSUBP = CSUBP * 4.184E+7
CJ    ADUMY = CSUBP + ( DRHODT/DRHODP )*( 1./RHO - CSUBT )
      ENTHLP = ENTHLP * 4.184E+10
      MW = 8.3145E+7/R
      GAMMA = CSUBP/( CSUBP - R )
C
      RETURN
      END
C_________________ VERSION 2.1 ___ 2/12/82 ___________________________
C
C       SUBROUTINE UPROP
C
C       PURPOSE
C          TO CALCULATE THE ENTHALPY AND DENSITY OF A HOMOGENEOUS
C          MIXTURE OF AIR, RESIDUAL GAS, AND FUEL AS A FUNCTION OF
C          EQUIVALENCE RATIO, TEMPERATURE, AND PRESSURE
C
C       USAGE
C          CALL UPROP (P, T, RESFRK, ENTHLP, CSUBP, CSUBT, RHO,
C         &            DRHODT, DRHODP, CHI)
C
C       DESCRIPTION OF PARAMETERS
C         GIVEN:
C         P     : ABSOLUTE PRESSURE (ATM)
C         T     : ABSOLUTE TEMPERATURE (DEG K)
C         RESFRK: RESIDUAL GAS FRACTION
C         PHI   : EQUIVALENCE RATIO
C       GIVEN IN COMMON AREA /FUEL/:
C         AF(I) : 6 DIMENSIONAL VECTOR OF ENTHALPY COEFFICIENTS SUCH
C                 THAT THE ENTHALPY OF FUEL VAPOR AS A FUNCTION
C                 OF TEMPERATURE (T DEG K) IS GIVEN BY:
C                  H(T) = AF(1)*ST + (AF(2)*ST**2)/2 + (AF(3)*ST**3)/3
```

```
C                        + (AF(4)*ST**4)/4 - AF(5)/ST +AF(6)
C                  WHERE ST = T/1000 AND H(T) = <KCAL/MOLE>
C                 FOR MOST APPLICATIONS, ENTHALPY FUNCTION H(T) SHOULD
C                 BE VALID AT LEAST OVER THE TEMPERATURE RANGE:
C                                  300 K < T < 1500 K
C
C                 ENTHALPY DATUM STATE IS AT T = 0 ABSOLUTE WITH
C                 O2, N2, AND H2 GASEOUS AND C SOLID GRAPHITE.
C       ENW     : AVERAGE NUMBER OF NITROGEN ATOMS PER FUEL MOLECULE
C       CX      : AVERAGE NUMBER OF CARBON ATOMS PER FUEL MOLECULE
C       HY      : AVERAGE NUMBER OF HYDROGEN ATOMS PER FUEL MOLECULE
C       OZ      : AVERAGE NUMBER OF OXYGEN ATOMS PER FUEL MOLECULE
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS NEEDED
C          NONE
C
C       METHOD
C          DESCRIBED IN APPENDIX A OF HIRES ET AL, SAE PAPER #760161
C
C       REMARKS
C          EXACTLY THE SAME LOGIC AS VERSION 2.0 (1/10/76) BUT CLEANED
C          UP CODING AND DOCUMENTATION BY S. POULOS. 2/12/82
C
C
      SUBROUTINE UPROP (P, T, RESFRK, ENTHLP, CSUBP, CSUBT, RHO,
     &                  DRHODT, DRHODP, CHI)
C
      LOGICAL RICH, LEAN
      INTEGER FUELTP
      REAL*4 MBAR,K
      REAL A(6,7,2), A1(42), A2(42), TABLE(7), AF(6), X(7)
      COMMON/OXDANT/ XI
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
      COMMON/FUPRP/ AF
      EQUIVALENCE (A1(1), A(1,1,1)), (A2(1), A(1,1,2))
C
C        INITIALIZE PARAMETERS, AND CHECK TO SEE IN WHAT TEMPERATURE
C        RANGE WE ARE SO THAT THE CORRECT FITTED COEFFICIENTS WILL BE
C        USED. FLAG TEMPERATURES TOO BIG OR TOO SMALL
C
      DATA A1/11.94033,2.088581,-0.47029,0.037363,-0.589447,-97.1418,
     1   6.139094,4.60783,-0.9356009,6.669498E-02,0.0335801,-56.62588,
     2   7.099556,1.275957,-0.2877457,0.022356,-0.1598696,-27.73464,
     3   5.555680,1.787191,-0.2881342,1.951547E-02,0.1611828,0.76498,
     4   7.865847,0.6883719,-0.031944,-2.68708E-03,-0.2013873,
     5   -0.893455,6.807771,1.453404,-0.328985,2.561035E-02,
     6   -0.1189462,-0.331835,6*0.0/
      DATA A2/4.737305,16.65283,-11.23249,2.828001,6.76702E-03,
     1   -93.75793,7.809672,-0.2023519,3.418708,-1.179013,1.43629E-03,
     2   -57.08004,6.97393,-0.8238319,2.942042,-1.176239,4.132409E-04,
     3   -27.19597,6.991878,0.1617044,-0.2182071,0.2968197,
     4   -1.625234E-02,-0.118189,6.295715,2.388387,-0.0314788,
     5   -0.3267433,4.35925E-03,0.103637,7.092199,-1.295825,3.20688,
     6   -1.202212,-3.457938E-04,-0.013967,6*0.0/
```

```
      DATA TABLE /-1.,1.,1.,-1.,0.,0.,0./
C
C        ENTER INTO ARRAYS A1 AND A2 THE FUEL PARAMETERS
C
      DO 5 I=1,6
          A1(I + 36)=AF(I)
          A2(I + 36)=AF(I)
    5 CONTINUE
      RICH = PHI .GT. 1.0
      LEAN = .NOT. RICH
      W    = ENW/CX
      Z    = CZ/CX
      DEL  = CX/HY
      EPS  = 4.*DEL/(1. + 4.*DEL - 2.*DEL*Z)
      IER  = 0
      IF (T .LT. 100.)  IER = 1
      IF (T .GT. 6000.) IER = 2
      IR = 1
      IF (T .LT. 500.)  IR  = 2
C
C        GET THE COMPOSITION IN MOLES/MOLE OXYGEN OF OXIDANT
C
      PCTRES = RESFRK
      PCTNEW = 1.0 - RESFRK
      IF (RICH) GO TO 10
      X(1) = EPS*PHI*PCTRES
      X(2) = (2.*(1. - EPS) + EPS*Z)*PHI*PCTRES
      X(3) = 0.
      X(4) = 0.
      X(5) = (1. - PHI)*PCTRES + PCTNEW
      DCDT = 0.
      GO TO 20
   10 K     = 3.5
      ALPHA = 1.0 - K
      BETA  = (2.*(1.- EPS*PHI) + K*(2.*(PHI - 1.) + EPS*PHI))
      GAMMA = 2.*K*EPS*PHI*(PHI - 1.)
      C     = (-BETA + SQRT(BETA*BETA + 4.*ALPHA*GAMMA))/(2.*ALPHA)
      X(1)  = (EPS*PHI - C)*PCTRES
      X(2)  = (2.0*(1. - EPS*PHI) + EPS*PHI*Z +C)*PCTRES
      X(3)  = C*PCTRES
      X(4)  = (2.0*(PHI - 1.) - C)*PCTRES
      X(5)  = PCTNEW
   20 X(6)  = XI + EPS*PHI*W/2.*PCTRES
      X(7)  = PCTNEW * EPS*PHI/CX
C
C        CONVERT COMPOSITION TO MOLE FRACTIONS AND CALCULATE AVERAGE
C        MOLECULAR WEIGHT
C
      IF (LEAN) TMOLES = XI + (1. + EPS*PHI/CX)*PCTNEW
     &          + (1. + (1.-EPS)*PHI + EPS*PHI*(Z + W/2.))*PCTRES
      IF (RICH) TMOLES = XI + (1. + EPS*PHI/CX)*PCTNEW
     &          + ((2. - EPS)*PHI + EPS*PHI*(Z + W/2.))*PCTRES
      DO 30 J = 1,7
          X(J) = X(J)/TMOLES
   30 CONTINUE
```

```
      MBAR = EPS*PHI*(12. + 1./DEL + 16.*Z + 14.*W) +32. +28.*XI
      MBAR = MBAR/TMOLES
C
C        CALCULATE H, CP, AND CT AS IN WRITEUP, USING FITTED
C        COEFFICIENTS FROM JANAF TABLES
C
      ENTHLP = 0.
      CSUBP  = 0.
      CSUBT  = 0.
      ST     = T/1000.
      DO 40 J = 1,7
          TH     = ((( A(4,J,IR)/4.*ST + A(3,J,IR)/3.)*ST
     &                + A(2,J,IR)/2.)*ST + A(1,J,IR) )*ST
          TCP    = (( A(4,J,IR)*ST + A(3,J,IR) )*ST
     &                + A(2,J,IR))*ST + A(1,J,IR)
          TH     = TH - A(5,J,IR)/ST + A(6,J,IR)
          TCP    = TCP + A(5,J,IR)/ST**2
          ENTHLP = ENTHLP + TH*X(J)
          CSUBP  = CSUBP + TCP*X(J)
   40 CONTINUE
      ENTHLP = ENTHLP/MBAR
      CSUBP  = CSUBP/MBAR
C
C        NOW CALCULATE RHO AND ITS PARTIAL DERIVATIVES
C        USING PERFECT GAS LAW
C
      RHO    = 0.012187*MBAR*P/T
      DRHODT = -RHO/T
      DRHODP = RHO/P
C
C        CALCULATE PSI AND CHI FOR BURNED GASES
C
      PSI = (XI + EPS*PHI*W/2.)/(1. + EPS*Z*PHI/2.)
      CHI = PHI*(1. + EPS*Z/2.)/(1. + EPS*Z*PHI/2.)
C
        RETURN
        END
C_________________ VERSION 3.2 ___ 2/12/82 ___________________________
C
C       SUBROUTINE CLDPRD
C
C       PURPOSE
C        TO CALCULATE THE SPECIFIC ENTHALPY OF THE PRODUCTS OF HC-AIR
C        COMBUSTION AT TEMPERATURES AND PRESSURES WHERE DISSOCIATION
C        OF THE PRODUCT GASES MAY BE IGNORED.  THE DENSITY OF THE
C        PRODUCT GAS IS ALSO CALCULATED, AS ARE THE PARTIAL
C        DERIVATIVES OF BOTH OF THESE QUANTITIES WITH RESPECT TO
C        PRESSURE AND TEMPERATURE.
C
C       USAGE
C          CALL CLDPRD (P, T, ENTHLP, CSUBP, CSUBT, RHO,
C         &             DRHODT, DRHODP, IER)
C
C       DESCRIPTION OF PARAMETERS
C       GIVEN:
```

```
C          P     : ABSOLUTE PRESSURE OF PRODUCTS (ATM)
C          T     : TEMPERATURE OF PRODUCTS (DEG K)
C          PHI   : EQUIVALENCE RATIO
C          DEL   : MOLAR C:H RATIO OF PRODUCTS
C          PSI   : MOLAR N:O RATIO OF PRODUCTS
C
C       RETURNS:
C          H     : SPECIFIC ENTHALPY OF PRODUCTS (KCAL/G)
C          CP    : PARTIAL DERIVATIVE OF H WITH RESPECT TO T
C                  (CAL/G-DEG K)
C          CT    : PARTIAL DERIVATIVE OF H WITH RESPECT TO P (CC/G)
C          RHO   : DENSITY OF THE PRODUCTS (G/CC)
C          DRHODT: PARTIAL DERIVATIVE OF RHO WITH RESPECT TO T
C                  (G/CC-DEG K)
C          DRHODP: PARTIAL DERIVATIVE OF RHO WITH RESPECT TO P
C                  (G/CC-ATM)
C          IER   : FLAG, SET TO 1 FOR T < 100 DEG K
C                               2 FOR T > 6000 DEG K
C                               0 OTHERWISE
C
C       RETURNS IN COMMON AREA /FROZEN/:
C         CPFROZ: FROZEN SPECIFIC HEAT (CAL/G-DEG K)
C       RETURNS IN COMMON AREA /MBARB/:
C         MBARB : AVERAGE MOLECULAR WEIGHT OF BURNED GASES
C
C       REMARKS
C          1) ENTHALPY DATUM STATE IS AT T = 0 ABSOLUTE WITH
C             O2,N2,H2 GASEOUS AND C SOLID GRAPHITE
C          2) MULTIPLY ATM-CC BY 0.0242173 TO CONVERT TO CAL
C          3) MODIFIED VERSION OF MIKE MARTIN'S PROGRAM
C          4) COMMON BLOCK MBARB ADDED BY B. BEARD 5/10/79
C          5) EXACTLY THE SAME LOGIC AS VERSION 3.1 (5/10/79),
C             BUT WITH CLEANED UP CODE AND DOCUMENTATION BY
C             S. POULOS. 2/12/82
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          NONE
C
C       METHOD
C         SEE MARTIN & HEYWOOD 'APPROXIMATE RELATIONSHIPS FOR THE
C         THERMODYNAMIC PROPERTIES OF HYDROCARBON-AIR COMBUSTION
C         PRODUCTS'
C
C
      SUBROUTINE CLDPRD (P, T, ENTHLP, CSUBP, CSUBT, RHO,
     &                   DRHODT, DRHODP, IER)
C
      LOGICAL RICH, LEAN
      REAL*4 MBAR, K
      REAL MBARB
      DIMENSION A(6,6,2), X(6)
      DIMENSION A1(36), A2(36)
      DIMENSION TABLE(7)
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
```

```
      COMMON/FROZEN/ CPFROZ
      COMMON/MBARB/ MBARB
      EQUIVALENCE (A1(1), A(1,1,1)), (A2(1), A(1,1,2))
C
C        INITIALIZE PARAMETERS, AND CHECK TO SEE IN WHAT TEMPERATURE
C        RANGE WE ARE SO THAT THE CORRECT FITTED COEFFICIENTS WILL BE
C        USED. FLAG TEMPERATURES TOO HIGH OR TOO LOW.
C
      DATA A1/11.94033,2.088581,-0.47029,.037363,-.589447,-97.1418,
     1  6.139094,4.60783,-.9356009,6.669498E-02,.0335801,-56.62588,
     2  7.099556,1.275957,-.2877457,.022356,-.1598696,-27.73464,
     3  5.555680,1.787191,-.2881342,1.951547E-02,.1611828,.76498,
     4  7.865847,.6883719,-.031944,-2.68708E-03,-.2013873,-.893455,
     5  6.807771,1.453404,-.328985,2.561035E-02,-.1189462,-.331835/
      DATA A2/4.737305,16.65283,-11.23249,2.828001,6.76702E-03,
     1  -93.75793,7.809672,-.2023519,3.418708,-1.179013,1.43629E-03,
     2  -57.08004,6.97393,-.8238319,2.942042,-1.176239,4.132409E-04,
     3  -27.19597,6.991878,.1617044,-.2182071,.2968197,-1.625234E-02,
     4  -.118189,6.295715,2.388387,-.0314788,-.3267433,4.35925E-03,
     5  .103637,7.092199,-1.295825,3.20688,-1.202212,-3.457938E-04,
     6  -.013967/
      DATA TABLE /-1.,1.,1.,-1.,0.,0.,0./
C
      RICH = PHI .GT. 1.0
      LEAN = .NOT. RICH
      EPS  = 4.*DEL/(1. + 4.*DEL)
      IER  = 0
      IF (T .LT. 100.)  IER = 1
      IF (T .GT. 6000.) IER = 2
      IR = 1
      IF (T .LT. 500.)  IR  = 2
C
C        GET THE COMPOSITION IN MOLES/MOLE OXYGEN
C
      IF (RICH) GO TO 10
      X(1) = EPS*PHI
      X(2) = 2.*(1.- EPS)*PHI
      X(3) = 0.
      X(4) = 0.
      X(5) = 1.- PHI
      DCDT = 0.
      GO TO 20
   10 K     = 3.5
      ALPHA = 1. - K
      BETA  = (2.*(1.-EPS*PHI) + K*(2.*(PHI - 1.) + EPS*PHI))
      GAMMA = 2.*K*EPS*PHI*(PHI - 1.)
      C     = ( -BETA + SQRT(BETA*BETA + 4.*ALPHA*GAMMA))/(2.*ALPHA)
      X(1)  = EPS*PHI - C
      X(2)  = 2.*(1. - EPS*PHI) + C
      X(3)  = C
      X(4)  = 2.*(PHI - 1.) - C
      X(5)  = 0.
   20 X(6)  = PSI
C
C        CONVERT COMPOSITION TO MOLE FRACTIONS AND CALCULATE AVERAGE
```

```
      COMMON/FROZEN/ CPFROZ
      COMMON/MBARB/ MBARB
      EQUIVALENCE (A1(1), A(1,1,1)), (A2(1), A(1,1,2))
C
C        INITIALIZE PARAMETERS, AND CHECK TO SEE IN WHAT TEMPERATURE
C        RANGE WE ARE SO THAT THE CORRECT FITTED COEFFICIENTS WILL BE
C        USED. FLAG TEMPERATURES TOO HIGH OR TOO LOW.
C
      DATA A1/11.94033,2.088581,-0.47029,.037363,-.589447,-97.1418,
     1  6.139094,4.60783,-.9356009,6.669498E-02,.0335801,-56.62588,
     2  7.099556,1.275957,-.2877457,.022356,-.1598696,-27.73464,
     3  5.555680,1.787191,-.2881342,1.951547E-02,.1611828,.76498,
     4  7.865847,.6883719,-.031944,-2.68708E-03,-.2013873,-.893455,
     5  6.807771,1.453404,-.328985,2.561035E-02,-.1189462,-.331835/
      DATA A2/4.737305,16.65283,-11.23249,2.828001,6.76702E-03,
     1  -93.75793,7.809672,-.2023519,3.418708,-1.179013,1.43629E-03,
     2  -57.08004,6.97393,-.8238319,2.942042,-1.176239,4.132409E-04,
     3  -27.19597,6.991878,.1617044,-.2182071,.2968197,-1.625234E-02,
     4  -.118189,6.295715,2.388387,-.0314788,-.3267433,4.35925E-03,
     5  .103637,7.092199,-1.295825,3.20688,-1.202212,-3.457938E-04,
     6  -.013967/
      DATA TABLE /-1.,1.,1.,-1.,0.,0.,0./
C
      RICH = PHI .GT. 1.0
      LEAN = .NOT. RICH
      EPS  = 4.*DEL/(1. + 4.*DEL)
      IER  = 0
      IF (T .LT. 100.)  IER = 1
      IF (T .GT. 6000.) IER = 2
      IR = 1
      IF (T .LT. 500.)  IR  = 2
C
C        GET THE COMPOSITION IN MOLES/MOLE OXYGEN
C
      IF (RICH) GO TO 10
      X(1) = EPS*PHI
      X(2) = 2.*(1.- EPS)*PHI
      X(3) = 0.
      X(4) = 0.
      X(5) = 1.- PHI
      DCDT = 0.
      GO TO 20
   10 K     = 3.5
      ALPHA = 1. - K
      BETA  = (2.*(1.-EPS*PHI) + K*(2.*(PHI - 1.) + EPS*PHI))
      GAMMA = 2.*K*EPS*PHI*(PHI - 1.)
      C     = ( -BETA + SQRT(BETA*BETA + 4.*ALPHA*GAMMA))/(2.*ALPHA)
      X(1)  = EPS*PHI - C
      X(2)  = 2.*(1. - EPS*PHI) + C
      X(3)  = C
      X(4)  = 2.*(PHI - 1.) - C
      X(5)  = 0.
   20 X(6)  = PSI
C
C        CONVERT COMPOSITION TO MOLE FRACTIONS AND CALCULATE AVERAGE
```

```
C        MOLECULAR WEIGHT
C
      IF (LEAN) TMOLES = 1. + PSI + PHI*(1.-EPS)
      IF (RICH) TMOLES = PSI + PHI*(2.-EPS)
      DO 30 J = 1, 6
          X(J) = X(J)/TMOLES
   30 CONTINUE
      MBAR = ((8.*EPS + 4.)*PHI + 32. + 28.*PSI)/TMOLES
C*****
      MBARB = MBAR
C*****
C        CALCULATE H, CP, AND CT AS IN WRITEUP, USING FITTED
C        COEFFICIENTS FROM JANAF TABLES
C
      ENTHLP = 0.
      CSUBP  = 0.
      CSUBT  = 0.
      CPFROZ = 0.
      ST     = T/1000.
      DO 40 J = 1,6
          TH     = ((( A(4,J,IR)/4.*ST + A(3,J,IR)/3.)*ST
     &               + A(2,J,IR)/2.)*ST + A(1,J,IR) )*ST
          TCP    = (( A(4,J,IR)*ST + A(3,J,IR) )*ST
     &               + A(2,J,IR) )*ST + A(1,J,IR)
          TH     = TH - A(5,J,IR)/ST + A(6,J,IR)
          TCP    = TCP + A(5,J,IR)/ST**2
          ENTHLP = ENTHLP + TH*X(J)
          CSUBP  = CSUBP + TCP*X(J)
   40 CONTINUE
      ENTHLP = ENTHLP/MBAR
      CSUBP  = CSUBP/MBAR
C
C        NOW CALULATE RHO AND ITS PARTIAL DERIVATIVES
C        USING PERFECT GAS LAW
C
      RHO    = .012187*MBAR*P/T
      DRHODT = -RHO/T
      DRHODP = RHO/P
C
      RETURN
      END
C_____________________________________________________________________
C
C      SUBROUTINE HPROD
C
C      PURPOSE
C         TO CALCULATE THE PROPERTIES OF THE PRODUCTS OF HYDROCARBON-
C         AIR COMBUSTION AS A FUNCTION OF TEMPERATURE AND PRESSURE,
C         USING AN APPROXIMATE CORRECTION FOR DISSOCIATION.
C         H AND RHO ARE CALCULATED AS FUNCTIONS OF R, T, AND PHI.
C         THE PARTIAL DERIVATIVES OF H AND RHO WITH RESPECT TO
C         P AND T ARE ALSO CALCULATED
C
C       USAGE
C          CALL HPROD (P, T, H, CP, CT, RHO, DRHODT, DRHODP)
```

```
C
C       DESCRIPTION OF PARAMETERS
C       GIVEN:
C         P     : ABSOLUTE PRESSURE OF PRODUCTS (ATM)
C         T     : TEMPERATURE OF PRODUCTS (DEG K)
C         PHI   : EQUIVALENCE RATIO
C         DEL   : MOLAR C:H RATIO OF PRODUCTS
C         PSI   : MOLAR N:O RATIO OF PRODUCTS
C
C       RETURNS:
C         H     : SPECIFIC ENTHALPY OF PRODUCTS (ATM)
C         CP    : PARTIAL DERIVATIVE OF H WITH RESPECT TO T
C                 (CAL/G-DEG K)
C         CT    : PARTIAL DERIVATIVE OF H WITH RESPECT TO P (CC/G)
C         RHO   : DENSITY OF THE PRODUCTS (G/CC)
C         DRHODT: PARTIAL DERIVATIVE OF RHO WITH RESPECT TO T
C                 (G/CC-DEG K)
C         DRHODP: PARTIAL DERIVATIVE OF RHO WITH RESPECT TO P
C                (G/CC-ATM)
C
C       RETURNS IN COMMON AREA /FROZEN/:
C         CPFROZ: FROZEN SPECIFIC HEAT (CAL/G-DEG K)
C       RETURNS IN COMMON AREA /MBARB/:
C         MBARB : AVERAGE MOLECULAR WEIGHT OF BURNED GASES
C
C       REMARKS
C          1) ENTHALPY DATUM STATE IS AT T = 0 ABSOLUTE WITH
C             O2, N2, H2 GASEOUS AND C SOLID GRAPHITE
C          2) MULTIPLY ATM-CC BY 0.0242173 TO CONVERT TO CAL
C          3) MODIFIED VERSION OF MIKE MARTIN'S PROGRAM
C          4) COMMON BLOCK MBARB ADDED BY B. BEARD 5/10/79
C          5) EXACTLY THE SAME LOGIC AS VERSION 3.5 (5/10/79),
C             BUT WITH CLEANED UP CODE AND DOCUMENTATION BY
C             S. POULOS. 2/12/82
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C         CLDPRD
C
C       METHOD
C         SEE MARTIN AND HEYWOOD 'APPROXIMATE RELATIONSHIPS FOR THE
C         THERMODYNAMIC PROPERTIES OF HYDROCARBON-AIR COMBUSTION
C         PRODUCTS'
C
C
      SUBROUTINE HPROD (P, T, H, CP, CT, RHO, DRHODT, DRHODP)
C
      LOGICAL RICH, LEAN, NOTHOT, NOTWRM, NOTCLD
      REAL MCP, MWT, K1, K2
      REAL MBARB
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
      COMMON/FROZEN/ CPFROZ
      COMMON/MBARB/ MBARB
C
C        INITIALIZE PARAMETERS USED IN THE CALCULATION
```

```
C
      DATA R,ROVER2 /1.9869,0.99345/, PSCALE /2.42173E-2/
      DATA TCOLD,THOT /1000.,1100./
C
      RICH   = PHI .GE. 1.0
      LEAN   = .NOT. RICH
      NOTHOT = T .LT. THOT
      NOTCLD = T .GT.TCOLD
      NOTWRM = .NOT. (NOTCLD .AND. NOTHOT)
      EPS    = (4.*DEL)/(1. + 4.*DEL)
C
C        USE SIMPLE ROUTINE FOR LOW TEMPERATURE MIXES
C
      IF (NOTCLD) GO TO 5
      CALL CLDPRD (P, T, H, CP, CT, RHO, DRHODT, DRHODP, IER)
      RETURN
C
C        CALCULATE EQUILIBRIUM CONSTANTS FOR DISSOCIATION (EQS. 3.9
C        & 3.10) (NOTE THAT THESE HAVE UNITS ATM**(.5) )
C
    5 K1 = 5.819E-6 * EXP(0.9674*EPS + 35810./T)
      K2 = 2.961E-5 * EXP(2.593*EPS  + 28980./T)
C
C        CALCULATE A, X, Y, & U AS IN EQS. 5.24, 3.6, 5.25, 3.7, 2.18,
C        2.19, & 3.8
C
      C5 = 2.- EPS + PSI
      A  = (C5/(4.*P*K1*K1*EPS))**(.33333333)
C
      C6 = EPS + 2.*C5
      X  = A*EPS*(3.*C5 + C6*A)/(3.*(1.+ 2.*A)*C5 + 2.*C6*A*A)
C
      Z  = ABS((1.-PHI)/X)
      IF (LEAN) Y = X/SQRT(1.+ .666667*Z + 1.3333333*(1.-PHI))
      IF (RICH) Y = X/(1.+.6666667*Z +.3333333*Z*Z -.6666667*(PHI-1.))
      U  = C5*(EPS - 2.*X)/(4.*K1*K2*P*X)
C
C     CALCULATE THE ENTHALPY OF FORMATION FOR THIS APPROXIMATE
C     COMPOSITION AS IN EQS. 3.21, 3.22, & 5.7
C     ALSO GET THE COEFFICIENTS FOR T & TV TERMS IN 3.15 USING 5.3 & 5.4
C
      HF   = 1000.*((121.5 + 29.59*EPS)*Y + 117.5*U)
      HF   = HF + (20372.*EPS - 114942.)*PHI
      C1   = 7.*PSI + 5.*Y + 3.*U
      C2   = 2.*(PSI - 3.*Y - U)
C
      IF (LEAN) GO TO 10
C
C        RICH CASE
C
      HF = HF + 1000.*(134.39 - 6.5/EPS)*(PHI - 1.)
      C1 = 2. + 2.*(7.- 4.*EPS)*PHI + C1
      C2 = 8. + 2.*(2.- 3.*EPS)*PHI + C2
      GO TO 20
C
```

```
C        LEAN CASE

   10 C1 = 7. +     (9.- 8.*EPS)*PHI + C1
      C2 = 2. + 2.* (5.- 3.*EPS)*PHI + C2
C
C        ADD IN TRANSLATIONAL, VIBRATIONAL, AND ROTATIONAL TERMS
C        TO GET TOTAL ENTHALPY, USING EQS. 3.16, 5.6, 3.11, & 3.15
C
   20 TV     = (3256.- 2400.*EPS + 300.*PSI)/(1.- .5*EPS + .09*PSI)
      EXPTVT = EXP(TV/T)
      TVTIL  = TV/(EXPTVT - 1.)
      MCP    = (8.*EPS + 4.)*PHI + 32. + 28.*PSI
C
      H      = 0.001*ROVER2*(C1*T + C2*TVTIL + HF)/MCP
C
C        CALCULATE THE AVERAGE MOLECULAR WEIGHT, AND GET DENSITY
C        BY USING THE PERFECT GAS LAW - EQS. 3.12, 3.13, & 3.14
C
      IF (LEAN) MWT = MCP/(1. + (1.- EPS)*PHI + PSI + Y + U)
      IF (RICH) MWT = MCP/(     (2.- EPS)*PHI + PSI + Y + U)
C*****
      MBARB = MWT
C*****
      RHO = MWT*P*PSCALE/(R*T)
C
C        GET PARTIAL DERIVATIVES IF DESIRED
C
C        THE FOLLOWING USES IN ORDER EQS. 5.8, 5.9, 5.32, 5.31, 5.30,
C        5.29, 5.28, & 5.26
C
      C3 = (121.5 + 29.59*EPS)*1000.
      C4 = 1.175E5
C
      DUDTPX = 64790.*U/(T*T)
      DUDPTX = -U/P
      DUDXPT = -U*EPS/(X*(EPS - 2.*X))
C
      DADTP  = 23873.*A/(T*T)
      DADPT  = -A/(3.*P)
C
      T5 = 3.*C5
      DXDA = T5*EPS*(T5 + 2.*C6*A)/(T5*(1. + 2.*A) + 2.*C6*A*A)**2
C
C        FOLLOWING USES EQS. 5.23, 5.19-5.22, 5.18-5.14, 5.12, & 5.13
C
      IF (LEAN) DYDX = (Y*Y*Y)/(X*X*X) * (1.+ Z + 1.333333*(1.-PHI))
      IF (RICH) DYDX = (Y*Y)/(X*X)*(1.+ 4.*Z/3. + Z*Z -2.*(PHI-1.)/3.)
C
      DYDTP = DYDX*DXDA*DADTP
      DYDPT = DYDX*DXDA*DADPT
      DUDTP = DUDXPT*DXDA*DADTP + DUDTPX
      DUDPT = DUDXPT*DXDA*DADPT + DUDPTX
C
      DHFDPT = C3*DYDPT + C4*DUDPT
      DC2DPT = -2.*(3.*DYDPT + DUDPT)
```

```
      DC1DPT = 5.*DYDPT + 3.*DUDPT
      DHFDTP = C3*DYDTP + C4*DUDTP
      DC2DTP = -2.*(3.*DYDTP + DUDTP)
      DC1DTP = 5.*DYDTP + 3.*DUDTP
C
      DTVDTP = (TVTIL*TVTIL)/(T*T)*EXPTVT
C
C        FOLLOWING USES EQS. 5.10, & 5.11
C
      CPFROZ = ROVER2/MCP*(C1 + C2*DTVDTP)
C
      CP = ROVER2/MCP*(C1 + T*DC1DTP + C2*DTVDTP + TVTIL*DC2DTP
     &     + DHFDTP)
      CT = ROVER2/MCP*(T*DC1DPT + TVTIL*DC2DPT + DHFDPT)*PSCALE
C
C        FOLLOWING USES EQS. 5.46, 5.35-5.37, 5.33, & 5.34
C
      IF (LEAN) D = 1. + (1.- EPS)*PHI + PSI + Y + U
      IF (RICH) D =      (2.- EPS)*PHI + PSI + Y + U
      G     = -MCP/(D*D)
      DMDTP = G*(DYDTP + DUDTP)
      DMDPT = G*(DYDPT + DUDPT)
C
      DRHODT = PSCALE*P*(DMDTP - MWT/T)/(R*T)
      DRHODP = PSCALE*(MWT + P*DMDPT)/(R*T)
C
C        IF CALCULATING FOR AN INTERMEDIATE TEMPERATURE, USE A
C        WEIGHTED AVERAGE OF THE RESULTS FROM THIS ROUTINE AND
C        THOSE FROM THE SIMPLE ROUTINE
C
      IF (NOTWRM) RETURN
C
      CALL CLDPRD (P, T, TH, TCP, TCT, TRHO, TDRT, TDRP, IER)
      W1 = (T - TCOLD)/(THOT - TCOLD)
      W2 = 1.0 - W1
C
      H   = W1*H   + W2*TH
      RHO = W1*RHO + W2*TRHO
      CP  = W1*CP  + W2*TCP
      CT  = W1*CT  + W2*TCT
      DRHODT = W1*DRHODT + W2*TDRT
      DRHODP = W1*DRHODP + W2*TDRP
C
      RETURN
      END
C
C
C     SUBROUTINE ITRATE
C
C     PURPOSE
C        THIS SUBROUTINE IS CALLED TO OBTAIN T GIVEN P, H, RESFRK,
C        AND A GUESS FOR T.  'ITRATE' CALLS 'THERMO' WITH TGUESS.
C        'THERMO' RETURNS WITH THE ENTHALPY CORRESPONDING TO THE
C        GIVEN TGUESS.  THEN A NEW CORRECTED VALUE FOR TGUESS
C        IS CALCULATED BY USING THE DEFINITION OF CSUBP AND THE
```

```
C        KNOWN VALUES OF CORRECT H AND RETURNED HGUESS.  THIS PRO-
C        CEDURE IS REPEATED AT MOST MAXTRY TIMES, OR FEWER TIMES
C        IF ACCURACY MAXERR IS ACHIEVED.
C
C     USAGE
C        CALL ITRATE (T, TGUESS, P, RESFRK, ENTHLP, CSUBP, CSUBT,
C       &             RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY,
C       &             GDUMY, HDUMY)
C
C     DESCRIPTION OF PARAMETERS
C        PARAMETER  INPUT  OUTPUT  DESCRIPTION
C
C        T          YES    NO      CRANK ANGLE (DEG)
C        TGUESS     YES    YES     TEMPERATURE GUESS (K)
C        ------     ---    --      (CORRECTED VALUE IS RETURNED)
C        P          YES    NO      PRESSURE (ATM)
C        RESFRK     YES    NO      MASS BURNED / TOTAL MASS
C        ------     ---    --      ( <1. FOR UNBURNED ZONE ONLY )
C        ENTHLP     YES    NO      ENTHALPY ON WHICH TO ITERATE (ERG)
C        HGUESS     NO     NO      ENTHALPY GUESS (ERG)
C        CSUBP      NO     YES     DH/DT @ CONSTANT P (ERG/K)
C        CSUBT      NO     YES     DH/DP @ CONSTANT T (ERG/ATM)
C        RHO        NO     YES     DENSITY
C        DRHODT     NO     YES     PARTIAL OF RHO WITH RESPECT TO T
C        DRHODP     NO     YES     PARTIAL OF RHO WITH RESPECT TO P
C        MW         NO     YES     MOLECULAR WEIGHT
C        GAMMA      NO     YES     RATIO OF SPECIFIC HEATS
C        ADUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        BDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        GDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C        HDUMY      NO     YES     SEE ASSIGNMENT STATEMENTS BELOW
C
C     REMARKS
C        NONE
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        THERMO
C
C     METHOD
C        SEE PURPOSE, ABOVE
C
C     WRITTEN BY S. G. POULOS
C     EDITED BY S. G. POULOS
C
      SUBROUTINE ITRATE (T, TGUESS, P, RESFRK, ENTHLP, CSUBP, CSUBT,
     &                   RHO, DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY,
     &                   GDUMY, HDUMY)
C
      REAL MW, MAXERR
      COMMON/ITRLIM/ MAXTRY, MAXERR
C
      DO 10 I = 1, MAXTRY
         CALL THERMO (T, TGUESS, P, RESFRK, HGUESS, CSUBP, CSUBT, RHO,
     &           DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
         TOLD = TGUESS
```

```
         TGUESS = TOLD + (ENTHLP - HGUESS)/CSUBP
         IF( ABS((TGUESS - TOLD)/TGUESS ) .LE. MAXERR ) GO TO 20
   10 CONTINUE
C
   20 CALL THERMO (T, TGUESS, P, RESFRK, HGUESS, CSUBP, CSUBT, RHO,
     &        DRHODT, DRHODP, GAMMA, MW, ADUMY, BDUMY, GDUMY, HDUMY)
C
      RETURN
      END
C_____________________________________________________________________
C
C     SUBROUTINE FUELDT
C
C     PURPOSE
C        THIS SUBROUTINE IS CALLED TO SET THE VALUES OF THE FUEL
C        RELATED PARAMETERS AT THE START OF PROGRAM EXECUTION. THE
C        ONLY INPUT REQUIRED IS THE FUEL TYPE.  THE PARAMETERS WHICH
C        ARE SET ARE: I) THE ATOM RATIOS WHICH SPECIFY THE PROPERTIES
C        OF THE FUEL- AIR MIXTURE, THE SET OF ENTHALPY COEFFICIENTS
C        ASSOCIATED WITH THE FUEL (USED IN THE PROPERTY ROUTINES), AND
C        THE FUEL HEATING VALUE AND STOICHIOMETRIC FUEL/AIR RATIO;
C        II) THE ATOM RATIOS REQUIRED BY SUBROUTINE 'PTCHEM' FOR
C        CALCULATION OF EQUILIBRIUM BURNED GAS COMPOSITION.
C
C     USAGE
C        CALL FUELDT
C
C     DESCRIPTION OF PARAMETERS
C        PARAMETER  INPUT  OUTPUT  DESCRIPTION
C
C        FUELTP     YES    NO      FUEL TYPE
C        PSI        NO     YES     MOLAR N2 TO 02 RATIO FOR AIR
C        XI         NO     YES     MOLAR N2 TO 02 RATIO FOR AIR
C        CX         NO     YES     # OF CARBON ATOMS/FUEL MOLECULE
C        DEL        NO     YES     CARBON/HYDROGEN RATIO OF FUEL
C        HY         NO     YES     HYDROGEN ATOMS PER FUEL MOLECULE
C        ENW        NO     YES     NITROGEN ATOMS PER FUEL MOLECULE
C        OZ         NO     YES     OXYGEN ATOMS PER FUEL MOLECULE
C        QLOWER     NO     YES     LOWER HEATING VALUE OF THE FUEL
C        FASTO      NO     YES     STOICHIOMETRIC FUEL/AIR RATIO
C        AF(I)      NO     YES     FUEL COEFFICIENT ARRAY
C        D(I)       NO     YES     ATOM RATIO ARRAY (SEE PTCHEM)
C
C     REMARKS
C        ONLY ISSOCTANE AND PROPANE ARE AVAILABLE FOR
C        USE AS FUELS.
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        ______    ______    ______    ______
C
C     METHOD
C        SEE PURPOSE, ABOVE
C
C     WRITTEN BY S. G. POULOS
C     EDITED BY S. G. POULOS
```

```
C
      SUBROUTINE FUELDT
C
      INTEGER FUELTP
      REAL AF(6)
      REAL*8 D(4)
C
      COMMON/FUEL/ FUELTP, ENW, CX, HY, OZ, DEL, PSI, PHICON, PHI,
     &             QLOWER, FASTO
      COMMON/FUPRP/ AF
      COMMON/OXDANT/ XI
      COMMON/CHEM/ D
C
      PSI = 3.76
      XI = 3.76
      IF (FUELTP .GT. 1) GO TO 10
C
C        FOLLOWING DATA FOR ISOOCTANE (FUELTP = 1)
C
      CX = 8.0
      DEL = 8.0/18.0
      HY = 18.0
      ENW = 0.0
      OZ = 0.0
      QLOWER = 44.392
      FASTO = 1./15.11
C
      AF(1) =   -0.55313
      AF(2) =  181.62
      AF(3) =  -97.787
      AF(4) =   20.402
      AF(5) =   -0.03095
      AF(6) =  -60.518
C
      GO TO 20
C
C        FOLLOWING DATA FOR PROPANE (FUELTP = 2)
C
   10 CX = 3.0
      DEL = 3.0/8.0
      HY = 8.0
      ENW = 0.0
      OZ = 0.0
      QLOWER = 46.3
      FASTO = 0.0638
C
      AF(1) =  - 1.4867
      AF(2) =   74.339
      AF(3) =  -39.0649
      AF(4) =    8.05426
      AF(5) =    0.0121948
      AF(6) =  -18.4611
C
   20 PHICON = (32. + 28. * PSI) * (DEL + .25)/(12. * DEL + 1.0)
C
```

```
C        CALCULATE ATOM RATIOS FOR USE BY 'PTCHEM'
C
      D(1) = 1./DEL
      D(2) = 1.0
      D(4) = 2.*(1. + 0.25/DEL)/PHI
      D(3) = PSI*D(4)
C
      RETURN
      END
C
C
      SUBROUTINE PTCHEM (TMP, PRS, D, XMOFR, ISENT)
C
C
C     THE ORIGINAL MIT EQUILIBRIUM PTCHEM SUBROUTINE HAS BEEN REDUCED
C     AND IS DESIGNED TO PROCESS 4 ELEMENTS & 14 SPECIES
C
C     ARRAY D CONTAINS RATIOS OF ELEMENTS H, C, N, O RESPECTIVELY;
C
C     E.G., DATA D / 1.43, 1.0, 13.0, 3.46 /
C
C
C     ARRAY XMOFR CONTAINS MOLE FRACTIONS OF EACH SPECIES;
C     SYMBOLS OF EACH SPECIES 1 TO 14 (HEX'E') ARE AS FOLLOWS:
C
C    0          'HCO ', 'CO  ', 'CO2 ', 'H   ', 'OH  ', 'H2  ',
C    1          'H2O2', 'N   ', 'NO  ', 'NO2 ', 'N2  ', 'N2O ',
C    2          'O   ', 'O2  ' /
C
C
C     INPUT:    ( DOUBLE PREC. )
C
C        TMP   = TEMPERATURE ( DEGREES K )
C        PRS   = PRESSURE  ( ATMOSPHERES )
C        D     = ARRAY(4)  ( RATIO, 4 ELEMENTS )
C
C     OUTPUT:   ( DOUBLE PREC. )
C
C        XMOFR = ARRAY(14) ( MOLE FRACTIONS, 14 SPECIES )
C        ISENT = ERROR CODE:
C            0 = NO ERROR
C            1 = TEMP TOO HIGH
C            2 = TEMP TOO LOW
C            3 = ( UNUSED )
C            4 = TOO MANY ITERATIONS, RESULTS DOUBTFUL, LOOK UP NU
C            5 = TOO MANY ITERATIONS
C            6 = THERE ARE NO GASES PRESENT
C            7 = CHECK IF THERE ARE ENOUGH SPECIES
C            8 = TOO MANY TRIES FOR T
C
C     AUTHOR:  DAN DANTZER     LATEST REVISION 12/14/72
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
```

```
      EQUIVALENCE ( ITOTSP, M      )
      EQUIVALENCE ( ITOTSP, M1     )
      EQUIVALENCE ( NELEM , N      )
      EQUIVALENCE ( G(1,1), CMN(1) )
C
      DIMENSION D(1), XMOFR(1)
      DIMENSION A(4,14), ZL(14,8)
      DIMENSION ZL1(56), ZL2(56)
C
C        FOLLOWING ARRAYS ARE 'NELEM' IN SIZE
C
      DIMENSION DD(4), XMU(4), XNU(4), E(4), F(4,4)
C
C        FOLLOWING ARRAYS ARE 'NELEM' + 2  = 'INXNSL'IN SIZE
C
      DIMENSION R(6), G(6,6)
C
C        FOLLOWING ARRAYS ARE 'ITOTSP' IN SIZE
C
      DIMENSION CMN(14), CP(14), CPT(14), X(14), XMAX(14)
      DIMENSION HORT(14), SR(14), C(14)
C
C
C        NSAME  = INITIALIZATION FLAG
C        NELEM  = NUMBER OF ELEMENTS INVOLVED
C        ITOTSP = TOTAL NUMBER OF SPECIES INVOLVED
C
      DATA NSAME  /  0 /
      DATA NELEM  /  4 /
      DATA INXNSL /  6 /
      DATA ITOTSP / 14 /
C
      DATA  DD, XMU, XNU, E, F  /  32 * 0. /
      DATA  R, G  /  42 * 0. /
      DATA  A, CMN, CP, CPT, X, XMAX  /  126 * 0. /
C
C         ARRAY ZL(K,J) CONTAINS THE HI-TEMP DATA FOR EACH OF 14 SPECIES
C
C           K = SPECIES NUMBER ( 1 - 14 )
C           J = DATA ( 1 - 8 EACH SPECIES )
C
C         DATA WAS TAKEN FROM ORIGINAL DATA SET ( CARDS), ARRANGED IN
C         ROW,COL (14,8),  BUT DATA STATEMENT INTERNALLY STORES
C         BY COL,ROW;  THEREFORE,  DATA MUST BE REVERSED FOR PROG EXEC.
C         ORIGINAL DATA ARRANGEMENT MAINTAINED FOR CONVENIENCE IN MAKING
C         CHANGES TO SPECIES DATA.
C
C         NUMBER OF CONTINUATION CARDS LIMITED TO 19
C         CONTINUATION COL #6 CONTAINS THE HEXIDECIMAL SPECIES NUMBER
C
      DATA  ZL1  /
     1  9.3434439D 00, 2.9512196D 00,-6.7088366D-01, 5.0901942D-02,
     1 -4.1140777D-01,-6.9903498D 00, 6.2395554D 01, 2.3009987D 00,
     2  7.0995646D 00, 1.2759562D 00,-2.8774744D-01, 2.2356123D-02,
     2 -1.5986955D-01,-2.9023636D 01, 5.4823700D 01, 1.2890015D 00,
```

```
     3  1.1940331D 01, 2.0885811D 00,-4.7029203D-01, 3.7363116D-02,
     3 -5.8944768D-01,-9.9468796D 01, 6.2207169D 01, 2.3269949D 00,
     4  4.9679995D 00,-5.9678716D-13, 1.9345219D-13,-1.8732195D-14,
     4 -7.4194936D-14, 5.0619217D 01, 3.3404419D 01, 1.0119925D 00,
     5  5.6197815D 00, 1.9668446D 00,-3.8645178D-01, 2.7364515D-02,
     5  1.3418680D-01, 8.0923529D 00, 5.0777405D 01, 2.0489998D 00,
     6  5.5556803D 00, 1.7871914D 00,-2.8813416D-01, 1.9515470D-02,
     6  1.6118270D-01,-1.2590199D 00, 3.8126053D 01, 2.0239983D 00,
     7  6.1390944D 00, 4.6078291D 00,-9.3560094D-01, 6.6694975D-02,
     7  3.3580098D-02,-5.9687866D 01, 5.1456772D 01, 3.0619965D 00 /
      DATA  ZL2  /
     8  5.1632023D 00,-1.8977487D-01, 3.6921006D-02, 3.6241105D-03,
     8 -3.2718156D-02, 1.1134157D 02, 4.2783066D 01, 1.0359793D 00,
     9  7.5180626D 00, 1.0245209D 00,-2.3053735D-01, 1.7926671D-02,
     9 -1.9369543D-01, 1.8756821D 01, 5.8378296D 01, 2.0729971D 00,
     A  1.2123282D 01, 1.2564993D 00,-3.0112296D-01, 2.3658749D-02,
     A -6.2256289D-01, 2.3839598D 00, 6.8810959D 01, 3.1109982D 00,
     B  6.8077698D 00, 1.4534035D 00,-3.2898575D-01, 2.5610346D-02,
     B -1.1894619D-01,-2.4038353D 00, 5.3165161D 01, 2.0719986D 00,
     C  1.2365961D 01, 1.7261782D 00,-4.0528846D-01, 3.1418435D-02,
     C -5.8120877D-01, 1.4105570D 01, 6.4331467D 01, 3.1099997D 00,
     D  5.1006308D 00,-1.5177220D-01, 4.8953138D-02,-2.8814352D-03,
     D  8.9299418D-03, 5.8080948D 01, 4.4752548D 01, 1.0379944D 00,
     E  7.8658457D 00, 6.8837190D-01,-3.1944100D-02,-2.6870817D-03,
     E -2.0138729D-01,-2.9684544D 00, 5.7424637D 01, 2.0749989D 00 /
C
      IF (NSAME) 2000, 2000, 2004
 2000 NSAME = 1
      AR = 1.98726D0
      ITMAX = 500
      ITW  = 400
      DIF = 15.0D0
      DIF1 = DIF
      T = 1.0D0
      TLUB = 6000.
      XN = N
      TOL1 = .01D0
      TOL3 = .00001D0
      TOL5 = 1.0D-5
      TOL6 = .1D0
      TOL7 = 10.0D0
C
C         ARRAY A(I,K) CONTAINS NUMBER OF ATOMS PER ELEMENT
C
C           I = ELEMENTS H, C, N, O RESPECTIVELY
C           K = SPECIES NUMBER
C
      A(1,1) = 1.
      A(1,4) = 1.
      A(1,5) = 1.
      A(1,6) = 2.
      A(1,7) = 2.
      A(2,1) = 1.
      A(2,2) = 1.
      A(2,3) = 1.
```

```
      A(3,8) = 1.
      A(3,9) = 1.
      A(3,10)= 1.
      A(3,11)= 2.
      A(3,12)= 2.
      A(4,1) = 1.
      A(4,2) = 1.
      A(4,3) = 2.
      A(4,5) = 1.
      A(4,7) = 1.
      A(4,9) = 1.
      A(4,10)= 2.
      A(4,12)= 1.
      A(4,13)= 1.
      A(4,14)= 2.
C
C     LOAD DATA INTO SPECIES ARRAY ZL
C
      II = 0
      DO 1060 K = 1, 7
         I = K + 7
         DO 1050 J = 1, 8
            II = II + 1
            ZL(K,J) = ZL1(II)
            ZL(I,J) = ZL2(II)
 1050    CONTINUE
 1060 CONTINUE
C
C         END OF ONE-TIME INITIALIZATION
C
 2004 IF ( TMP - 700.0 )  2112, 2112, 2355
 2355 IF ( TMP - TLUB ) 2012, 2012, 2009
 2009 ISENT = 1
      GO TO 5000
 2112 ISENT = 2
      GO TO 5000
 2012 TK  = TMP / 1.0D+3
      XLP = DLOG( PRS )
C
C         START OF ORIGINAL 'HS' SUBROUTINE
C
      DO 5004 K10 = 1, M
         IF ( ZL(K10,1) )  5003, 5002, 5003
 5002    HORT(K10) = 0.11111111D0
         SR (K10) = -1.0D6
         GO TO 5004
 5003    HORT(K10) = (((( ZL(K10,4) * TK / 4.D0 + ZL(K10,3) / 3.0D0 ) *
     1                TK + ZL(K10,2) / 2.D0 ) * TK  + ZL(K10,1) ) * TK -
     2                ZL(K10,5) / TK + ZL(K10,6) ) / ( AR * TK )
         SR(K10) = ( ZL(K10,1) * DLOG(TK) + TK * ( ZL(K10,2) +
     1              ZL(K10,3) * 0.5D0 * TK + ZL(K10,4) * TK **2/3.D0 ) -
     2              ZL(K10,5) * 0.5D0 / TK ** 2 + ZL(K10,7) ) / AR
 5004 C(K10) = HORT(K10) - SR(K10) + XLP
C
C         END OF ORIGINAL 'HS' SUBROUTINE
```

```
C
      ISENT = 0
      ITER = 0
      ITTRDG = 0
      TOL4 = 0.1D0
      YMAX = 0.
      DO 412 J = 1, M1
         XMAX(J) = 1.0D10
         DO 412 I = 1, N
         IF (A(I,J)) 412, 412, 413
  413    IF (D(I)/A(I,J) - XMAX(J)) 414, 412, 412
  414    XMAX(J) = D(I)/A(I,J)
  412 CONTINUE
      XO = 0.0D0
      DO 90 I = 1, N
         XO = XO + D(I)
   90 CONTINUE
      AVD = XO/XN
      YO = 0.0D0
      DO 93 J = 1, M
         YO = YO + XMAX(J)
   93 CONTINUE
      XO = DMIN1(XO,YO)
      XO = XO * 1.05D0
      DO 825 J = 1, M1
         XMAX(J) = XMAX(J)*1.05D0
  825 CONTINUE
C
C         CAUTION !!! ASSUMPTION: THAT EACH ELEMENT MOLE RATIO DIVIDED
C         BY THE SUM OF THE RATIOS WILL BE GREATER THAN 0.01;
C         IF = OR < 0.01, THEN MUST USE FOLLOWING ROUTINE;
C         I.E., REMOVE THE 'C' COMMENT FROM COL 1 FROM HERE TO LABEL
C         530 & FROM LABEL 503 TO 475
C         NOTE:  ARRAY K2 IS DIMENSIONED (2)
C
C     L1 = 0
C     NGO5 = 1
C     RATIO = .01D0
C     DO 526 I = 1, N
C        IF ( D(I)/SUM - RATIO ) 527, 527, 526
C 527    L1 = L1 + 1
C        K2(L1) = I
C 526 CONTINUE
C     NL = L1
C     IF (NL) 528, 528, 529
C 528 NGO5 = 1
C     GO TO 530
C 529 NGO5 = 2
C 530 CONTINUE
C
      ITER4 = 0
      NGO6 = 1
      RH2 = 1.0D0
C
C         END OF ENTRY INITIALIZATION & CHECKING OF TEMPERATURE
```

```
C
C         BEGINNING OF MAIN PROGRAM LOOP
C
   99 NGO = 1
      ITER2 = 0
      IF (ITER - 1) 473, 473, 474
  474 RH2 = DSQRT(H2)
      RH2 = DMIN1(RH2,1.0D0)
  473 NGO1 = 1
      DO 101 J = 1, M1
         SUM = C(J)
         DO 1002 I = 1, N
            SUM = SUM + XMU(I)*A(I,J)
 1002    CONTINUE
         CP(J) = SUM
  101 CONTINUE
      DO 1012 J = 1, M1
 1012    CPT(J) = CP(J)
      DO 1001 I = 1, N
 1001    DD(I)= D(I)
      L1 = 0
      SUMEX = 0.0D0
      DO 102 J = 1, M
         J1 = J
         IF (CPT(J) + 30.0D0) 420, 420, 421
  421    XJ = DEXP(-CPT(J))
         SUMEX = SUMEX + XJ
         X(J) = XJ
  102 CONTINUE
      IF (SUMEX - 1.0D0 - TOL1*RH2) 103, 103, 107
  103 IF (SUMEX - 1.0D0 + TOL1*RH2) 112, 104, 104
  104 L1 = 1
      DO 106 I = 1, N
         SUM = 0.0
         DO 105 J = 1, M
  105    SUM = SUM + A(I,J)*X(J)
         F(I,1) = SUM
  106 CONTINUE
      YMAX = X0
      GO TO 123
  420 CPT1 = CPT(J1)
      DO 423 J = 1, M
         J1 = J
         CPT2 = CPT(J) - CPT1
         IF (CPT2 + 30.0D0) 420, 420, 422
  422    X(J) = DEXP(-CPT2)
  423 CONTINUE
  107 SUM = 0.0D0
      DO 416 J = 1, M
  416    SUM = SUM + X(J)
      SUM= X0/SUM
      DO 108 J = 1, M
  108    X(J) = X(J)*SUM
      DO 109 I = 1, N
         SUM = DD(I)
```

```
         DO 110 J = 1, M
  110       SUM = SUM - A(I,J)*X(J)
  109 DD(I) = SUM
      GO TO 123
  112 DO 113 J = 1, M
  113    X(J) = 0.0D0
  123 L = L1
C
C         L = 0 OR 1 ONLY
C
      IF (L) 131, 131, 200
  131 DO 132 I = 1, N
  132    XNU(I) = -DD(I)
      GO TO 23
  430 DO 431 J = 1, M1
         CMN(J) = 0.0D0
         DO 431 I = 1, N
  431       CMN(J) = CMN(J) + XNU(I)*A(I,J)
      COMP = TOL5*TOL5/H2
      XNUD = 0.0D0
      DO 432 I = 1, N
  432    XNUD = XNUD + XNU(I)*D(I)
  446 H = -XNUD
      ITTRDG = ITTRDG + 1
      IF (ITTRDG - 5000) 2358, 2358, 2356
 2356 ISENT = 8
      GO TO 5000
 2358 CONTINUE
      DO 433 J = 1, M1
  433    CPT(J) = CP(J) + T*CMN(J)
      EXMIN = 1.0D10
      DO 510 J = 1, M
         IF (CPT(J) - EXMIN) 511, 510, 510
  511    CONTINUE
         EXMIN = CPT(J)
  510 CONTINUE
      SUMEX = 0.0D0
      DO 513 J = 1, M
         IF (CPT(J) - EXMIN - DIF1) 517, 517, 516
  516    X(J) = 0.0D0
         GO TO 513
  517    X(J) = DEXP(EXMIN - CPT(J))
         SUMEX = SUMEX + X(J)
  513 CONTINUE
      IF (EXMIN) 521, 521, 519
  519 IF (SUMEX*DEXP(-EXMIN) - 1.0D0) 443, 521, 521
  521 PROD = X0/SUMEX
      DO 522 J = 1, M
  522    H = H + PROD*X(J)*CMN(J)
  443 CONTINUE
      IF (H) 458, 458, 457
  457 T0 = T
      H0 = H
      GO TO (448,459) , NGO
  448 T = T + T
```

```
      IF (T - 1.0D15) 446, 446, 908
  908 ISENT = 7
      GO TO 5000
  458 T1 = T
      NGO = 2
  459 IF (DABS(H/H2) - TOL3) 453, 453, 460
  460 IF ((T1 - T0)**2 - COMP) 453, 453, 452
  452 T = 0.5D0*(T0 + T1)
      GO TO 446
  453 DO 454 I = 1, N
  454    XMU(I) = XMU(I) + T*XNU(I)
      GO TO  99
C
C         L = 1 AT THIS POINT
C
  200 SDUM = 0.0D0
      SUM = 0.0D0
      DO 2 I = 1, N
         SDUM = SDUM + F(I,1)*F(I,1)
    2 SUM = SUM + F(I,1)*DD(I)
      E(1) = SUM/SDUM
      G(1,1) = 0.0D0
      Y1     = 0.0D0
      Z1     = 0.0D0
   71 SUM = E(1) + G(1,1)*Z1
      IF (SUM) 10, 10, 11
   10 Z1 = 0.0D0
      GO TO 8
   11 IF (SUM - YMAX) 471, 471, 472
  472 Z1 = YMAX
      GO TO 8
  471 Z1 = SUM
    8 IF ( DABS(Z1 - Y1) - TOL3 ) 15, 15, 13
   13 Y1 = Z1
      GO TO 71
   15 DO 16 I = 1, N
   16 XNU(I) = -DD(I) + F(I,1)*Z1
      DO 19 J = 1, M
   19 X(J) = X(J)*Z1
C
C         L = 0 OR 1 BELOW THIS POINT
C
   23 DO 24 I = 1, N
         IF (DABS(XNU(I)) - TOL4*AVD) 24, 24, 25
   24 CONTINUE
      NGO1 = 2
      GO TO 330
   25 GO TO (550,551), NGO6
  551 ITER4 = ITER4 + 1
      IF (L) 550, 550, 814
  814 IF (ITER4 - 20) 550, 552, 552
  552 ITER4 = 0
      NGO1 = 2
      GO TO 330
  550 H0 = 0.0D0
```

```
      DO 26 I = 1, N
   26    HO = HO + XNU(I)**2
      H2 = HO
      TO = 0.0D0
      T = 1.0D0
      GO TO 330
C
C         SEE COMMENTS ABOVE ( LABEL 527 )
C
  503 CONTINUE
C
C     GO TO (475,476), NGO5
C 476 K = ITER - ( ITER/N )*N
C     IF (K) 531, 532, 531
C 532 K = N
C 531 DO 540 L1 = 1, NL
C        IF(K2(L1) - K) 540, 541, 540
C 540 CONTINUE
C     GO TO 475
C 541 IF (DABS(XNU(K)) - TOL5*D(K)) 475, 475, 542
C 542 IF (DABS(XNU(K)) - 1.D-15) 475, 475, 8000
C8000 SUM = XNU(K)
C     DO 543 I = 1, N
C 543    XNU(I)=0.0D0
C     XNU(K) = SUM
C     HO = SUM*SUM
C     DIF1 = 100.0D0
C     H2 = HO
C     TO = 0.0D0
C     T = 1.0D0
C     GO TO 430
C 475 DIF1 = DIF
C
      GO TO 430
  330 ITER = ITER + 1
      IF (ITER - ITW) 333, 670, 670
  670 ISENT = 4
      GO TO 5079
C
  333 GO TO (503,500), NGO1
C
  500 NTOT = N + L
      DO 820 I = 1, N
  820    DD(I) = XMU(I)
      DO 821 J = 1, M1
  821    CPT(J) = X(J)
      IF (L) 727, 727, 502
  502 XBAR = Z1
      ITER2 = 1
  750 IF (XBAR) 727, 727, 701
  701 G(N+1,N+1) = 0.0D0
      DO 704 K = 1, M
         SUM = -C(K)
         DO 705 I = 1, N
  705       SUM = SUM - XMU(I)*A(I,K)
```

```
         IF (SUM - 30.0D0) 425, 727, 727
 425     X(K) = XBAR*DEXP(SUM)
 704 CONTINUE
     DO 706 I = 1, N
         SUM = 0.0D0
         DO 707 K = 1, M
 707        SUM = SUM + A(I,K)*X(K)
         SUM = SUM/XBAR
         G(I,N + 1) = SUM
         G(N + 1,I) = SUM
         R(I) = D(I) - SUM*XBAR
         DO 706 J = 1, N
             SUM = 0.0D0
             DO 709 K = 1, M
 709            SUM = SUM - A(I,K)*A(J,K)*X(K)
         G(I,J)=SUM
 706 CONTINUE
     SUM = -XBAR
     DO 710 K = 1, M
 710     SUM = SUM + X(K)
     R(N + 1) = SUM/XBAR
     DO 728 I = 1, N
         IF (DABS(R(I)) - TOL6*AVD) 728, 728, 727
 728 CONTINUE
     GO TO 730
 727 DO 903 I = 1, N
         IF (DABS(XNU(I)) - TOL5*D(I)) 903, 903, 801
 903 CONTINUE
     DO 822 I = 1, N
 822     XMU(I) = DD(I)
     DO 823 J = 1, M1
 823     X(J) = CPT(J)
     GO TO 5079
 801 TOL4 = 0.1D0*TOL4
     NGO6 = 2
     H2 = 1.0D0
     GO TO 99
 730 IPP = NXNSOL(INXNSL,NTOT,G,R)
     IF (IPP - 2) 8001, 727, 8001
8001 ITER2 = ITER2 + 1
     DO 465 I = 1, N
     IF (DABS(R(I)) - TOL7) 465, 465, 727
 465 CONTINUE
     DO 731 I = 1, NTOT
         IF (DABS(R(I)) - TOL5) 731, 731, 739
 731 CONTINUE
     GO TO 737
 733 DO 734 I = 1, N
 734     XMU(I) = XMU(I) + R(I)
     XBAR = XBAR + R(N + 1)
     GO TO 750
 739 CONTINUE
     IF (ITER2 - ITMAX) 733, 740, 740
 740 ISENT = 5
     GO TO 5000
```

```
C
C         TEST IF X(J) NEGATIVE
C
  737 DO 470 J = 1, M1
         IF (X(J)) 801, 470, 470
  470 CONTINUE
C
C    END OF MAIN PROGRAM LOOP
C
 5079 SUMNI = 0.0D0
      DO 5010 I = 1, M
         SUMNI = SUMNI + X(I)
 5010 CONTINUE
      IF (SUMNI) 751, 752, 751
  752 ISENT = 6
      GO TO 5000
  751 CONTINUE
      DO 5011 I = 1, M
 5011    XMOFR(I) = X(I)/SUMNI
C
C         NOTE: ROUTINE TO CHECK FOR A SINGULAR DERIVATIVE MATRIX
C         IN RTP & RPP ARRAYS HAS BEEN REMOVED.
C
      RETURN
C
C         ERROR: SET MOLE FRACTIONS TO 0.
C
 5000 DO 5100 I = 1, ITOTSP
         XMOFR(I) = 0.
 5100 CONTINUE
C
      RETURN
      END
C_____________________________________________________________________
      FUNCTION NXNSOL(IM,IN,A,B)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(2),B(2)
      INTEGER XROW
      XROW(K000FX,K001FX)=K001FX*M-M+K000FX
      M=IM
      N=IN
      N1=N-1
      DO 44  J=1,N1
      K=J
      J1=J+1
      JJ=XROW(J,J)
      WS1=DABS(A(JJ))
C     LOOP TO FIND LARGEST
      DO 11 L=J1,N
      LJ=XROW(L,J)
      WWS1=DABS(A(LJ))
      IF (WS1-WWS1) 12,11,11
   12 WS1=WWS1
      K=L
```

```
   11 CONTINUE
      IF (J-K) 13,31,31
C                          $ IF DIAG NOT LARGEST INTERCHANGE ROWS
   13 DO 26   L=J,N
      JL=XROW(J,L)
      KL=XROW(K,L)
      WS1=A(JL)
      A(JL)=A(KL)
 26   A(KL)=WS1
      WS1=B(J)
      B(J)=B(K)
      B(K)=WS1
 31   DO 33    L=J1,N
      JL= XROW(J,L)
      IF(A(JJ))33,54,33
 33   A(JL)= A(JL)/A(JJ)
      B(J)=B(J)/A(JJ)
      DO 43    L =1,N
      IF (L -J) 37,43,37
 37   LJ = XROW(L ,J)
 38   DO 41    L2=J1,N
      LL2= XROW(L ,L2)
      JL2= XROW(J,L2)
 41   A(LL2) = A(LL2 )-A(LJ )*A(JL2)
      B(L) = B(L)-A(LJ)*B(J)
 43   CONTINUE
 44   CONTINUE
C       LAST COLUMN HAS NOT BEEN DONE YET
      NN= XROW(N,N)
      IF (A(NN)) 46,54,46
   46 B(N)= B(N)/A(NN)
      DO 50 L=1,N1
      LN= XROW(L,N)
   50 B(L)= B(L)- A(LN)*B(N)
      NXNSOL=1
      RETURN
   54 NXNSOL=2
      RETURN
      END
```

```
  *****    *********   *******              *********  *      **  ********
 *******   *********  *********             *********  **     **  *********
**     **  **    **   **     **             **         ***    **  **     **
**     **       **    **     **             **         ****   **  **     **
**     **      ****   *********   *******   ******     ** **  **  **     **
**     **     ******   ********   *******   ******     **  ** **  **     **
**     **         **         **             **         **   ****  **     **
**     **  **     **  **     **             **         **    ***  **     **
 *******   *********  *********             *********  **     **  *********
  *****     *******    *******              *********  **      *  ********
```

NORMAN	NORMAN	USERID	ORIGIN
BIN-0296	MITVMA	DISTCODE	SYSTEM
PROGRAM	CODE	FILENAME	FILETYPE
03/11/83	12:13:26	FILE CREATION DATE	
7362	A	SPOOLID	CLASS
03/11/83	12:15:25	FILE PRINT DATE	
NX1	003	FORM	DEVICE

PRINT RATE: FIRST COPY IS 6.5 CENTS/PAGE, SUBSEQUENT COPIES ARE 3.0 CENTS/PAGE

PRINT COST: $9.22 FOR 2 COPIES OF 97 PAGE REPORT

```
*********  *      **  ********              *****    *********   *******
*********  **     **  *********            *******   *********  *********
**         ***    **  **     **           **     **  **    **   **     **
**         ****   **  **     **           **     **       **    **     **
******     ** **  **  **     **   *******  **     **      ****   *********
******     **  ** **  **     **   *******  **     **     ******   ********
**         **   ****  **     **           **     **         **         **
**         **    ***  **     **           **     **  **     **   **     **
*********  **     **  *********            *******   *********  *********
*********  **      *  ********              *****     *******    *******
```

XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX

DATE: 11 MAR 83 AT 12:16:35

DEPARTMENT: NODEPT

JOB ID: 221 REPORT NO. 4

FILE ID:

INPUT PROCESSING TIME: 00:00:36

OUTPUT PROCESSING TIME: 00:04:58

REPORT COMPLETION CODE: 50

PAGES TO BIN: 197

PAGES TO TRAY: 0

PAPER PATH HOLES: 0

LINES PRINTED: 10930

ONLINE IDLE (SEC): 0

BLOCKS READ: 0

BLOCKS SKIPPED: 0

RECORDS READ: 5607

DJDE RECORDS READ: 6

MAXIMUM COPY COUNT: 2

OVERPRINTS: 0

COLLATE: YES

SF/MF: MULTI

SIMPLEX/DUPLEX: SIMPLEX

JDE,JDL USED: NX1,DFAULT

ACCTINFO:

INITIAL FONT LIST: MIT001

INITIAL FORM LIST: -NONE

INITIAL CME LIST: -NONE

XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX
XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX 9700 XEROX

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SANDIA MATHEMATICAL PROGRAM LIBRARY
C     APPLIED MATHEMATICS DIVISION 2613
C     SANDIA LABORATORIES
C     ALBUQUERQUE, NEW MEXICO  87115
C     CONTROL DATA 6600/7600 VERSION 7.2  SEPTEMBER 1977
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *                 ISSUED BY SANDIA LABORATORIES,                  *
C   *                   A PRIME CONTRACTOR TO THE                     *
C   *   UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION  *
C   *                                                                 *
C   * * * * * * * * * * * * * *  NOTICE   * * * * * * * * * * * * * * *
C   *
C   * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE
C   * UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE
C   * UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,
C   * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,
C   * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS
C   * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY
C   * FOR THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,
C   * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS
C   * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.
C   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   * THE PRIMARY DOCUMENT FOR THE LIBRARY OF WHICH THIS ROUTINE IS   *
C   * A PART IS SAND75-0545.                                          *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ERRCHK(N,M)
      INTEGER*2 M
      DIMENSION M(100)
      NWDS = (IABS(N)+1)/2
      PRINT 10, (M(I), I=1,NWDS)
   10 FORMAT(1H0, 60A2)
      IF (N .GT. 0) RETURN
      STOP
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SANDIA MATHEMATICAL PROGRAM LIBRARY
C     APPLIED MATHEMATICS DIVISION 2613
C     SANDIA LABORATORIES
C     ALBUQUERQUE, NEW MEXICO  87115
C     CONTROL DATA 6600/7600 VERSION 7.2  SEPTEMBER 1977
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   *                 ISSUED BY SANDIA LABORATORIES,                  *
C   *                   A PRIME CONTRACTOR TO THE                     *
C   *   UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION  *
```

```
C   *                                                                *
C   * * * * * * * * * * * * * *  NOTICE   * * * * * * * * * * * * * * *
C   *
C   * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE
C   * UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE
C   * UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,
C   * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,
C   * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS
C   * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY
C   * FOR THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,
C   * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS
C   * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.
C   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                *
C   * THE PRIMARY DOCUMENT FOR THE LIBRARY OF WHICH THIS ROUTINE IS   *
C   * A PART IS SAND75-0545.                                         *
C   *                                                                *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   WRITTEN BY L. F. SHAMPINE AND M. K. GORDON
C
C   ABSTRACT
C
C
C   THE METHODS IN SUBROUTINE  STEP1  APPROXIMATE THE SOLUTION NEAR  X
C   BY A POLYNOMIAL.  SUBROUTINE  INTRP  APPROXIMATES THE SOLUTION AT
C   XOUT  BY EVALUATING THE POLYNOMIAL THERE.  INFORMATION DEFINING THIS
C   POLYNOMIAL IS PASSED FROM  STEP1  SO  INTRP  CANNOT BE USED ALONE.
C
C   THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS, THE INITIAL
C   VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C   FURTHER DETAILS ON USE OF THIS CODE ARE AVAILABLE IN *SOLVING
C   ORDINARY DIFFERENTIAL EQUATIONS WITH ODE, STEP, AND INTRP*,
C   BY L. F. SHAMPINE AND M. K. GORDON, SLA-73-1060.
C
C   INPUT TO INTRP --
C
C   THE USER PROVIDES STORAGE IN THE CALLING PROGRAM FOR THE ARRAYS IN
C   THE CALL LIST
C      DIMENSION Y(NEQN),YOUT(NEQN),YPOUT(NEQN),PHI(NEQN,16),PSI(12)
C   AND DEFINES
C      XOUT -- POINT AT WHICH SOLUTION IS DESIRED.
C   THE REMAINING PARAMETERS ARE DEFINED IN  STEP1  AND PASSED TO
C   INTRP  FROM THAT SUBROUTINE
C
C   OUTPUT FROM  INTRP --
C
C      YOUT(*) -- SOLUTION AT  XOUT
C      YPOUT(*) -- DERIVATIVE OF SOLUTION AT  XOUT
C   THE REMAINING PARAMETERS ARE RETURNED UNALTERED FROM THEIR INPUT
C   VALUES.  INTEGRATION WITH  STEP1  MAY BE CONTINUED.
C
      SUBROUTINE INTRP(X,Y,XOUT,YOUT,YPOUT,NEQN,KOLD,PHI,PSI)
```

```
      IMPLICIT REAL *8 (A-H,O-Z)
C
C
      GENERIC
      DIMENSION Y(25),YOUT(25),YPOUT(25),PHI(25,16),PSI(12)
      DIMENSION G(13),W(13),RHO(13)
      DATA G(1)/1.0/,RHO(1)/1.0/
C
      HI = XOUT - X
      KI = KOLD + 1
      KIP1 = KI + 1
C
C   INITIALIZE W(*) FOR COMPUTING G(*)
C
      DO 5 I = 1,KI
        TEMP1 = I
 5      W(I) = 1.0/TEMP1
      TERM = 0.0
C
C   COMPUTE G(*)
C
      DO 15 J = 2,KI
        JM1 = J - 1
        PSIJM1 = PSI(JM1)
        GAMMA = (HI + TERM)/PSIJM1
        ETA = HI/PSIJM1
        LIMIT1 = KIP1 - J
        DO 10 I = 1,LIMIT1
 10       W(I) = GAMMA*W(I) - ETA*W(I+1)
        G(J) = W(1)
        RHO(J) = GAMMA*RHO(JM1)
 15     TERM = PSIJM1
C
C   INTERPOLATE
C
      DO 20 L = 1,NEQN
        YPOUT(L) = 0.0
 20     YOUT(L) = 0.0
      DO 30 J = 1,KI
        I = KIP1 - J
        TEMP2 = G(I)
        TEMP3 = RHO(I)
        DO 25 L = 1,NEQN
          YOUT(L) = YOUT(L) + TEMP2*PHI(L,I)
 25       YPOUT(L) = YPOUT(L) + TEMP3*PHI(L,I)
 30     CONTINUE
      DO 35 L = 1,NEQN
 35     YOUT(L) = Y(L) + HI*YOUT(L)
      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SANDIA MATHEMATICAL PROGRAM LIBRARY
C     APPLIED MATHEMATICS DIVISION 2613
C     SANDIA LABORATORIES
```

```
C     ALBUQUERQUE, NEW MEXICO  87115
C     CONTROL DATA 6600/7600 VERSION 7.2  SEPTEMBER 1977
C
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  *                                                                   *
C  *                 ISSUED BY SANDIA LABORATORIES                     *
C  *                   A PRIME CONTRACTOR TO THE                       *
C  *                UNITED STATES DEPARTMENT OF ENERGY                 *
C  *                                                                   *
C  * * * * * * * * * * * * * * * NOTICE  * * * * * * * * * * * * * * * *
C  *                                                                   *
C  * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE   *
C  * UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE      *
C  * UNITED STATES DEPARTMENT OF ENERGY NOR ANY OF THEIR EMPLOYEES,    *
C  * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR EMPLOYEES  *
C  * MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL      *
C  * LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS OR     *
C  * USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT OR PROCESS      *
C  * DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE          *
C  * OWNED RIGHTS.                                                     *
C  *                                                                   *
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  *                                                                   *
C  * THE PRIMARY DOCUMENT FOR THE LIBRARY OF WHICH THIS ROUTINE IS     *
C  * PART IS SAND77-1441.                                              *
C  *                                                                   *
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C   WRITTEN BY M. K. GORDON, 5122
C
C***********************************************************************
C   ABSTRACT
C***********************************************************************
C   SUBROUTINE  ODERT  INTEGRATES A SYSTEM OF  NEQN  FIRST ORDER
C   ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
C             DY(I)/DT = F(T,Y(1),...,Y(NEQN))
C             Y(I) GIVEN AT T.
C   THE SUBROUTINE INTEGRATES FROM  T  IN THE DIRECTION OF  TOUT  UNTIL
C   IT LOCATES THE FIRST ROOT OF THE NONLINEAR EQUATION
C         G(T,Y(1),...,Y(NEQN),YP(1),...,YP(NEQN)) = 0.
C   UPON FINDING THE ROOT, THE CODE RETURNS WITH ALL PARAMETERS IN THE
C   CALL LIST SET FOR CONTINUING THE INTEGRATION TO THE NEXT ROOT OR
C   THE FIRST ROOT OF A NEW FUNCTION  G  .  IF NO ROOT IS FOUND, THE
C   INTEGRATION PROCEEDS TO  TOUT .  AGAIN ALL PARAMETERS ARE SET TO
C   CONTINUE.
C
C   THE DIFFERENTIAL EQUATIONS ARE ACTUALLY SOLVED BY A SUITE OF CODES,
C   DERT1 ,STEP1 , AND INTRP .  ODERT ALLOCATES VIRTUAL STORAGE IN
C   THE WORK ARRAYS  WORK  AND  IWORK  AND CALLS  DERT1 .  DERT1  IS A
C   SUPERVISOR WHICH DIRECTS THE INTEGRATION.  IT CALLS ON  STEP1  TO
C   ADVANCE THE SOLUTION AND  INTRP  TO INTERPOLATE THE SOLUTION AND
C   ITS DERIVATIVE.  STEP1  USES A MODIFIED DIVIDED DIFFERENCE FORM OF
C   THE ADAMS PECE FORMULAS AND LOCAL EXTRAPOLATION.  IT ADJUSTS THE
C   ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A
```

```
C   GENERALIZED SENSE.  NORMALLY EACH CALL TO  STEP1  ADVANCES THE
C   SOLUTION ONE STEP IN THE DIRECTION OF  TOUT .  FOR REASONS OF
C   EFFICIENCY  ODERT  INTEGRATES BEYOND  TOUT  INTERNALLY, THOUGH
C   NEVER BEYOND T+10*(TOUT-T), AND CALLS  INTRP  TO INTERPOLATE THE
C   SOLUTION AND DERIVATIVE AT  TOUT .  AN OPTION IS PROVIDED TO STOP
C   THE INTEGRATION AT  TOUT  BUT IT SHOULD BE USED ONLY IF IT IS
C   IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
C
C   AFTER EACH INTERNAL STEP,  DERT1  EVALUATES THE FUNCTION  G  AND
C   CHECKS FOR A CHANGE IN SIGN IN THE FUNCTION VALUE FROM THE
C   PRECEDING STEP.  SUCH A CHANGE INDICATES A ROOT LIES IN THE
C   INTERVAL OF THE STEP JUST COMPLETED.  DERT1 THEN CALLS SUBROUTINE
C   ROOT  TO REDUCE THE BRACKETING INTERVAL UNTIL THE ROOT IS
C   DETERMINED TO THE DESIRED ACCURACY.  SUBROUTINE  ROOT  USES A
C   COMBINATION OF THE SECANT RULE AND BISECTION TO DO THIS.  THE
C   SOLUTION AND DERIVATIVE VALUES REQUIRED ARE OBTAINED BY
C   INTERPOLATION WITH  INTRP .  THE CODE LOCATES ONLY THOSE ROOTS
C   FOR WHICH  G  CHANGES SIGN IN  (T,TOUT)  AND FOR WHICH A
C   BRACKETING INTERVAL EXISTS.  IN PARTICULAR, IT WILL NOT DETECT A
C   ROOT AT THE INITIAL POINT  T .
C
C   THE CODES  STEP1 , INTRP , ROOT , AND THAT PORTION OF  DERT1
C   WHICH DIRECTS THE INTEGRATION ARE EXPLAINED AND DOCUMENTED IN THE
C   TEXT, COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS, THE
C   INITIAL VALUE PROBLEM, BY L. F. SHAMPINE AND M. K. GORDON.
C
C   DETAILS OF THE USE OF ODERT ARE GIVEN IN SAND-75-0211.
C
C***********************************************************************
C   THE PARAMETERS FOR  ODERT  ARE
C***********************************************************************
C      F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      Y(*) -- SOLUTION VECTOR AT  T
C      T -- INDEPENDENT VARIABLE
C      TOUT -- ARBITRARY POINT BEYOND THE ROOT DESIRED
C      RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
C           ERROR TEST.  AT EACH STEP THE CODE REQUIRES
C             ABS(LOCAL ERROR) .LE. ABS(Y)*RELERR + ABSERR
C           FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
C      IFLAG -- INDICATES STATUS OF INTEGRATION
C      WORK,IWORK -- ARRAYS TO HOLD INFORMATION INTERNAL TO THE CODE
C           WHICH IS NECESSARY FOR SUBSEQUENT CALLS
C      G - FUNCTION OF T, Y(*), YP(*) WHOSE ROOT IS DESIRED.
C      REROOT, AEROOT -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR
C           ACCEPTING THE ROOT.  THE INTERVAL CONTAINING THE ROOT IS
C           REDUCED UNTIL IT SATISFIES
C            0.5*ABS(LENGTH OF INTERVAL) .LE. REROOT*ABS(ROOT)+AEROOT
C           WHERE ROOT IS THAT ENDPOINT YIELDING THE SMALLER VALUE OF
C           G  IN MAGNITUDE.  PURE RELATIVE ERROR IS NOT RECOMMENDED
C           IF THE ROOT MIGHT BE ZERO.
C***********************************************************************
C   FIRST CALL TO  ODERT --
C***********************************************************************
C   THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE
```

```
C   ARRAYS IN THE CALL LIST,
C              Y(NEQN), WORK(100+21*NEQN), IWORK(5)
C   AND DECLARE  F , G  IN AN EXTERNAL STATEMENT.  HE MUST SUPPLY THE
C   SUBROUTINE  F(T,Y,YP)  TO EVALUATE
C           DY(I)/DT = YP(I) = F(T,Y(1),...,Y(NEQN))
C   AND THE FUNCTION  G(T,Y,YP)  TO EVALUATE
C           G = G(T,Y(1),...,Y(NEQN),YP(1),...,YP(NEQN)).
C   NOTE THAT THE ARRAY YP IS AN INPUT ARGUMENT AND SHOULD NOT BE
C   COMPUTED IN THE FUNCTION SUBPROGRAM.  FINALLY THE USER MUST
C   INITIALIZE THE PARAMETERS
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      Y(*) -- VECTOR OF INITIAL CONDITIONS
C      T -- STARTING POINT OF INTEGRATION
C      TOUT -- ARBITRARY POINT BEYOND THE ROOT DESIRED
C      RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
C                       FOR INTEGRATING THE EQUATIONS
C      IFLAG -- +1,-1.  INDICATOR TO INITIALIZE THE CODE.  NORMAL INPUT
C           IS +1.  THE USER SHOULD SET IFLAG=-1 ONLY IF IT IS
C           IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
C      REROOT,AEROOT -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR
C                       COMPUTING THE ROOT OF  G
C
C   ALL PARAMETERS EXCEPT F, G, NEQN, TOUT, REROOT AND AEROOT MAY BE
C   ALTERED BY THE CODE ON OUTPUT SO MUST BE VARIABLES IN THE CALLING
C   PROGRAM.
C*********************************************************************
C   OUTPUT FROM  ODERT --
C*********************************************************************
C      NEQN -- UNCHANGED
C      Y(*) -- SOLUTION AT  T
C      T -- LAST POINT REACHED IN INTEGRATION.  NORMAL RETURN HAS
C           T = TOUT OR T = ROOT
C      TOUT -- UNCHANGED
C      RELERR,ABSERR -- NORMAL RETURN HAS TOLERANCES UNCHANGED.  IFLAG=3
C           SIGNALS TOLERANCES INCREASED
C      IFLAG = 2 -- NORMAL RETURN.  INTEGRATION REACHED  TOUT
C            = 3 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE ERROR
C                   TOLERANCES TOO SMALL.  RELERR ,  ABSERR  INCREASED
C                   APPROPRIATELY FOR CONTINUING
C            = 4 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE MORE THAN
C                   500 STEPS NEEDED
C            = 5 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE EQUATIONS
C                   APPEAR TO BE STIFF
C            = 6 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE SOLUTION
C                   VANISHED MAKING PURE RELATIVE ERROR IMPOSSIBLE.
C                   MUST USE NON-ZERO  ABSERR  TO CONTINUE
C            = 7 -- INVALID INPUT PARAMETERS (FATAL ERROR)
C            = 8 -- NORMAL RETURN.  A ROOT WAS FOUND WHICH SATISFIED
C                   THE ERROR CRITERION OR HAD A ZERO RESIDUAL
C            = 9 -- ABNORMAL RETURN.  AN ODD ORDER POLE OF  G  WAS
C                   FOUND.
C            =10 -- ABNORMAL RETURN.  TOO MANY EVALUATIONS OF  G  WERE
C                   REQUIRED (AS PROGRAMMED 500 ARE ALLOWED.)
C           THE VALUE OF  IFLAG  IS RETURNED NEGATIVE WHEN THE INPUT
C           VALUE IS NEGATIVE AND THE INTEGRATION DOES NOT REACH
```

```
C           TOUT , I.E., -3,...,-6,-8,-9,-10.
C      WORK(*),IWORK(*) -- INFORMATION GENERALLY OF NO INTEREST TO THE
C                          USER BUT NECESSARY FOR SUBSEQUENT CALLS
C      REROOT,AEROOT -- UNCHANGED
C*********************************************************************
C   SUBSEQUENT CALLS TO  ODERT --
C*********************************************************************
C   SUBROUTINE  ODERT  RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
C   THE INTEGRATION.  IF THE INTEGRATION DID NOT REACH  TOUT  AND THE
C   USER WANTS TO CONTINUE, HE JUST CALLS AGAIN.  IF THE INTEGRATION
C   REACHED  TOUT , THE USER NEED ONLY DEFINE A NEW  TOUT  AND CALL
C   AGAIN.  THE OUTPUT VALUE OF  IFLAG  IS THE APPROPRIATE INPUT VALUE
C   FOR SUBSEQUENT CALLS.  THE ONLY SITUATION IN WHICH IT SHOULD BE
C   ALTERED IS TO STOP THE INTEGRATION INTERNALLY AT THE NEW  TOUT ,
C   I.E., CHANGE OUTPUT  IFLAG=2  TO INPUT  IFLAG=-2 .  ONLY THE ERROR
C   TOLERANCES AND THE FUNCTION  G  MAY BE CHANGED BY THE USER BEFORE
C   CONTINUING.  ALL OTHER PARAMETERS MUST REMAIN UNCHANGED.  A NEW
C   FUNCTION  G  IS DETECTED AUTOMATICALLY.
C
      SUBROUTINE ODERT(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK,
     1  G,REROOT,AEROOT)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
      GENERIC
      LOGICAL START,PHASE1,NORND
      DIMENSION Y(25),WORK(625),IWORK(5)
      EXTERNAL F,G
      DATA IALPHA,IBETA,ISIG,IV,IW,IGG,IPHASE,IPSI,IX,IH,IHOLD,ISTART,
     1  ITOLD,IDELSN,IGX,ITROOT/1,13,25,38,50,62,75,76,88,89,90,91,
     2  92,93,94,95/
      IYY = 100
      IWT = IYY + NEQN
      IP = IWT + NEQN
      IYP = IP + NEQN
      IYPOUT = IYP + NEQN
      IPHI = IYPOUT + NEQN
      IF(IABS(IFLAG) .EQ. 1) GO TO 1
      START = WORK(ISTART) .GT. 0.0
      PHASE1 = WORK(IPHASE) .GT. 0.0
      NORND = IWORK(2) .NE. -1
1     CALL DERT1(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,G,REROOT,AEROOT,
     1  WORK(IYY),WORK(IWT),WORK(IP),WORK(IYP),WORK(IYPOUT),WORK(IPHI),
     2  WORK(IALPHA),WORK(IBETA),WORK(ISIG),WORK(IV),WORK(IW),WORK(IGG),
     3  PHASE1,WORK(IPSI),WORK(IX),WORK(IH),WORK(IHOLD),START,
     4  WORK(ITOLD),WORK(IDELSN),WORK(IGX),WORK(ITROOT),IWORK(1),
     5  NORND,IWORK(3),IWORK(4),IWORK(5))
      WORK(ISTART) = -1.0
      IF(START) WORK(ISTART) = 1.0
      WORK(IPHASE) = -1.0
      IF(PHASE1) WORK(IPHASE) = 1.0
      IWORK(2) = -1
      IF(NORND) IWORK(2) = 1
      RETURN
      END
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE DERT1(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,G,REROOT,
     1  AEROOT,YY,WT,P,YP,YPOUT,PHI,ALPHA,BETA,SIG,V,W,GG,PHASE1,PSI,
     2  X,H,HOLD,START,TOLD,DELSGN,GX,TROOT,NS,NORND,K,KOLD,ISNOLD)
C ***NAME CHANGED FROM DERT TO DERT1 TO AVOID A NAMING CONFLICT.
C
C   ODERT  MERELY ALLOCATES STORAGE FOR  DERT  TO RELIEVE THE USER OF
C   THE INCONVENIENCE OF A LONG CALL LIST.  CONSEQUENTLY  DERT  IS USED
C   AS DESCRIBED IN THE COMMENTS FOR  ODERT .
C
C   THE CODES STEP, INTRP AND ROOT AND THAT PORTION OF DERT DIRECTING
C   THE INTEGRATION ARE COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS, THE INITIAL
C   VALUE PROBLEM BY L. F. SHAMPINE AND M. K. GORDON.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      GENERIC
      LOGICAL STIFF,CRASH,START,PHASE1,NORND
      DIMENSION Y(25),YY(25),WT(25),PHI(25,16),P(25),YP(25),
     1  YPOUT(25),PSI(12),ALPHA(12),BETA(12),SIG(13),V(12),W(12),
     2  GG(13)
      COMMON/MLDRT/SPACE(10)
      EXTERNAL F,G
C
C*********************************************************************
C*  THE ONLY MACHINE DEPENDENT CONSTANT IS BASED ON THE MACHINE UNIT  *
C*  ROUNDOFF ERROR  U  WHICH IS THE SMALLEST POSITIVE NUMBER SUCH THAT *
C*  1.0+U .GT. 1.0 .  U  MUST BE CALCULATED AND  FOURU=4.0*U  INSERTED *
C*  IN THE FOLLOWING STATEMENT BEFORE USING  ODERT .  THE SUBROUTINE  *
C*  MACHIN  CALCULATES  U .  FOURU  AND  TWOU=2.0*U  MUST ALSO BE     *
C*  INSERTED IN SUBROUTINE  STEP  BEFORE CALLING  ODERT .             *
C*********************************************************************
      DATA FOURU/8.8E-16/
C*********************************************************************
C
C   THE CONSTANT  MAXNUM  IS THE MAXIMUM NUMBER OF STEPS ALLOWED IN ONE
C   CALL TO  ODERT .  THE USER MAY CHANGE THIS LIMIT BY ALTERING THE
C   FOLLOWING STATEMENT
      DATA MAXNUM/500/
C
C            ***            ***            ***
C   TEST FOR IMPROPER PARAMETERS
C
      IF(IABS(IFLAG) .EQ. 7) CALL ERRCHK(-31,
     1  31HIN ODERT, ENTERED WITH IFLAG=7.)
      IF(NEQN .LT. 1) CALL ERRCHK(32,
     1  32HIN ODERT, NEQN MUST BE POSITIVE.)
      IF(NEQN .LT. 1) GO TO 10
      IF(T .EQ. TOUT) CALL ERRCHK(61,
     1 61HIN ODERT, ENDPOINTS OF INTEGRATION INTERVAL MUST BE DISTINCT.)
      IF(T .EQ. TOUT) GO TO 10
      IF(RELERR .LT. 0.0  .OR.  ABSERR .LT. 0.0)  CALL ERRCHK(49,
     1  49HIN ODERT, RELERR AND ABSERR MUST BE NON-NEGATIVE.)
```

```
      IF(RELERR .LT. 0.0  .OR.  ABSERR .LT. 0.0) GO TO 10
      EPS = MAX(RELERR,ABSERR)
      IF(EPS .LE. 0.0) CALL ERRCHK(51,
     1  51HIN ODERT, EITHER RELERR OR ABSERR MUST BE POSITIVE.)
      IF(EPS .LE. 0.0) GO TO 10
      IF(REROOT .LT. 0.0  .OR.  AEROOT .LT. 0.0) CALL ERRCHK(49,
     1  49HIN ODERT, REROOT AND AEROOT MUST BE NON-NEGATIVE.)
      IF(REROOT .LT. 0.0  .OR.  AEROOT .LT. 0.0) GO TO 10
      IF(REROOT+AEROOT .LE. 0.0) CALL ERRCHK(51,
     1  51HIN ODERT, EITHER REROOT OR AEROOT MUST BE POSITIVE.)
      IF(REROOT+AEROOT .LE. 0.0) GO TO 10
      IF(IFLAG .EQ. 0) CALL ERRCHK(34,
     1  34HIN ODERT, INVALID INPUT FOR IFLAG.)
      IF(IFLAG .EQ. 0) GO TO 10
      ISN = ISIGN(1,IFLAG)
      IFLAG = IABS(IFLAG)
      IF(IFLAG .EQ. 1) GO TO 20
      IF(T .NE. TOLD) CALL ERRCHK(68,
     1  68HIN ODERT, INPUT VALUE OF T MUST BE OUTPUT VALUE FROM PRECEDIN
     2G CALL.)
      IF(T .NE. TOLD) GO TO 10
      IF(IFLAG .GE. 2  .AND.  IFLAG .LE. 6) GO TO 15
      IF(IFLAG .GE. 8  .AND.  IFLAG .LE. 10) GO TO 15
      CALL ERRCHK(-34,34HIN ODERT, INVALID INPUT FOR IFLAG.)
   10 IFLAG = 7
      RETURN
C
15    CONTINUE
      IF (ISNOLD.LT.0 .OR. DELSGN*(TOUT-T).LT.0.) GO TO 20
C--      EVALUATE  G  AT EITHER TOUT (OUTPUT POINT THIS CALL)  OR AT
C--         X (POINT TO WHICH INTERNAL INTEGRATION HAS ALREADY
C--         PROCEEDED), WHICHEVER OCCURS FIRST.
      T2=X
      IF((X-T.GT.0..AND.X-TOUT.GT.0.).OR.(X-T.LT.0..AND.X-TOUT.LT.0.))
     1            T2=TOUT
      CALL INTRP(X,YY,T2,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      GOFT2=G(T2,Y,YPOUT)
C--   NOW EVALUATE AT T1=T
      T1=T
      CALL INTRP(X,YY,T1,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      GOFT1=G(T1,Y,YPOUT)
C--      NOW SEE IF A ROOT OF  G  OCCURS IN CLOSED INTERVAL (T1,T2).
      IF( GOFT1.EQ.0. .OR. GOFT2.EQ.0.)              GO TO 134
      IF( SIGN(1.D0,GOFT1)  * SIGN(1.D0,GOFT2) .LT. 0.D0 ) GO TO 134
                GO TO 21
C
C   ON EACH CALL SET INTERVAL OF INTEGRATION AND COUNTER FOR NUMBER OF
C   STEPS.  ADJUST INPUT ERROR TOLERANCES TO DEFINE WEIGHT VECTOR FOR
C   SUBROUTINE  STEP
C
20    T2=T
      CALL F(T2,Y,YPOUT)
      GOFT2 = G(T2,Y,YPOUT)
   21 CONTINUE
      DEL = TOUT - T
```

```
      ABSDEL = ABS(DEL)
      TEND = T + 10.0*DEL
      IF(ISN .LT. 0) TEND = TOUT
      NOSTEP = 0
      KLE4 = 0
      STIFF = .FALSE.
      RELEPS = RELERR/EPS
      ABSEPS = ABSERR/EPS
      IF(IFLAG .EQ. 1) GO TO 30
      IF(ISNOLD .LT. 0) GO TO 30
      IF(DELSGN*DEL .GT. 0.0) GO TO 50
C
C   ON START AND RESTART ALSO SET WORK VARIABLES X AND YY(*), STORE THE
C   DIRECTION OF INTEGRATION, AND INITIALIZE THE STEP SIZE.
C
   30 START = .TRUE.
      X = T
      TROOT = T
      DO 40 L = 1,NEQN
   40   YY(L) = Y(L)
      DELSGN = SIGN(1.0D0,DEL)
      H = SIGN(MAX(ABS(TOUT-X),FOURU*ABS(X)),TOUT-X)
C
C   IF ALREADY PAST OUTPUT POINT, INTERPOLATE AND RETURN
C
50    CONTINUE
      IF(ABS(X-T) .LT. ABSDEL) GO TO 60
      CALL INTRP(X,YY,TOUT,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      IFLAG = 2
      T = TOUT
      TOLD = T
      ISNOLD = ISN
      RETURN
C
C   IF CANNOT GO PAST OUTPUT POINT AND SUFFICIENTLY CLOSE,
C   EXTRAPOLATE AND RETURN
C
   60 IF(ISN .GT. 0  .OR.  ABS(TOUT-X) .GE. FOURU*ABS(X)) GO TO 80
      H = TOUT - X
      CALL F(X,YY,YP)
      DO 70 L = 1,NEQN
   70   Y(L) = YY(L) + H*YP(L)
C *** NEXT STMT ADDED BY LIENESCH TO ENSURE YPOUT VALUES WILL ALWAYS BE
C *** AVAILABLE UNDER ANY CIRCUMSTANCES
      CALL F(X,Y,YPOUT)
      IFLAG = 2
      T = TOUT
      TOLD = T
      ISNOLD = ISN
      RETURN
C
C   TEST FOR TOO MUCH WORK
C
   80 IF(NOSTEP .LT. MAXNUM) GO TO 100
      IFLAG = ISN*4
```

```
      IF(STIFF) IFLAG = ISN*5
      DO 90 L = 1,NEQN
   90   Y(L) = YY(L)
      T = X
      TOLD = T
      ISNOLD = 1
      RETURN
C
C   LIMIT STEP SIZE, SET WEIGHT VECTOR AND TAKE A STEP
C
  100 H = SIGN(MIN(ABS(H),ABS(TEND-X)),H)
      DO 110 L = 1,NEQN
        WT(L) = RELEPS*ABS(YY(L)) + ABSEPS
        IF(WT(L) .LE. 0.0) GO TO 160
  110 CONTINUE
      CALL STEP1(F,NEQN,YY,X,H,EPS,WT,START,
     1  HOLD,K,KOLD,CRASH,PHI,P,YP,PSI,
     2  ALPHA,BETA,SIG,V,W,GG,PHASE1,NS,NORND)
C
C   TEST FOR TOLERANCES TOO SMALL.  IF SO, SET THE DERIVATIVE AT X
C   BEFORE RETURNING
C
      IF(.NOT.CRASH) GO TO 130
      IFLAG = ISN*3
      RELERR = EPS*RELEPS
      ABSERR = EPS*ABSEPS
      DO 120 L = 1,NEQN
        YP(L) = PHI(L,1)
  120   Y(L) = YY(L)
      T = X
      TOLD = T
      ISNOLD = 1
      RETURN
C
C   AUGMENT COUNTER ON WORK AND TEST FOR STIFFNESS.  ALSO TEST FOR A
C   ROOT IN THE STEP JUST COMPLETED
C
  130 NOSTEP = NOSTEP + 1
      KLE4 = KLE4 + 1
      IF(KOLD .GT. 4) KLE4 = 0
      IF(KLE4 .GE. 50) STIFF = .TRUE.
      T1=T2
      GOFT1=GOFT2
      T2=TOUT
C--     EVALUATE G AT INTERNAL INTEGRATION POINT X UNLESS X IS PAST TOUT
C--     IF X IS PAST TOUT EVALUATE G AT TOUT.
      IF( ABS(X-T).LT.ABSDEL) T2=X
      CALL INTRP(X,YY,T2,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      GOFT2=G(T2,Y,YPOUT)
      IF(GOFT1.EQ.0. .OR. GOFT2.EQ.0.) GO TO 134
      IF( SIGN(1.D0,GOFT1)*SIGN(1.0D0,GOFT2) .LT.0.D0)GO TO 134
      GO TO 50
C
C   LOCATE ROOT OF G.  INTERPOLATE WITH  INTRP  FOR SOLUTION AND
C   DERIVATIVE VALUES
```

```
C
134   JFLAG=1
C--      HERE ROOT IS BETWEEN  T1 AND T2
      B=T1
      IF(GOFT1.EQ.0.)GO TO 150
      B=T2
      IF(GOFT2.EQ.0.)GO TO 150
      C=T1
  140 CALL ROOT(T,GT,B,C,REROOT,AEROOT,JFLAG)
      IF(JFLAG .GT. 0) GO TO 150
        IF( T.EQ.T1)GT=GOFT1
        IF( T.EQ.T2)GT=GOFT2
        IF( T.EQ.T1 .OR.T.EQ.T2)GO TO 140
      CALL INTRP(X,YY,T,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      GT = G(T,Y,YPOUT)
      GO TO 140
  150 CONTINUE
      IFLAG = JFLAG+7
      IF(JFLAG .EQ. 2  .OR.  JFLAG .EQ. 4) IFLAG = 8
      IF(JFLAG .EQ. 3) IFLAG = 9
      IF(JFLAG .EQ. 5) IFLAG = 10
      IFLAG = IFLAG*ISN
      CALL INTRP(X,YY,B,Y,YPOUT,NEQN,KOLD,PHI,PSI)
      T = B
      IF(ABS(T-TROOT) .LE. REROOT*ABS(T) + AEROOT) GO TO 50
      TROOT = T
      TOLD = T
      ISNOLD = 1
      RETURN
  160 CALL ERRCHK(72,72HIN ODERT, PURE ABSOLUTE ERROR IMPOSSIBLE.  USE N
     1ON-ZERO VALUE OF ABSERR.)
      IFLAG = 6
      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C  ROOT COMPUTES A ROOT OF THE NONLINEAR EQUATION F(X)=0
C  WHERE F(X) IS A CONTINUOUS REAL FUNCTION OF A SINGLE REAL
C  VARIABLE X.  THE METHOD USED IS A COMBINATION OF BISECTION
C  AND THE SECANT RULE.
C
C  NORMAL INPUT CONSISTS OF A CONTINUOUS FUNCTION F AND AN
C  INTERVAL (B,C) SUCH THAT F(B)*F(C).LE.0.0.  EACH ITERATION
C  FINDS NEW VALUES OF B AND C SUCH THAT THE INTERVAL (B,C) IS
C  SHRUNK AND F(B)*F(C).LE.0.0.  THE STOPPING CRITERION IS
C
C          ABS(B-C).LE.2.0*(RELERR*ABS(B)+ABSERR)
C
C  WHERE RELERR=RELATIVE ERROR AND ABSERR=ABSOLUTE ERROR ARE
C  INPUT QUANTITIES.  SET THE FLAG, IFLAG, POSITIVE TO INITIALIZE
C  THE COMPUTATION.  AS B,C AND IFLAG ARE USED FOR BOTH INPUT AND
C  OUTPUT, THEY MUST BE VARIABLES IN THE CALLING PROGRAM.
C
C  IF 0 IS A POSSIBLE ROOT, ONE SHOULD NOT CHOOSE ABSERR=0.0.
C
```

```
C  THE OUTPUT VALUE OF B IS THE BETTER APPROXIMATION TO A ROOT
C  AS B AND C ARE ALWAYS REDEFINED SO THAT ABS(F(B)).LE.ABS(F(C)).
C
C  TO SOLVE THE EQUATION, ROOT MUST EVALUATE F(X) REPEATEDLY. THIS
C  IS DONE IN THE CALLING PROGRAM.  WHEN AN EVALUATION OF F IS
C  NEEDED AT T, ROOT RETURNS WITH IFLAG NEGATIVE.  EVALUATE FT=F(T)
C  AND CALL ROOT AGAIN.  DO NOT ALTER IFLAG.
C
C  WHEN THE COMPUTATION IS COMPLETE, ROOT RETURNS TO THE CALLING
C  PROGRAM WITH IFLAG POSITIVE.
C
C     IFLAG=1  IF F(B)*F(C).LT.0 AND THE STOPPING CRITERION IS MET.
C
C          =2  IF A VALUE B IS FOUND SUCH THAT THE COMPUTED VALUE
C              F(B) IS EXACTLY ZERO.  THE INTERVAL (B,C) MAY NOT
C              SATISFY THE STOPPING CRITERION.
C
C          =3  IF ABS(F(B)) EXCEEDS THE INPUT VALUES ABS(F(B)),
C              ABS(F(C)).   IN THIS CASE IT IS LIKELY THAT B IS CLOSE
C              TO A POLE OF F.
C
C          =4  IF NO ODD ORDER ROOT WAS FOUND IN THE INTERVAL.  A
C              LOCAL MINIMUM MAY HAVE BEEN OBTAINED.
C
C          =5  IF TOO MANY FUNCTION EVALUATIONS WERE MADE.
C              (AS PROGRAMMED, 500 ARE ALLOWED.)
C
C  THIS CODE IS A MODIFICATION OF THE CODE  ZEROIN  WHICH IS COMPLETELY
C  EXPLAINED AND DOCUMENTED IN THE TEXT, NUMERICAL COMPUTING, AN
C  INTRODUCTION  BY L. F. SHAMPINE AND R. C. ALLEN.
C
      SUBROUTINE ROOT(T,FT,B,C,RELERR,ABSERR,IFLAG)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      GENERIC
      COMMON/MLDRT/A,ACBS,AE,FA,FB,FC,FX,IC,KOUNT,RE
C*********************************************************************
C*  THE ONLY MACHINE DEPENDENT CONSTANT IS BASED ON THE MACHINE UNIT   *
C*  ROUNDOFF ERROR  U  WHICH IS THE SMALLEST POSITIVE NUMBER SUCH THAT *
C*  1.0+U .GT. 1.0 .  U  MUST BE CALCULATED AND INSERTED IN THE        *
C*  FOLLOWING DATA STATEMENT BEFORE USING  ROOT .  THE ROUTINE  MACHIN *
C*  CALCULATES  U .                                                    *
C*********************************************************************
      DATA U /2.2E-16/
C*********************************************************************
C
      IF(IFLAG.LT.0.0) GO TO 100
      RE=MAX(RELERR,U)
      AE=MAX(ABSERR,0.0D0)
      IC=0
      ACBS=ABS(B-C)
      A=C
      T=A
      IFLAG=-1
      RETURN
```

```
  100 IFLAG=IABS(IFLAG)
      GO TO (200,300,400),IFLAG
  200 FA=FT
      T=B
      IFLAG=-2
      RETURN
  300 FB=FT
      FC=FA
      KOUNT=2
      FX=MAX(ABS(FB),ABS(FC))
      GO TO 1
  400 FB=FT
      IF(FB.EQ.0.0) GO TO 9
      KOUNT=KOUNT+1
      IF(SIGN(1.0D0,FB).NE.SIGN(1.0D0,FC))GO TO 1
      C=A
      FC=FA
    1 IF(ABS(FC).GE.ABS(FB))GO TO 2
C
C  INTERCHANGE B AND C SO THAT ABS(F(B)).LE.ABS(F(C)).
C
      A=B
      FA=FB
      B=C
      FB=FC
      C=A
      FC=FA
    2 CMB=0.5*(C-B)
      ACMB=ABS(CMB)
      TOL=RE*ABS(B)+AE
C
C  TEST STOPPING CRITERION AND FUNCTION COUNT.
C
      IF(ACMB.LE.TOL)GO TO 8
      IF(KOUNT.GE.500)GO TO 12
C
C  CALCULATE NEW ITERATE IMPLICITLY AS B+P/Q
C  WHERE WE ARRANGE P.GE.0.  THE IMPLICIT
C  FORM IS USED TO PREVENT OVERFLOW.
C
      P=(B-A)*FB
      Q=FA-FB
      IF(P.GE.0.0)GO TO 3
      P=-P
      Q=-Q
C
C  UPDATE A, CHECK IF REDUCTION IN THE SIZE OF BRACKETING
C  INTERVAL IS SATISFACTORY.  IF NOT, BISECT UNTIL IT IS.
C
    3 A=B
      FA=FB
      IC=IC+1
      IF(IC.LT.4)GO TO 4
      IF(8.0*ACMB.GE.ACBS)GO TO 6
      IC=0
```

```
      ACBS=ACMB
C
C  TEST FOR TOO SMALL A CHANGE.
C
    4 IF(P.GT.ABS(Q)*TOL)GO TO 5
C
C  INCREMENT BY TOLERANCE.
C
      B=B+SIGN(TOL,CMB)
      GO TO 7
C
C  ROOT OUGHT TO BE BETWEEN B AND (C+B)/2.
C
    5 IF(P.GE.CMB*Q)GO TO 6
C
C  USE SECANT RULE.
C
      B=B+P/Q
      GO TO 7
C
C  USE BISECTION.
C
    6 B=0.5*(C+B)
C
C  HAVE COMPLETED COMPUTATION FOR NEW ITERATE B.
C
    7 T=B
      IFLAG=-3
      RETURN
C
C  FINISHED.  SET IFLAG.
C
    8 IF(SIGN(1.0D0,FB).EQ.SIGN(1.0D0,FC))GO TO 11
      IF(ABS(FB).GT.FX)GO TO 10
      IFLAG=1
      RETURN
    9 IFLAG=2
      RETURN
   10 IFLAG=3
      RETURN
   11 IFLAG=4
      RETURN
   12 IFLAG=5
      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SANDIA MATHEMATICAL PROGRAM LIBRARY
C     APPLIED MATHEMATICS DIVISION 2613
C     SANDIA LABORATORIES
C     ALBUQUERQUE, NEW MEXICO  87115
C     CONTROL DATA 6600/7600 VERSION 7.2  SEPTEMBER 1977
C
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                   *
```

```
C   *                 ISSUED BY SANDIA LABORATORIES,                  *
C   *                   A PRIME CONTRACTOR TO THE                     *
C   *   UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION  *
C   *                                                                 *
C   * * * * * * * * * * * * * *  NOTICE   * * * * * * * * * * * * * * *
C   *
C   * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE
C   * UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE
C   * UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,
C   * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,
C   * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS
C   * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY
C   * FOR THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,
C   * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS
C   * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.
C   *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *                                                                 *
C   * THE PRIMARY DOCUMENT FOR THE LIBRARY OF WHICH THIS ROUTINE IS   *
C   * A PART IS SAND75-0545.                                          *
C   *                                                                 *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   WRITTEN BY L. F. SHAMPINE AND M. K. GORDON
C
C   ABSTRACT
C
C   SUBROUTINE  STEP1  IS NORMALLY USED INDIRECTLY THROUGH SUBROUTINE
C   ODE .  BECAUSE  ODE  SUFFICES FOR MOST PROBLEMS AND IS MUCH EASIER
C   TO USE, USING IT SHOULD BE CONSIDERED BEFORE USING  STEP1  ALONE.
C
C   SUBROUTINE STEP1 INTEGRATES A SYSTEM OF  NEQN  FIRST ORDER ORDINARY
C   DIFFERENTIAL EQUATIONS ONE STEP, NORMALLY FROM X TO X+H, USING A
C   MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE FORMULAS.  LOCAL
C   EXTRAPOLATION IS USED TO IMPROVE ABSOLUTE STABILITY AND ACCURACY.
C   THE CODE ADJUSTS ITS ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR
C   PER UNIT STEP IN A GENERALIZED SENSE.  SPECIAL DEVICES ARE INCLUDED
C   TO CONTROL ROUNDOFF ERROR AND TO DETECT WHEN THE USER IS REQUESTING
C   TOO MUCH ACCURACY.
C
C   THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS, THE INITIAL
C   VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C   FURTHER DETAILS ON USE OF THIS CODE ARE AVAILABLE IN *SOLVING
C   ORDINARY DIFFERENTIAL EQUATIONS WITH ODE, STEP, AND INTRP*,
C   BY L. F. SHAMPINE AND M. K. GORDON, SLA-73-1060.
C
C
C   THE PARAMETERS REPRESENT --
C      F -- SUBROUTINE TO EVALUATE DERIVATIVES
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      Y(*) -- SOLUTION VECTOR AT X
C      X -- INDEPENDENT VARIABLE
C      H -- APPROPRIATE STEP SIZE FOR NEXT STEP.  NORMALLY DETERMINED BY
C           CODE
```

```
C      EPS -- LOCAL ERROR TOLERANCE
C      WT(*) -- VECTOR OF WEIGHTS FOR ERROR CRITERION
C      START -- LOGICAL VARIABLE SET .TRUE. FOR FIRST STEP,  .FALSE.
C           OTHERWISE
C      HOLD -- STEP SIZE USED FOR LAST SUCCESSFUL STEP
C      K -- APPROPRIATE ORDER FOR NEXT STEP (DETERMINED BY CODE)
C      KOLD -- ORDER USED FOR LAST SUCCESSFUL STEP
C      CRASH -- LOGICAL VARIABLE SET .TRUE. WHEN NO STEP CAN BE TAKEN,
C           .FALSE. OTHERWISE.
C      YP(*) -- DERIVATIVE OF SOLUTION VECTOR AT  X  AFTER SUCCESSFUL
C           STEP
C   THE ARRAYS  PHI, PSI  ARE REQUIRED FOR THE INTERPOLATION SUBROUTINE
C   INTRP .  THE ARRAY  P  IS INTERNAL TO THE CODE.  THE REMAINING NINE
C   VARIABLES AND ARRAYS ARE INCLUDED IN THE CALL LIST ONLY TO ELIMINATE
C   LOCAL RETENTION OF VARIABLES BETWEEN CALLS.
C
C   INPUT TO STEP1
C
C      FIRST CALL --
C
C   THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR ALL ARRAYS
C   IN THE CALL LIST, NAMELY
C
C     DIMENSION Y(NEQN),WT(NEQN),PHI(NEQN,16),P(NEQN),YP(NEQN),PSI(12),
C    1  ALPHA(12),BETA(12),SIG(13),V(12),W(12),G(13)
C                                --                --    **NOTE**
C
C   THE USER MUST ALSO DECLARE  START ,  CRASH ,  PHASE1  AND  NORND
C   LOGICAL VARIABLES AND  F  AN EXTERNAL SUBROUTINE, SUPPLY THE
C   SUBROUTINE  F(X,Y,YP)  TO EVALUATE
C      DY(I)/DX = YP(I) = F(X,Y(1),Y(2),...,Y(NEQN))
C   AND INITIALIZE ONLY THE FOLLOWING PARAMETERS.
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      Y(*) -- VECTOR OF INITIAL VALUES OF DEPENDENT VARIABLES
C      X -- INITIAL VALUE OF THE INDEPENDENT VARIABLE
C      H -- NOMINAL STEP SIZE INDICATING DIRECTION OF INTEGRATION
C           AND MAXIMUM SIZE OF STEP.  MUST BE VARIABLE
C      EPS -- LOCAL ERROR TOLERANCE PER STEP.  MUST BE VARIABLE
C      WT(*) -- VECTOR OF NON-ZERO WEIGHTS FOR ERROR CRITERION
C      START -- .TRUE.
C
C   STEP1  REQUIRES THAT THE L2 NORM OF THE VECTOR WITH COMPONENTS
C   LOCAL ERROR(L)/WT(L)  BE LESS THAN  EPS  FOR A SUCCESSFUL STEP.  THE
C   ARRAY  WT  ALLOWS THE USER TO SPECIFY AN ERROR TEST APPROPRIATE
C   FOR HIS PROBLEM.  FOR EXAMPLE,
C      WT(L) = 1.0  SPECIFIES ABSOLUTE ERROR,
C            = ABS(Y(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF THE
C                 L-TH COMPONENT OF THE SOLUTION,
C            = ABS(YP(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF
C                 THE L-TH COMPONENT OF THE DERIVATIVE,
C            = AMAX1(WT(L),ABS(Y(L)))  ERROR RELATIVE TO THE LARGEST
C                 MAGNITUDE OF L-TH COMPONENT OBTAINED SO FAR,
C            = ABS(Y(L))*RELERR/EPS + ABSERR/EPS  SPECIFIES A MIXED
C                 RELATIVE-ABSOLUTE TEST WHERE  RELERR  IS RELATIVE
C                 ERROR,  ABSERR  IS ABSOLUTE ERROR AND  EPS =
```

```
C                 AMAX1(RELERR,ABSERR) .
C
C      SUBSEQUENT CALLS --
C
C   SUBROUTINE  STEP1  IS DESIGNED SO THAT ALL INFORMATION NEEDED TO
C   CONTINUE THE INTEGRATION, INCLUDING THE STEP SIZE  H  AND THE ORDER
C   K , IS RETURNED WITH EACH STEP.  WITH THE EXCEPTION OF THE STEP
C   SIZE, THE ERROR TOLERANCE, AND THE WEIGHTS, NONE OF THE PARAMETERS
C   SHOULD BE ALTERED.  THE ARRAY  WT  MUST BE UPDATED AFTER EACH STEP
C   TO MAINTAIN RELATIVE ERROR TESTS LIKE THOSE ABOVE.  NORMALLY THE
C   INTEGRATION IS CONTINUED JUST BEYOND THE DESIRED ENDPOINT AND THE
C   SOLUTION INTERPOLATED THERE WITH SUBROUTINE  INTRP .  IF IT IS
C   IMPOSSIBLE TO INTEGRATE BEYOND THE ENDPOINT, THE STEP SIZE MAY BE
C   REDUCED TO HIT THE ENDPOINT SINCE THE CODE WILL NOT TAKE A STEP
C   LARGER THAN THE  H  INPUT.  CHANGING THE DIRECTION OF INTEGRATION,
C   I.E., THE SIGN OF  H , REQUIRES THE USER SET  START = .TRUE. BEFORE
C   CALLING  STEP1  AGAIN.  THIS IS THE ONLY SITUATION IN WHICH  START
C   SHOULD BE ALTERED.
C
C   OUTPUT FROM STEP1
C
C      SUCCESSFUL STEP --
C
C   THE SUBROUTINE RETURNS AFTER EACH SUCCESSFUL STEP WITH  START  AND
C   CRASH  SET .FALSE. .  X  REPRESENTS THE INDEPENDENT VARIABLE
C   ADVANCED ONE STEP OF LENGTH  HOLD  FROM ITS VALUE ON INPUT AND  Y
C   THE SOLUTION VECTOR AT THE NEW VALUE OF  X .  ALL OTHER PARAMETERS
C   REPRESENT INFORMATION CORRESPONDING TO THE NEW  X  NEEDED TO
C   CONTINUE THE INTEGRATION.
C
C      UNSUCCESSFUL STEP --
C
C   WHEN THE ERROR TOLERANCE IS TOO SMALL FOR THE MACHINE PRECISION,
C   THE SUBROUTINE RETURNS WITHOUT TAKING A STEP AND  CRASH = .TRUE. .
C   AN APPROPRIATE STEP SIZE AND ERROR TOLERANCE FOR CONTINUING ARE
C   ESTIMATED AND ALL OTHER INFORMATION IS RESTORED AS UPON INPUT
C   BEFORE RETURNING.  TO CONTINUE WITH THE LARGER TOLERANCE, THE USER
C   JUST CALLS THE CODE AGAIN.  A RESTART IS NEITHER REQUIRED NOR
C   DESIRABLE.
C
      SUBROUTINE STEP1(F,NEQN,Y,X,H,EPS,WT,START,
     1 HOLD,K,KOLD,CRASH,PHI,P,YP,PSI,
     2 ALPHA,BETA,SIG,V,W,G,PHASE1,NS,NORND)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      GENERIC
      LOGICAL START,CRASH,PHASE1,NORND
      DIMENSION Y(25),WT(25),PHI(25,16),P(25),YP(25),PSI(12),
     1  ALPHA(12),BETA(12),SIG(13),V(12),W(12),G(13)
      DIMENSION TWO(13),GSTR(13)
      EXTERNAL F
C*********************************************************************
C*  THE ONLY MACHINE DEPENDENT CONSTANTS ARE BASED ON THE MACHINE UNIT *
C*  ROUNDOFF ERROR  U  WHICH IS THE SMALLEST POSITIVE NUMBER SUCH THAT *
C*  1.0+U .GT. 1.0  .  THE USER MUST CALCULATE  U  AND INSERT          *
```

```
C*  TWOU=2.0*U  AND  FOURU=4.0*U  IN THE DATA STATEMENT BEFORE CALLING *
C*  THE CODE.  THE ROUTINE  MACHIN  CALCULATES  U .                     *
      DATA TWOU,FOURU/4.4E-16,8.8E-16/
C*********************************************************************
C
      DATA TWO/2.0,4.0,8.0,16.0,32.0,64.0,128.0,256.0,512.0,1024.0,
     1  2048.0,4096.0,8192.0/
      DATA GSTR/0.500,0.0833,0.0417,0.0264,0.0188,0.0143,0.0114,0.00936,
     1  0.00789,0.00679,0.00592,0.00524,0.00468/
C
C
C       ***     BEGIN BLOCK 0     ***
C   CHECK IF STEP SIZE OR ERROR TOLERANCE IS TOO SMALL FOR MACHINE
C   PRECISION.  IF FIRST STEP, INITIALIZE PHI ARRAY AND ESTIMATE A
C   STARTING STEP SIZE.
C                   ***
C
C   IF STEP SIZE IS TOO SMALL, DETERMINE AN ACCEPTABLE ONE
C
      CRASH = .TRUE.
      IF(ABS(H) .GE. FOURU*ABS(X)) GO TO 5
      H = SIGN(FOURU*ABS(X),H)
      RETURN
 5    P5EPS = 0.5*EPS
C
C   IF ERROR TOLERANCE IS TOO SMALL, INCREASE IT TO AN ACCEPTABLE VALUE
C
      ROUND = 0.0
      DO 10 L = 1,NEQN
 10     ROUND = ROUND + (Y(L)/WT(L))**2
      ROUND = TWOU*SQRT(ROUND)
      IF(P5EPS .GE. ROUND) GO TO 15
      EPS = 2.0*ROUND*(1.0 + FOURU)
      RETURN
 15   CRASH = .FALSE.
      G(1) = 1.0
      G(2) = 0.5
      SIG(1) = 1.0
      IF(.NOT.START) GO TO 99
C
C   INITIALIZE.  COMPUTE APPROPRIATE STEP SIZE FOR FIRST STEP
C
      CALL F(X,Y,YP)
      SUM = 0.0
      DO 20 L = 1,NEQN
        PHI(L,1) = YP(L)
        PHI(L,2) = 0.0
 20     SUM = SUM + (YP(L)/WT(L))**2
      SUM = SQRT(SUM)
      ABSH = ABS(H)
      IF(EPS .LT. 16.0*SUM*H*H) ABSH = 0.25*SQRT(EPS/SUM)
      H = SIGN(MAX(ABSH,FOURU*ABS(X)),H)
      HOLD = 0.0
      K = 1
      KOLD = 0
```

```
      START = .FALSE.
      PHASE1 = .TRUE.
      NORND = .TRUE.
      IF(P5EPS .GT. 100.0*ROUND) GO TO 99
      NORND = .FALSE.
      DO 25 L = 1,NEQN
 25     PHI(L,15) = 0.0
 99   IFAIL = 0
C       ***     END BLOCK 0     ***
C
C       ***     BEGIN BLOCK 1     ***
C   COMPUTE COEFFICIENTS OF FORMULAS FOR THIS STEP.  AVOID COMPUTING
C   THOSE QUANTITIES NOT CHANGED WHEN STEP SIZE IS NOT CHANGED.
C                   ***
C
 100  KP1 = K+1
      KP2 = K+2
      KM1 = K-1
      KM2 = K-2
C
C   NS IS THE NUMBER OF STEPS TAKEN WITH SIZE H, INCLUDING THE CURRENT
C   ONE.  WHEN K.LT.NS, NO COEFFICIENTS CHANGE
C
      IF(H .NE. HOLD) NS = 0
      IF (NS.LE.KOLD) NS = NS+1
      NSP1 = NS+1
      IF (K .LT. NS) GO TO 199
C
C   COMPUTE THOSE COMPONENTS OF ALPHA(*),BETA(*),PSI(*),SIG(*) WHICH
C   ARE CHANGED
C
      BETA(NS) = 1.0
      REALNS = NS
      ALPHA(NS) = 1.0/REALNS
      TEMP1 = H*REALNS
      SIG(NSP1) = 1.0
      IF(K .LT. NSP1) GO TO 110
      DO 105 I = NSP1,K
        IM1 = I-1
        TEMP2 = PSI(IM1)
        PSI(IM1) = TEMP1
        BETA(I) = BETA(IM1)*PSI(IM1)/TEMP2
        TEMP1 = TEMP2 + H
        ALPHA(I) = H/TEMP1
        REALI = I
 105    SIG(I+1) = REALI*ALPHA(I)*SIG(I)
 110  PSI(K) = TEMP1
C
C   COMPUTE COEFFICIENTS G(*)
C
C   INITIALIZE V(*) AND SET W(*).
C
      IF(NS .GT. 1) GO TO 120
      DO 115 IQ = 1,K
        TEMP3 = IQ*(IQ+1)
```

```
        V(IQ) = 1.0/TEMP3
 115    W(IQ) = V(IQ)
      GO TO 140
C
C   IF ORDER WAS RAISED, UPDATE DIAGONAL PART OF V(*)
C
 120  IF(K .LE. KOLD) GO TO 130
      TEMP4 = K*KP1
      V(K) = 1.0/TEMP4
      NSM2 = NS-2
      IF(NSM2 .LT. 1) GO TO 130
      DO 125 J = 1,NSM2
        I = K-J
 125    V(I) = V(I) - ALPHA(J+1)*V(I+1)
C
C   UPDATE V(*) AND SET W(*)
C
 130  LIMIT1 = KP1 - NS
      TEMP5 = ALPHA(NS)
      DO 135 IQ = 1,LIMIT1
        V(IQ) = V(IQ) - TEMP5*V(IQ+1)
 135    W(IQ) = V(IQ)
      G(NSP1) = W(1)
C
C   COMPUTE THE G(*) IN THE WORK VECTOR W(*)
C
 140  NSP2 = NS + 2
      IF(KP1 .LT. NSP2) GO TO 199
      DO 150 I = NSP2,KP1
        LIMIT2 = KP2 - I
        TEMP6 = ALPHA(I-1)
        DO 145 IQ = 1,LIMIT2
 145      W(IQ) = W(IQ) - TEMP6*W(IQ+1)
 150    G(I) = W(1)
 199    CONTINUE
C       ***     END BLOCK 1     ***
C
C       ***     BEGIN BLOCK 2     ***
C   PREDICT A SOLUTION P(*), EVALUATE DERIVATIVES USING PREDICTED
C   SOLUTION, ESTIMATE LOCAL ERROR AT ORDER K AND ERRORS AT ORDERS K,
C   K-1, K-2 AS IF CONSTANT STEP SIZE WERE USED.
C                   ***
C
C   CHANGE PHI TO PHI STAR
C
      IF(K .LT. NSP1) GO TO 215
      DO 210 I = NSP1,K
        TEMP1 = BETA(I)
        DO 205 L = 1,NEQN
 205      PHI(L,I) = TEMP1*PHI(L,I)
 210    CONTINUE
C
C   PREDICT SOLUTION AND DIFFERENCES
C
 215  DO 220 L = 1,NEQN
```

```
        PHI(L,KP2) = PHI(L,KP1)
        PHI(L,KP1) = 0.0
 220    P(L) = 0.0
      DO 230 J = 1,K
        I = KP1 - J
        IP1 = I+1
        TEMP2 = G(I)
        DO 225 L = 1,NEQN
          P(L) = P(L) + TEMP2*PHI(L,I)
 225      PHI(L,I) = PHI(L,I) + PHI(L,IP1)
 230    CONTINUE
      IF(NORND) GO TO 240
      DO 235 L = 1,NEQN
        TAU = H*P(L) - PHI(L,15)
        P(L) = Y(L) + TAU
 235    PHI(L,16) = (P(L) - Y(L)) - TAU
      GO TO 250
 240  DO 245 L = 1,NEQN
 245    P(L) = Y(L) + H*P(L)
 250  XOLD = X
      X = X + H
      ABSH = ABS(H)
      CALL F(X,P,YP)
C
C   ESTIMATE ERRORS AT ORDERS K,K-1,K-2
C
      ERKM2 = 0.0
      ERKM1 = 0.0
      ERK = 0.0
      DO 265 L = 1,NEQN
        TEMP3 = 1.0/WT(L)
        TEMP4 = YP(L) - PHI(L,1)
        IF(KM2)265,260,255
 255    ERKM2 = ERKM2 + ((PHI(L,KM1)+TEMP4)*TEMP3)**2
 260    ERKM1 = ERKM1 + ((PHI(L,K)+TEMP4)*TEMP3)**2
 265    ERK = ERK + (TEMP4*TEMP3)**2
      IF(KM2)280,275,270
 270  ERKM2 = ABSH*SIG(KM1)*GSTR(KM2)*SQRT(ERKM2)
 275  ERKM1 = ABSH*SIG(K)*GSTR(KM1)*SQRT(ERKM1)
 280  TEMP5 = ABSH*SQRT(ERK)
      ERR = TEMP5*(G(K)-G(KP1))
      ERK = TEMP5*SIG(KP1)*GSTR(K)
      KNEW = K
C
C   TEST IF ORDER SHOULD BE LOWERED
C
      IF(KM2)299,290,285
 285  IF(MAX(ERKM1,ERKM2) .LE. ERK) KNEW = KM1
      GO TO 299
 290  IF(ERKM1 .LE. 0.5*ERK) KNEW = KM1
C
C   TEST IF STEP SUCCESSFUL
C
 299  IF(ERR .LE. EPS) GO TO 400
C       ***     END BLOCK 2     ***
```

```
C
C       ***     BEGIN BLOCK 3     ***
C   THE STEP IS UNSUCCESSFUL.  RESTORE  X, PHI(*,*), PSI(*) .
C   IF THIRD CONSECUTIVE FAILURE, SET ORDER TO ONE.  IF STEP FAILS MORE
C   THAN THREE TIMES, CONSIDER AN OPTIMAL STEP SIZE.  DOUBLE ERROR
C   TOLERANCE AND RETURN IF ESTIMATED STEP SIZE IS TOO SMALL FOR MACHINE
C   PRECISION.
C                   ***
C
C   RESTORE X, PHI(*,*) AND PSI(*)
C
      PHASE1 = .FALSE.
      X = XOLD
      DO 310 I = 1,K
        TEMP1 = 1.0/BETA(I)
        IP1 = I+1
        DO 305 L = 1,NEQN
 305      PHI(L,I) = TEMP1*(PHI(L,I) - PHI(L,IP1))
 310    CONTINUE
      IF(K .LT. 2) GO TO 320
      DO 315 I = 2,K
 315    PSI(I-1) = PSI(I) - H
C
C   ON THIRD FAILURE, SET ORDER TO ONE.  THEREAFTER, USE OPTIMAL STEP
C   SIZE
C
 320  IFAIL = IFAIL + 1
      TEMP2 = 0.5
      IF(IFAIL - 3) 335,330,325
 325  IF(P5EPS .LT. 0.25*ERK) TEMP2 = SQRT(P5EPS/ERK)
 330  KNEW = 1
 335  H = TEMP2*H
      K = KNEW
      IF(ABS(H) .GE. FOURU*ABS(X)) GO TO 340
      CRASH = .TRUE.
      H = SIGN(FOURU*ABS(X),H)
      EPS = EPS + EPS
      RETURN
 340  GO TO 100
C       ***     END BLOCK 3     ***
C
C       ***     BEGIN BLOCK 4     ***
C   THE STEP IS SUCCESSFUL.  CORRECT THE PREDICTED SOLUTION, EVALUATE
C   THE DERIVATIVES USING THE CORRECTED SOLUTION AND UPDATE THE
C   DIFFERENCES.  DETERMINE BEST ORDER AND STEP SIZE FOR NEXT STEP.
C                   ***
 400  KOLD = K
      HOLD = H
C
C   CORRECT AND EVALUATE
C
      TEMP1 = H*G(KP1)
      IF(NORND) GO TO 410
      DO 405 L = 1,NEQN
        RHO = TEMP1*(YP(L) - PHI(L,1)) - PHI(L,16)
```

```
        Y(L) = P(L) + RHO
 405    PHI(L,15) = (Y(L) - P(L)) - RHO
      GO TO 420
 410  DO 415 L = 1,NEQN
 415    Y(L) = P(L) + TEMP1*(YP(L) - PHI(L,1))
 420  CALL F(X,Y,YP)
C
C   UPDATE DIFFERENCES FOR NEXT STEP
C
      DO 425 L = 1,NEQN
        PHI(L,KP1) = YP(L) - PHI(L,1)
 425    PHI(L,KP2) = PHI(L,KP1) - PHI(L,KP2)
      DO 435 I = 1,K
        DO 430 L = 1,NEQN
 430      PHI(L,I) = PHI(L,I) + PHI(L,KP1)
 435    CONTINUE
C
C   ESTIMATE ERROR AT ORDER K+1 UNLESS:
C     IN FIRST PHASE WHEN ALWAYS RAISE ORDER,
C     ALREADY DECIDED TO LOWER ORDER,
C     STEP SIZE NOT CONSTANT SO ESTIMATE UNRELIABLE
C
      ERKP1 = 0.0
      IF(KNEW .EQ. KM1  .OR.  K .EQ. 12) PHASE1 = .FALSE.
      IF(PHASE1) GO TO 450
      IF(KNEW .EQ. KM1) GO TO 455
      IF(KP1 .GT. NS) GO TO 460
      DO 440 L = 1,NEQN
 440    ERKP1 = ERKP1 + (PHI(L,KP2)/WT(L))**2
      ERKP1 = ABSH*GSTR(KP1)*SQRT(ERKP1)
C
C   USING ESTIMATED ERROR AT ORDER K+1, DETERMINE APPROPRIATE ORDER
C   FOR NEXT STEP
C
      IF(K .GT. 1) GO TO 445
      IF(ERKP1 .GE. 0.5*ERK) GO TO 460
      GO TO 450
 445  IF(ERKM1 .LE. MIN(ERK,ERKP1)) GO TO 455
      IF(ERKP1 .GE. ERK  .OR.  K .EQ. 12) GO TO 460
C
C   HERE ERKP1 .LT. ERK .LT. AMAX1(ERKM1,ERKM2) ELSE ORDER WOULD HAVE
C   BEEN LOWERED IN BLOCK 2.  THUS ORDER IS TO BE RAISED
C
C   RAISE ORDER
C
 450  K = KP1
      ERK = ERKP1
      GO TO 460
C
C   LOWER ORDER
C
 455  K = KM1
      ERK = ERKM1
C
C   WITH NEW ORDER DETERMINE APPROPRIATE STEP SIZE FOR NEXT STEP
```

```
C
 460  HNEW = H + H
      IF(PHASE1) GO TO 465
      IF(P5EPS .GE. ERK*TWO(K+1)) GO TO 465
      HNEW = H
      IF(P5EPS .GE. ERK) GO TO 465
      TEMP2 = K+1
      R = (P5EPS/ERK)**(1.0/TEMP2)
      HNEW = ABSH*MAX(0.5D0,MIN(0.9D0,R))
      HNEW = SIGN(MAX(HNEW,FOURU*ABS(X)),H)
 465  H = HNEW
      RETURN
C       ***     END BLOCK 4     ***
      END
```

SAMPLE INPUT FOR CYCLE SIMULATION

```
$INPUT
FIRE   = .TRUE.
SPBURN = .TRUE.
FUELTP = 1
PHI    = 1.00
ECCEN  = 1.50
ROTRAD= 10.5
DEPTH  = 7.00
VFLANK = 35.00
RPM    = 3000.
TIPO   = -530.0
TIPC   = -180.0
TEPO   = 199.0
TEPC   = 588.5
TSPARK = -30.0
THIPO  = 120.0
THEPO  = 40.0
IPA    = 13.8
EPA    = 6.5
XBZERO = 0.0003
XBSTOP = 0.995
DTBRN  = 90.0
CONSPB =  5.0
EXSPB  = 1.50
PATM   = 1.000
TATM   = 300.0
PIM    = 0.980
TFRESH = 300.0
TEGR   = 300.0
EGR    = 0.0
PEM    = 1.02
TROTOR = 370.
TSIDE  = 370.
THOUS  = 370.
CONHT  = 0.0350
EXPHT  = .8
TPRINT = 10.00
TPRINX = 1.0
AREROT = .0002
CIINTG = 0.0001
CCINTG = 0.0001
CBINTG = 0.00005
CEINTG = 0.0001
MXTRY  = 1
REL    = .0002
MAXITS = 3
MAXERR = 0.03
MAXTRY = 2
AREALK = 0.0100
CREVOL = 0.875
TCREV  = 370
CON1   = 0.75
CON2   = 0.324
$END
```

SAMPLE MOTORING OUTPUT

M.I.T. ZERO-DIMENSIONAL WANKEL ENGINE CYCLE SIMULATION

>>>>> INPUT DATA <<<<<

>>>>> OPERATING MODE

MOTORED CYCLE

>>>>> OPERATING CONDITIONS

ENGINE SPEED = 2052.0 RPM

>>>>> MANIFOLD CONDITIONS

INTAKE MANIFOLD PRESSURE = 0.9300 ATM

EXHAUST MANIFOLD PRESSURE = 1.0200 ATM

FRESH CHARGE TEMPERATURE = 312.70 K

EXHAUST GAS RECIRCULATION = 0.00 %

EGR TEMPERATURE	=	300.00	K
INTAKE CHARGE TEMPERATURE	=	312.70	K
ATMOSPHERIC PRESSURE	=	0.9810	ATM
ATMOSPHERIC TEMPERATURE	=	300.60	K

>>>>> HEAT TRANSFER AND TURBULENCE PARAMETERS

HEAT TRANSFER CONSTANT	=	0.3770	
HEAT TRANSFER EXPONENT	=	0.8000	
ROTOR TEMPERATURE	=	350.00	K
SIDE WALL TEMPERATURE	=	350.00	K
HOUSING WALL TEMPERATURE	=	350.00	K

>>>>> ENGINE DESIGN PARAMETERS

ECCENTRICITY OF ROTOR	=	1.500	CM
RADIUS OF ROTOR	=	10.500	CM
DEPTH OF CHAMBER	=	7.000	CM
COMPRESSION RATIO	=	9.407	
DISPLACED VOLUME	=	572.875	CC
VOLUME OF ROTOR POCKET	=	35.000	CC
INTAKE PORT OPENS	=	-530.0	DEG CA
INTAKE PORT CLOSES	=	-180.0	DEG CA
EXHAUST PORT OPENS	=	199.0	DEG CA
EXHAUST PORT CLOSES	=	588.5	DEG CA

```
>>>>> LEAKAGE AND CREVICE VOLUME PARAMETERS

          LEAK AREA PER APEX      =     0.010000  CM*CM

          CREVICE VOLUME PER APEX=      0.875000  CC

          CREVICE GAS TEMPERATURE=    350.000000  K
```

```
>>>>> COMPUTATIONAL PARAMETERS

          MAXIMUM # OF ITERATIONS =      2

          OUTPUT AT ITERATION #   =      1

          TPRINT                  =     10.00
          TPRINX                  =      1.00
          XBZERO                  =      0.00050
          XESTOP                  =      0.00000
          XBSTOP                  =      0.99500
          CIINTG                  =      0.000100
          CCINTG                  =      0.000100
          CBINTG                  =      0.000010
          CEINTG                  =      0.000100
          AREROT                  =      0.000200
          REL                     =      0.000200
          ERMAX                   =      0.000000
          MAXERR                  =      0.030000
          MAXTRY                  =      2
```

>>>>> START OF INTAKE PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MIN (G)	MEX (G)	VIV (CM/SEC)	VEV (CM/SEC)	X1 (-)	Q DOT (KJ/DEG)	WORK (KJ)	IMF	IFG
-530.0	1.1000	330.00	0.00000	0.00000	0.0		1.00000		0.000000		
-520.0	1.0045	325.32	-0.00452	-0.00474	-8484.6	-3499.6	1.00000	-0.000030	0.000596	0	2
-510.0	0.9588	324.43	-0.01515	-0.02177	-5528.6	-7087.2	1.00000	-0.000030	0.001547	0	2
-501.1	0.9294	324.43	-0.01989	-0.03538	812.9	-8745.1	1.00000	-0.000030	0.002641	0	8
-500.0	0.9280	324.63	-0.01962	-0.03659	1509.3	-8822.5	1.00000	-0.000030	0.002800	0	2
-490.0	0.9127	325.79	-0.00948	-0.04088	4473.9	0.0	1.00000	-0.000029	0.004354	0	2
-484.7	0.9120	326.97	-0.00000	-0.04088	4568.4	0.0	1.00000	-0.000028	0.005293	0	8
-480.0	0.9124	327.40	0.00941	-0.04088	4332.2	0.0	1.00000	-0.000027	0.006192	1	2
-470.0	0.9140	327.19	0.03217	-0.04088	4137.6	0.0	1.00000	-0.000029	0.008307	1	2
-460.0	0.9161	326.94	0.05777	-0.04088	3864.7	0.0	1.00000	-0.000030	0.010672	1	2
-450.0	0.9169	326.59	0.08560	-0.04088	3744.6	0.0	1.00000	-0.000032	0.013257	1	2
-440.0	0.9186	326.34	0.11558	-0.04088	3503.9	0.0	1.00000	-0.000033	0.016025	1	2
-430.0	0.9197	326.11	0.14706	-0.04088	3334.6	0.0	1.00000	-0.000035	0.018941	1	2
-420.0	0.9212	325.97	0.17981	-0.04088	3086.1	0.0	1.00000	-0.000037	0.021964	1	2
-410.0	0.9222	325.86	0.21316	-0.04088	2903.4	0.0	1.00000	-0.000039	0.025054	1	2
-400.0	0.9225	325.74	0.24654	-0.04088	2848.1	0.0	1.00000	-0.000040	0.028168	1	2
-390.0	0.9228	325.70	0.27966	-0.04088	2790.0	0.0	1.00000	-0.000042	0.031260	1	2
-380.0	0.9234	325.76	0.31215	-0.04088	2670.9	0.0	1.00000	-0.000043	0.034291	1	2
-370.0	0.9236	325.83	0.34334	-0.04088	2629.9	0.0	1.00000	-0.000044	0.037220	1	2
-360.0	0.9243	326.00	0.37311	-0.04088	2485.8	0.0	1.00000	-0.000045	0.040007	1	2
-350.0	0.9242	326.15	0.40054	-0.04088	2503.0	0.0	1.00000	-0.000046	0.042614	1	2
-340.0	0.9247	326.42	0.42582	-0.04088	2389.5	0.0	1.00000	-0.000047	0.045005	1	2
-330.0	0.9252	326.73	0.44830	-0.04088	2287.9	0.0	1.00000	-0.000047	0.047147	1	2
-320.0	0.9261	327.15	0.46795	-0.04088	2051.7	0.0	1.00000	-0.000047	0.049012	1	2
-310.0	0.9268	327.59	0.48411	-0.04088	1877.2	0.0	1.00000	-0.000047	0.050575	1	2
-300.0	0.9268	328.04	0.49639	-0.04088	1874.6	0.0	1.00000	-0.000046	0.051814	1	2
-290.0	0.9286	328.71	0.50584	-0.04088	1230.7	0.0	1.00000	-0.000045	0.052712	1	2
-280.3	0.9304	329.43	0.51130	-0.04088	-673.5	0.0	1.00000	-0.000044	0.053245	1	8
-278.7	0.9301	329.51	0.51151	-0.04088	-276.6	0.0	1.00000	-0.000044	0.053300	0	8
-270.0	0.9301	330.10	0.51185	-0.04088	-305.4	0.0	1.00000	-0.000043	0.053439	0	2
-260.0	0.9315	330.89	0.50936	-0.04088	-1304.3	0.0	1.00000	-0.000041	0.053256	0	2
-250.0	0.9360	331.96	0.50462	-0.04088	-2612.5	0.0	1.00000	-0.000039	0.052709	0	2
-240.0	0.9464	333.59	0.49899	-0.04088	-4288.5	0.0	1.00000	-0.000035	0.051798	0	2
-230.0	0.9700	336.45	0.49571	-0.04088	-6556.6	0.0	1.00000	-0.000029	0.050519	0	2
-220.0	1.0078	340.51	0.49481	-0.04088	-8839.9	0.0	1.00000	-0.000021	0.048850	0	2
-210.0	1.0573	345.43	0.49457	-0.04088	-10838.0	0.0	1.00000	-0.000010	0.046770	0	2
-200.0	1.1192	351.10	0.49452	-0.04088	-12550.7	0.0	1.00000	0.000002	0.044251	0	2
-190.0	1.1949	357.52	0.49450	-0.04088	-13995.2	0.0	1.00000	0.000017	0.041261	0	2
-181.7	1.2703	363.44	0.49450	-0.04088	-14997.9	0.0	1.00000	0.000031	0.038392	0	8
-180.0	1.2871	364.71	0.49450	-0.04088	0.0	0.0	1.00000	0.000034	0.037764	0	2
-180.0	1.2871	364.71	0.49450	-0.04088	0.0	0.0	1.00000	0.000034	0.037764	0	2

>>>>> START OF COMPRESSION PROCESS

CA (DEG)	P (ATM)	TEMP (K)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
-180.0	1.2871	364.71	0.000034	0.037764	2
-170.0	1.3989	372.70	0.000054	0.033717	2
-160.0	1.5347	381.52	0.000077	0.029071	2
-150.0	1.6999	391.23	0.000105	0.023768	2
-140.0	1.9016	401.84	0.000137	0.017742	2
-130.0	2.1492	413.41	0.000175	0.010921	2
-120.0	2.4547	425.96	0.000221	0.003224	2
-110.0	2.8339	439.47	0.000276	-0.005435	2
-100.0	3.3068	453.89	0.000343	-0.015140	2
-90.0	3.8990	469.10	0.000424	-0.025964	2
-80.0	4.6410	484.85	0.000522	-0.037953	2
-70.0	5.5669	500.73	0.000640	-0.051086	2
-60.0	6.7101	516.23	0.000779	-0.065237	2
-50.0	8.0750	529.81	0.000935	-0.080064	2
-40.0	9.6198	539.95	0.001096	-0.094935	2
-30.0	11.2025	544.64	0.001236	-0.108815	2
-20.0	12.5466	541.76	0.001317	-0.120296	2
-10.0	13.2818	529.76	0.001296	-0.127879	2
-6.7	13.3335	523.78	0.001262	-0.129308	8
0.0	13.1140	508.64	0.001154	-0.130488	2
10.0	12.0541	480.84	0.000920	-0.128040	2
20.0	10.4106	449.55	0.000652	-0.121492	2
30.0	8.5997	418.11	0.000404	-0.112351	2
40.0	6.9279	388.85	0.000206	-0.102050	2
50.0	5.5318	362.93	0.000061	-0.091619	2
60.0	4.4279	340.64	-0.000039	-0.081658	2
70.0	3.5779	321.81	-0.000106	-0.072450	2
80.0	2.9295	306.04	-0.000151	-0.064090	2
90.0	2.4347	292.91	-0.000179	-0.056566	2
100.0	2.0549	282.04	-0.000198	-0.049826	2
110.0	1.7611	273.08	-0.000209	-0.043798	2
120.0	1.5316	265.76	-0.000216	-0.038408	2
130.0	1.3508	259.83	-0.000220	-0.033589	2
140.0	1.2073	255.11	-0.000222	-0.029281	2
150.0	1.0926	251.44	-0.000222	-0.025433	2
160.0	1.0005	248.71	-0.000221	-0.022001	2
170.0	0.9264	246.80	-0.000220	-0.018949	2
180.0	0.8669	245.64	-0.000217	-0.016247	2
190.0	0.8194	245.15	-0.000215	-0.013871	2

>>>>> START OF EXHAUST PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MEX (G)	VEV (M/SEC)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
199.0	0.7852	245.25	-0.04088	0.0	-0.000212	-0.011995	2
200.0	0.7821	245.31	-0.04097	-13339.1	-0.000211	-0.011802	2
210.0	0.7735	247.87	-0.05292	-13717.9	-0.000207	-0.010008	2
220.0	0.8028	253.72	-0.08392	-12805.9	-0.000203	-0.008426	2
230.0	0.8620	261.86	-0.12981	-10742.2	-0.000196	-0.007026	2
240.0	0.9364	270.75	-0.18160	-7649.8	-0.000187	-0.005827	2
250.0	0.9944	277.77	-0.22022	-4171.1	-0.000178	-0.004892	2
260.0	1.0199	281.93	-0.23596	-285.7	-0.000171	-0.004301	2
270.0	1.0208	284.05	-0.23308	747.0	-0.000165	-0.004101	2
280.0	1.0227	286.19	-0.22608	1355.4	-0.000159	-0.004301	2
290.0	1.0256	288.36	-0.21510	1960.6	-0.000153	-0.004901	2
300.0	1.0296	290.55	-0.20023	2552.8	-0.000146	-0.005896	2
310.0	1.0344	292.78	-0.18156	3127.3	-0.000139	-0.007276	2
320.0	1.0400	295.02	-0.15925	3680.4	-0.000131	-0.009026	2
330.0	1.0463	297.27	-0.13346	4205.4	-0.000124	-0.011129	2
340.0	1.0529	299.51	-0.10438	4696.2	-0.000116	-0.013560	2
350.0	1.0598	301.74	-0.07226	5147.1	-0.000109	-0.016291	2
360.0	1.0666	303.94	-0.03738	5551.7	-0.000101	-0.019290	2
370.0	1.0730	306.07	-0.00007	5904.0	-0.000094	-0.022516	2
380.0	1.0786	308.13	0.03929	6196.0	-0.000087	-0.025929	2
390.0	1.0830	310.08	0.08026	6420.1	-0.000080	-0.029478	2
400.0	1.0860	311.91	0.12233	6568.4	-0.000074	-0.033114	2
410.0	1.0870	313.61	0.16492	6632.3	-0.000068	-0.036782	2
420.0	1.0860	315.16	0.20741	6603.3	-0.000062	-0.040425	2
430.0	1.0827	316.57	0.24912	6473.8	-0.000057	-0.043987	2
440.0	1.0775	317.89	0.28935	6239.3	-0.000052	-0.047413	2
450.0	1.0706	319.15	0.32741	5904.3	-0.000048	-0.050653	2
460.0	1.0628	320.44	0.36267	5482.7	-0.000044	-0.053661	2
470.0	1.0551	321.88	0.39456	5017.9	-0.000040	-0.056397	2
480.0	1.0466	323.39	0.42286	4416.1	-0.000036	-0.058830	2
490.0	1.0382	325.04	0.44722	3686.8	-0.000033	-0.060932	2
500.0	1.0313	326.98	0.46730	2933.2	-0.000029	-0.062679	2
510.0	1.0270	329.26	0.48296	2336.3	-0.000026	-0.064055	2
520.0	1.0231	331.58	0.49434	1561.2	-0.000022	-0.065048	2
530.0	1.0212	333.98	0.50130	979.4	-0.000019	-0.065647	2
540.0	1.0199	336.21	0.50396	-314.3	-0.000016	-0.065847	2
550.0	1.0199	338.24	0.50234	-276.0	-0.000014	-0.065647	2

```
>--------------+--------------<
>                             <
>     CALCULATION RESULTS     <
>                             <
>--------------+--------------<
```

--> VOLUMETRIC EFFICIENCY; (%) BASED ON: INTAKE / ATM	---->	82.6	75.3
--> PUMPING MEAN EFFECTIVE PRESSURE; (KPA) : PEMP	---->	-14.	
--> GROSS INDICATED MEAN EFFECTIVE PRESSURE; (KPA) : IMEP	---->	-100.	
--> GROSS INDICATED SPECIFIC FUEL CONSUMPTION; (G/IKW-HR) : ISFC	---->	0.	
--> GROSS INDICATED THERMAL EFFICIENCY; (%)	---->	0.0	
--> NET INDICATED THERMAL EFFICIENCY; (%)	---->	0.0	
--> (HEAT TRANSFER PER CYCLE)/ (MASS OF FUEL TIMES LHV); (%)	---->	0.0	
--> IGNITION DELAY (0 - 10%) (CRANK ANGLE) / (MS)	---->	0.00	0.00
--> BURN DURATION (10 - 90%) (CRANK ANGLE) / (MS)	---->	0.00	0.00
--> MEAN EXHAUST TEMPERATURE; (K)	---->	328.0	

MASS IN CYLINDER AT TIVO = 0.08210 G

MASS IN CYLINDER AT TIVC = 0.61748 G

MASS OF FUEL INDUCTED = 0.00000 G

RESIDUAL FRACTION = 0.00000

HEATI = -0.010014 KJ (TIPO - -270)
WORKI = 0.053439 KJ

HEATCE = 0.088180 KJ (TIPC - +270
WORKCE = -0.057539 KJ

HEATE = -0.021527 KJ (+270 - TIPO)
WORKE = -0.061546 KJ

TOTAL ENTHALPY IN / CYCLE = 0.15581 KJ

TOTAL ENTHALPY OUT / CYCLE = 0.16607 KJ

TOTAL HEAT LOSS / CYCLE = 0.05664 KJ

TOTAL WORK OUTPUT / CYCLE = -0.06565 KJ

HEAT LOSS TO CREVICE/CYCLE = 0.00000 KJ

"LOST" FUEL ENERGY = 0.00000 KJ

NET ENERGY GAIN / CYCLE = 0.00125 KJ

(ENERGY GAIN)/(ENTHALPY IN) = 0.80377 %

(ENERGY GAIN)/(MFUEL*LHV) = 0.00000 %

M.I.T. ZERO-DIMENSIONAL WANKEL ENGINE CYCLE SIMULATION

>>>>> INPUT DATA <<<<<

>>>>> OPERATING MODE

MOTORED CYCLE

>>>>> OPERATING CONDITIONS

ENGINE SPEED = 2052.0 RPM

>>>>> MANIFOLD CONDITIONS

INTAKE MANIFOLD PRESSURE = 0.9300 ATM

EXHAUST MANIFOLD PRESSURE = 1.0200 ATM

FRESH CHARGE TEMPERATURE = 312.70 K

EXHAUST GAS RECIRCULATION = 0.00 %

EGR TEMPERATURE	=	300.00	K
INTAKE CHARGE TEMPERATURE	=	312.70	K
ATMOSPHERIC PRESSURE	=	0.9810	ATM
ATMOSPHERIC TEMPERATURE	=	300.60	K

>>>>> HEAT TRANSFER AND TURBULENCE PARAMETERS

HEAT TRANSFER CONSTANT	=	0.3770	
HEAT TRANSFER EXPONENT	=	0.8000	
ROTOR TEMPERATURE	=	350.00	K
SIDE WALL TEMPERATURE	=	350.00	K
HOUSING WALL TEMPERATURE	=	350.00	K

>>>>> ENGINE DESIGN PARAMETERS

ECCENTRICITY OF ROTOR	=	1.500	CM
RADIUS OF ROTOR	=	10.500	CM
DEPTH OF CHAMBER	=	7.000	CM
COMPRESSION RATIO	=	9.407	
DISPLACED VOLUME	=	572.875	CC
VOLUME OF ROTOR POCKET	=	35.000	CC
INTAKE PORT OPENS	=	-530.0	DEG CA
INTAKE PORT CLOSES	=	-180.0	DEG CA
EXHAUST PORT OPENS	=	199.0	DEG CA
EXHAUST PORT CLOSES	=	588.5	DEG CA

```
>>>>> LEAKAGE AND CREVICE VOLUME PARAMETERS

          LEAK AREA PER APEX      =     0.010000  CM*CM

          CREVICE VOLUME PER APEX=      0.875000  CC

          CREVICE GAS TEMPERATURE=    350.000000  K
```

```
>>>>> COMPUTATIONAL PARAMETERS

          MAXIMUM # OF ITERATIONS =       2

          OUTPUT AT ITERATION #   =       2

          TPRINT                  =      10.00
          TPRINX                  =       1.00
          XBZERO                  =       0.00050
          XESTOP                  =       0.00000
          XBSTOP                  =       0.99500
          CIINTG                  =       0.000100
          CCINTG                  =       0.000100
          CBINTG                  =       0.000010
          CEINTG                  =       0.000100
          AREROT                  =       0.000200
          REL                     =       0.000200
          ERMAX                   =       0.000000
          MAXERR                  =       0.030000
          MAXTRY                  =       2
```

>>>>> START OF INTAKE PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MIN (G)	MEX (G)	VIV (CM/SEC)	VEV (CM/SEC)	X1 (-)	Q DOT (KJ/DEG)	WORK (KJ)	IMF	IFG
-530.0	1.0199	338.24	0.00000	0.00000	0.0		1.00000		0.000000		
-520.0	1.0041	338.50	-0.00441	-0.00912	-8634.8	-3616.3	1.00000	-0.000013	0.000594	0	2
-510.0	0.9586	335.86	-0.01480	-0.02593	-5606.4	-7223.5	1.00000	-0.000016	0.001544	0	2
-502.4	0.9301	334.50	-0.01960	-0.03786	-383.0	-8841.6	1.00000	-0.000018	0.002470	0	8
-500.0	0.9287	334.81	-0.01914	-0.04051	1218.4	-8920.2	1.00000	-0.000017	0.002797	0	2
-490.0	0.9131	334.92	-0.00932	-0.04473	4478.2	0.0	1.00000	-0.000018	0.004352	0	2
-484.6	0.9127	335.69	-0.00000	-0.04473	4545.4	0.0	1.00000	-0.000017	0.005301	0	8
-480.0	0.9126	334.96	0.00927	-0.04473	4307.2	0.0	1.00000	-0.000018	0.006191	1	2
-470.0	0.9146	333.18	0.03174	-0.04473	4065.3	0.0	1.00000	-0.000021	0.008307	1	2
-460.0	0.9163	331.76	0.05681	-0.04473	3832.6	0.0	1.00000	-0.000024	0.010674	1	2
-450.0	0.9187	330.72	0.08426	-0.04473	3488.5	0.0	1.00000	-0.000026	0.013260	1	2
-440.0	0.9197	329.82	0.11336	-0.04473	3325.8	0.0	1.00000	-0.000028	0.016030	1	2
-430.0	0.9210	329.16	0.14394	-0.04473	3110.3	0.0	1.00000	-0.000031	0.018948	1	2
-420.0	0.9220	328.65	0.17539	-0.04473	2948.2	0.0	1.00000	-0.000033	0.021973	1	2
-410.0	0.9237	328.37	0.20750	-0.04473	2611.9	0.0	1.00000	-0.000034	0.025066	1	2
-400.0	0.9235	328.06	0.23902	-0.04473	2649.3	0.0	1.00000	-0.000036	0.028182	1	2
-390.0	0.9237	327.91	0.27005	-0.04473	2618.1	0.0	1.00000	-0.000038	0.031278	1	2
-380.0	0.9239	327.87	0.30008	-0.04473	2561.6	0.0	1.00000	-0.000039	0.034312	1	2
-370.0	0.9245	327.94	0.32883	-0.04473	2441.1	0.0	1.00000	-0.000040	0.037244	1	2
-360.0	0.9254	328.10	0.35634	-0.04473	2228.2	0.0	1.00000	-0.000041	0.040034	1	2
-350.0	0.9249	328.20	0.38109	-0.04473	2352.4	0.0	1.00000	-0.000042	0.042643	1	2
-340.0	0.9252	328.43	0.40403	-0.04473	2284.2	0.0	1.00000	-0.000043	0.045036	1	2
-330.0	0.9258	328.72	0.42468	-0.04473	2144.8	0.0	1.00000	-0.000043	0.047180	1	2
-320.0	0.9264	329.07	0.44263	-0.04473	1979.3	0.0	1.00000	-0.000043	0.049047	1	2
-310.0	0.9273	329.50	0.45768	-0.04473	1715.8	0.0	1.00000	-0.000043	0.050611	1	2
-300.0	0.9279	329.96	0.46923	-0.04473	1525.2	0.0	1.00000	-0.000042	0.051850	1	2
-290.0	0.9289	330.53	0.47748	-0.04473	1074.6	0.0	1.00000	-0.000041	0.052749	1	2
-280.0	0.9301	331.17	0.48202	-0.04473	-278.9	0.0	1.00000	-0.000040	0.053294	1	2
-277.8	0.9308	331.37	0.48272	-0.04473	-947.9	0.0	1.00000	-0.000040	0.053364	1	8
-273.4	0.9299	331.55	0.48242	-0.04473	413.4	0.0	1.00000	-0.000039	0.053454	0	8
-270.0	0.9302	331.80	0.48220	-0.04473	-462.0	0.0	1.00000	-0.000039	0.053476	0	2
-260.0	0.9317	332.57	0.47935	-0.04473	-1413.2	0.0	1.00000	-0.000037	0.053293	0	2
-250.0	0.9365	333.63	0.47437	-0.04473	-2721.9	0.0	1.00000	-0.000035	0.052746	0	2
-240.0	0.9473	335.26	0.46865	-0.04473	-4416.5	0.0	1.00000	-0.000031	0.051835	0	2
-230.0	0.9711	338.09	0.46520	-0.04473	-6654.9	0.0	1.00000	-0.000025	0.050554	0	2
-220.0	1.0093	342.16	0.46429	-0.04473	-8933.8	0.0	1.00000	-0.000017	0.048883	0	2
-210.0	1.0589	347.04	0.46406	-0.04473	-10912.6	0.0	1.00000	-0.000006	0.046800	0	2
-200.0	1.1200	352.61	0.46401	-0.04473	-12594.7	0.0	1.00000	0.000006	0.044277	0	2
-190.0	1.1944	358.88	0.46400	-0.04473	-14014.3	0.0	1.00000	0.000020	0.041287	0	2
-180.0	1.2848	365.91	0.46400	-0.04473	0.0	0.0	1.00000	0.000037	0.037794	0	2
-180.0	1.2848	365.91	0.46400	-0.04473	0.0	0.0	1.00000	0.000037	0.037794	0	2

>>>>> START OF COMPRESSION PROCESS

CA (DEG)	P (ATM)	TEMP (K)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
-180.0	1.2848	365.91	0.000037	0.037794	2
-170.0	1.3942	373.72	0.000056	0.033758	2
-160.0	1.5267	382.34	0.000079	0.029131	2
-150.0	1.6876	391.80	0.000105	0.023861	2
-140.0	1.8836	402.16	0.000137	0.017886	2
-130.0	2.1233	413.42	0.000174	0.011138	2
-120.0	2.4178	425.60	0.000218	0.003545	2
-110.0	2.7815	438.68	0.000270	-0.004969	2
-100.0	3.2322	452.58	0.000333	-0.014474	2
-90.0	3.7922	467.16	0.000409	-0.025029	2
-80.0	4.4870	482.12	0.000499	-0.036655	2
-70.0	5.3434	497.00	0.000606	-0.049308	2
-60.0	6.3840	511.21	0.000730	-0.062830	2
-50.0	7.6055	523.48	0.000866	-0.076868	2
-40.0	8.9547	532.08	0.001000	-0.090791	2
-30.0	10.2957	535.21	0.001110	-0.103630	2
-20.0	11.3857	530.98	0.001163	-0.114121	2
-10.0	11.9209	518.17	0.001125	-0.120966	2
-7.9	11.9384	514.43	0.001105	-0.121833	8
0.0	11.6797	496.99	0.000987	-0.123302	2
10.0	10.6949	469.91	0.000776	-0.121128	2
20.0	9.2320	439.87	0.000541	-0.115321	2
30.0	7.6366	409.81	0.000326	-0.107209	2
40.0	6.1650	381.83	0.000155	-0.098053	2
50.0	4.9331	357.01	0.000030	-0.088761	2
60.0	3.9562	335.63	-0.000056	-0.079869	2
70.0	3.2018	317.53	-0.000113	-0.071636	2
80.0	2.6248	302.36	-0.000150	-0.064149	2
90.0	2.1837	289.73	-0.000174	-0.057405	2
100.0	1.8446	279.26	-0.000190	-0.051357	2
110.0	1.5819	270.65	-0.000199	-0.045944	2
120.0	1.3767	263.62	-0.000204	-0.041101	2
130.0	1.2150	257.96	-0.000207	-0.036768	2
140.0	1.0867	253.48	-0.000208	-0.032892	2
150.0	0.9846	250.08	-0.000208	-0.029426	2
160.0	0.9030	247.64	-0.000206	-0.026331	2
170.0	0.8376	246.03	-0.000204	-0.023574	2
180.0	0.7852	245.17	-0.000202	-0.021129	2
190.0	0.7436	244.97	-0.000199	-0.018975	2

>>>>> START OF EXHAUST PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MEX (G)	VEV (M/SEC)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
199.0	0.7138	245.32	-0.04473	0.0	-0.000196	-0.017271	2
200.0	0.7111	245.41	-0.04484	-15870.5	-0.000196	-0.017095	2
210.0	0.7079	248.57	-0.05765	-16086.6	-0.000192	-0.015460	2
220.0	0.7444	255.46	-0.09090	-14980.2	-0.000187	-0.014004	2
230.0	0.8134	265.00	-0.14083	-12689.4	-0.000179	-0.012695	2
240.0	0.9014	275.49	-0.19907	-9353.8	-0.000169	-0.011552	2
250.0	0.9751	283.99	-0.24569	-5634.1	-0.000158	-0.010644	2
260.0	1.0157	289.32	-0.26958	-1730.4	-0.000150	-0.010060	2
270.0	1.0208	291.62	-0.26919	751.2	-0.000144	-0.009860	2
280.0	1.0227	293.60	-0.26232	1374.9	-0.000139	-0.010061	2
290.0	1.0257	295.64	-0.25142	2001.9	-0.000133	-0.010661	2
300.0	1.0299	297.73	-0.23656	2621.3	-0.000127	-0.011656	2
310.0	1.0350	299.86	-0.21779	3227.8	-0.000120	-0.013036	2
320.0	1.0410	302.04	-0.19523	3814.9	-0.000113	-0.014788	2
330.0	1.0479	304.25	-0.16903	4376.0	-0.000106	-0.016893	2
340.0	1.0552	306.47	-0.13936	4901.1	-0.000099	-0.019329	2
350.0	1.0627	308.68	-0.10649	5378.5	-0.000092	-0.022067	2
360.0	1.0700	310.83	-0.07076	5796.0	-0.000085	-0.025074	2
370.0	1.0765	312.88	-0.03260	6143.7	-0.000079	-0.028311	2
380.0	1.0818	314.80	0.00751	6413.0	-0.000072	-0.031734	2
390.0	1.0856	316.57	0.04901	6603.0	-0.000066	-0.035294	2
400.0	1.0878	318.17	0.09136	6712.4	-0.000061	-0.038937	2
410.0	1.0880	319.61	0.13398	6737.1	-0.000056	-0.042610	2
420.0	1.0862	320.90	0.17627	6673.5	-0.000051	-0.046254	2
430.0	1.0825	322.06	0.21760	6518.0	-0.000047	-0.049816	2
440.0	1.0769	323.11	0.25734	6264.9	-0.000043	-0.053241	2
450.0	1.0699	324.11	0.29489	5913.8	-0.000040	-0.056479	2
460.0	1.0619	325.12	0.32965	5471.8	-0.000037	-0.059485	2
470.0	1.0536	326.21	0.36114	4950.3	-0.000034	-0.062220	2
480.0	1.0448	327.37	0.38905	4295.5	-0.000031	-0.064652	2
490.0	1.0367	328.72	0.41299	3557.1	-0.000028	-0.066752	2
500.0	1.0326	330.55	0.43254	3109.9	-0.000025	-0.068499	2
510.0	1.0276	332.38	0.44801	2433.7	-0.000022	-0.069875	2
520.0	1.0234	334.30	0.45920	1639.4	-0.000019	-0.070867	2
530.0	1.0210	336.32	0.46605	901.1	-0.000016	-0.071466	2
540.0	1.0203	338.29	0.46862	503.4	-0.000014	-0.071667	2
550.0	1.0194	339.96	0.46704	-693.7	-0.000012	-0.071467	2

```
>--------------+--------------<
>                             <
>     CALCULATION RESULTS     <
>                             <
>--------------+--------------<
```

--> VOLUMETRIC EFFICIENCY; (%) BASED ON: INTAKE / ATM	---->	77.5	70.6
--> PUMPING MEAN EFFECTIVE PRESSURE; (KPA) : PEMP	---->	-14.	
--> GROSS INDICATED MEAN EFFECTIVE PRESSURE; (KPA) : IMEP	---->	-111.	
--> GROSS INDICATED SPECIFIC FUEL CONSUMPTION; (G/IKW-HR) : ISFC	---->	0.	
--> GROSS INDICATED THERMAL EFFICIENCY; (%)	---->	0.0	
--> NET INDICATED THERMAL EFFICIENCY; (%)	---->	0.0	
--> (HEAT TRANSFER PER CYCLE)/ (MASS OF FUEL TIMES LHV); (%)	---->	0.0	
--> IGNITION DELAY (0 - 10%) (CRANK ANGLE) / (MS)	---->	0.00	0.00
--> BURN DURATION (10 - 90%) (CRANK ANGLE) / (MS)	---->	0.00	0.00
--> MEAN EXHAUST TEMPERATURE; (K)	---->	338.1	

```
MASS IN CYLINDER AT TIVO =  0.07427 G

MASS IN CYLINDER AT TIVC =  0.61433 G

MASS OF FUEL INDUCTED    =  0.00000 G

RESIDUAL FRACTION        =  0.00000
```

```
HEATI   =   -0.008384 KJ  (TIPO   -   -270)
WORKI   =    0.053476 KJ

HEATCE  =    0.077640 KJ  (TIPC   -    +270
WORKCE  =   -0.063336 KJ

HEATE   =   -0.018311 KJ  (+270   -   TIPO)
WORKE   =   -0.061607 KJ
```

```
TOTAL ENTHALPY IN / CYCLE    =  0.14620 KJ

TOTAL ENTHALPY OUT / CYCLE   =  0.15920 KJ

TOTAL HEAT LOSS / CYCLE      =  0.05094 KJ

TOTAL WORK OUTPUT / CYCLE    = -0.07147 KJ

HEAT LOSS TO CREVICE/CYCLE   =  0.00704 KJ

"LOST" FUEL ENERGY           =  0.00000 KJ

NET ENERGY GAIN / CYCLE      = -0.00048 KJ

(ENERGY GAIN)/(ENTHALPY IN) = -0.33137 %

(ENERGY GAIN)/(MFUEL*LHV)    =  0.00000 %
```

LEAKAGE AND CREVICE VOLUME DATA

CA (DEG)	CHAMBER MASS (G)	LEAD CREVICE MASS (G)	LAG CREVICE MASS (G)	LEAD LEAKAGE MASS ( )	LAG LEAKAGE MASS ( )	LEAD CREVICE COMPOSITION	LAG CREVICE COMPOSITION
-520.0	0.0791	0.000000	0.000882	-0.000243	0.000150	0.000000	1.000000
-510.0	0.0857	0.000000	0.000842	-0.000525	0.000296	0.000000	1.000000
-502.4	0.0930	0.000000	0.000817	-0.000764	0.000394	0.000000	1.000000
-500.0	0.0961	0.000000	0.000816	-0.000843	0.000422	0.000000	1.000000
-490.0	0.1105	0.000000	0.000802	-0.001201	0.000521	0.000000	1.000000
-484.6	0.1200	0.000000	0.000802	-0.001413	0.000553	0.000000	1.000000
-480.0	0.1294	0.000000	0.000000	-0.001608	0.000545	0.000000	0.000000
-470.0	0.1525	0.000000	0.000000	-0.002076	0.000465	0.000000	0.000000
-460.0	0.1782	0.000000	0.000000	-0.002618	0.000357	0.000000	0.000000
-450.0	0.2064	0.000000	0.000000	-0.003255	0.000245	0.000000	0.000000
-440.0	0.2363	0.000000	0.000000	-0.004009	0.000133	0.000000	0.000000
-430.0	0.2679	0.000000	0.000000	-0.004911	0.000020	0.000000	0.000000
-420.0	0.3006	0.000000	0.000000	-0.005995	-0.000094	0.000000	0.000000
-410.0	0.3341	0.000000	0.000000	-0.007302	-0.000210	0.000000	0.000000
-400.0	0.3673	0.000000	0.000000	-0.008868	-0.000328	0.000000	0.000000
-390.0	0.4003	0.000000	0.000000	-0.010714	-0.000450	0.000000	0.000000
-380.0	0.4326	0.000000	0.000000	-0.012824	-0.000574	0.000000	0.000000
-370.0	0.4638	0.000000	0.000000	-0.015124	-0.000701	0.000000	0.000000
-360.0	0.4935	0.000000	0.000000	-0.017236	-0.000831	0.000000	0.000000
-350.0	0.5206	0.000000	0.000000	-0.019478	-0.000965	0.000000	0.000000
-340.0	0.5457	0.000000	0.000000	-0.021473	-0.001100	0.000000	0.000000
-330.0	0.5682	0.000000	0.000000	-0.023158	-0.001238	0.000000	0.000000
-320.0	0.5876	0.000000	0.000000	-0.024530	-0.001378	0.000000	0.000000
-310.0	0.6039	0.000000	0.000000	-0.025629	-0.001518	0.000000	0.000000
-300.0	0.6165	0.000000	0.000000	-0.026508	-0.001658	0.000000	0.000000
-290.0	0.6256	0.000000	0.000000	-0.027214	-0.001797	0.000000	0.000000
-280.0	0.6308	0.000000	0.000000	-0.027788	-0.001933	0.000000	0.000000
-277.8	0.6317	0.000000	0.000000	-0.027898	-0.001962	0.000000	0.000000
-273.4	0.6316	0.000000	0.000000	-0.028108	-0.002021	0.000000	0.000000
-270.0	0.6316	0.000000	0.000000	-0.028261	-0.002066	0.000000	0.000000
-260.0	0.6293	0.000000	0.000000	-0.028657	-0.002196	0.000000	0.000000
-250.0	0.6248	0.000000	0.000000	-0.028994	-0.002321	0.000000	0.000000
-240.0	0.6195	0.000000	0.000000	-0.029284	-0.002438	0.000000	0.000000
-230.0	0.6164	0.000000	0.000000	-0.029527	-0.002542	0.000000	0.000000
-220.0	0.6157	0.000000	0.000000	-0.029719	-0.002619	0.000000	0.000000
-210.0	0.6156	0.000000	0.000930	-0.029841	-0.002608	0.000000	1.000000
-200.0	0.6153	0.000984	0.000984	-0.029787	-0.002516	1.000000	1.000000
-190.0	0.6149	0.001050	0.001050	-0.029632	-0.002382	1.000000	1.000000
-180.0	0.6143	0.001129	0.001129	-0.029433	-0.002211	1.000000	1.000000

-170.0	0.6137	0.001225	0.001225	-0.029201	-0.002005	1.000000	1.000000
-160.0	0.6130	0.001342	0.001342	-0.028944	-0.001764	1.000000	1.000000
-150.0	0.6121	0.001483	0.001483	-0.028659	-0.001484	1.000000	1.000000
-140.0	0.6111	0.001655	0.001655	-0.028343	-0.001169	1.000000	1.000000
-130.0	0.6100	0.001866	0.001866	-0.027989	-0.000814	1.000000	1.000000
-120.0	0.6087	0.002125	0.002125	-0.027588	-0.000413	1.000000	1.000000
-110.0	0.6071	0.002444	0.002444	-0.027128	0.000047	1.000000	1.000000
-100.0	0.6053	0.002840	0.002840	-0.026597	0.000578	1.000000	1.000000
-90.0	0.6031	0.003332	0.003332	-0.025976	0.001199	1.000000	1.000000
-80.0	0.6004	0.003943	0.003943	-0.025245	0.001930	1.000000	1.000000
-70.0	0.5971	0.004695	0.004695	-0.024376	0.002798	1.000000	1.000000
-60.0	0.5932	0.005610	0.005610	-0.023340	0.003835	1.000000	1.000000
-50.0	0.5886	0.006683	0.006683	-0.022103	0.005072	1.000000	1.000000
-40.0	0.5833	0.007868	0.007868	-0.020637	0.006538	1.000000	1.000000
-30.0	0.5775	0.009047	0.009047	-0.018930	0.008245	1.000000	1.000000
-20.0	0.5718	0.010005	0.010005	-0.017003	0.010172	1.000000	1.000000
-10.0	0.5667	0.010475	0.010475	-0.014928	0.012247	1.000000	1.000000
-7.9	0.5658	0.010490	0.010490	-0.014491	0.012684	1.000000	1.000000
0.0	0.5629	0.010263	0.010263	-0.012825	0.014350	1.000000	1.000000
10.0	0.5606	0.009397	0.009397	-0.010833	0.016342	1.000000	1.000000
20.0	0.5597	0.008112	0.008112	-0.009063	0.018112	1.000000	1.000000
30.0	0.5595	0.006710	0.006710	-0.007569	0.019606	1.000000	1.000000
40.0	0.5596	0.005417	0.005417	-0.006349	0.020826	1.000000	1.000000
50.0	0.5598	0.004335	0.004335	-0.005370	0.021805	1.000000	1.000000
60.0	0.5600	0.003476	0.003476	-0.004586	0.022589	1.000000	1.000000
70.0	0.5600	0.002813	0.002813	-0.003954	0.023220	1.000000	1.000000
80.0	0.5600	0.002306	0.002306	-0.003440	0.023734	1.000000	1.000000
90.0	0.5600	0.001919	0.001919	-0.003016	0.024159	1.000000	1.000000
100.0	0.5598	0.001621	0.001621	-0.002661	0.024514	1.000000	1.000000
110.0	0.5597	0.001390	0.001390	-0.002364	0.024816	1.000000	1.000000
120.0	0.5596	0.001210	0.001210	-0.002124	0.025070	1.000000	1.000000
130.0	0.5595	0.001068	0.001068	-0.001939	0.025274	1.000000	1.000000
140.0	0.5594	0.000955	0.000955	-0.001816	0.025417	1.000000	1.000000
150.0	0.5595	0.000000	0.000000	-0.001790	0.025445	0.000000	0.000000
160.0	0.5598	0.000000	0.000000	-0.001916	0.025280	0.000000	0.000000
170.0	0.5601	0.000000	0.000000	-0.002047	0.025101	0.000000	0.000000
180.0	0.5604	0.000000	0.000000	-0.002197	0.024889	0.000000	0.000000
190.0	0.5608	0.000653	0.000000	-0.002357	0.024653	1.000000	0.000000

200.0	0.5614	0.000000	0.000000	-0.002522	0.024395	0.000000	0.000000
210.0	0.5746	0.000000	0.000000	-0.002682	0.024111	0.000000	0.000000
220.0	0.6083	0.000000	0.000000	-0.002824	0.023795	0.000000	0.000000
230.0	0.6587	0.000000	0.000000	-0.002946	0.023440	0.000000	0.000000
240.0	0.7173	0.000000	0.000000	-0.002948	0.023039	0.000000	0.000000
250.0	0.7643	0.000857	0.000000	-0.002901	0.022579	1.000000	0.000000
260.0	0.7886	0.000892	0.000000	-0.002800	0.022048	1.000000	0.000000
270.0	0.7887	0.000897	0.000000	-0.002689	0.021427	1.000000	0.000000
280.0	0.7825	0.000899	0.000000	-0.002576	0.020696	1.000000	0.000000
290.0	0.7723	0.000901	0.000000	-0.002463	0.019828	1.000000	0.000000
300.0	0.7584	0.000905	0.000000	-0.002349	0.018792	1.000000	0.000000
310.0	0.7407	0.000909	0.000000	-0.002233	0.017555	1.000000	0.000000
320.0	0.7195	0.000915	0.000000	-0.002114	0.016089	1.000000	0.000000
330.0	0.6949	0.000921	0.000000	-0.001992	0.014382	1.000000	0.000000
340.0	0.6670	0.000927	0.000000	-0.001867	0.012455	1.000000	0.000000
350.0	0.6361	0.000934	0.000000	-0.001738	0.010380	1.000000	0.000000
360.0	0.6023	0.000940	0.000000	-0.001607	0.008276	1.000000	0.000000
370.0	0.5660	0.000946	0.000000	-0.001472	0.006284	1.000000	0.000000
380.0	0.5275	0.000951	0.000000	-0.001334	0.004514	1.000000	0.000000
390.0	0.4874	0.000954	0.000000	-0.001195	0.003020	1.000000	0.000000
400.0	0.4461	0.000956	0.000000	-0.001054	0.001800	1.000000	0.000000
410.0	0.4043	0.000956	0.000000	-0.000914	0.000821	1.000000	0.000000
420.0	0.3627	0.000954	0.000000	-0.000774	0.000037	1.000000	0.000000
430.0	0.3219	0.000951	0.000000	-0.000635	-0.000594	1.000000	0.000000
440.0	0.2825	0.000946	0.000000	-0.000499	-0.001108	1.000000	0.000000
450.0	0.2452	0.000940	0.000000	-0.000365	-0.001533	1.000000	0.000000
460.0	0.2107	0.000933	0.000000	-0.000236	-0.001888	1.000000	0.000000
470.0	0.1794	0.000926	0.000000	-0.000112	-0.002185	1.000000	0.000000
480.0	0.1516	0.000918	0.000000	0.000005	-0.002425	1.000000	0.000000
490.0	0.1278	0.000911	0.000000	0.000108	-0.002610	1.000000	0.000000
500.0	0.1083	0.000907	0.000000	0.000185	-0.002733	1.000000	0.000000
510.0	0.0928	0.000000	0.000903	0.000176	-0.002740	0.000000	1.000000
520.0	0.0816	0.000000	0.000899	0.000084	-0.002639	0.000000	1.000000
530.0	0.0748	0.000000	0.000897	-0.000051	-0.002507	0.000000	1.000000
540.0	0.0722	0.000000	0.000897	-0.000222	-0.002357	0.000000	1.000000
550.0	0.0739	0.000000	0.000896	-0.000428	-0.002197	0.000000	1.000000

HEAT TRANSFER DATA

CA (DEG)	VELHTX (CM/SEC)	HTPARO (KW/M**2)	HTPASI (KW/M**2)	HTPAHO (KW/M**2)	Q% ROTOR (%)	Q% SIDE (%)	Q% HOUSING (%)
-520.0	564.1	-5059139.0	-5059139.0	-5059139.0	43.831	3.556	52.613
-510.0	564.1	-6018203.0	-6018203.0	-6018203.0	43.160	4.320	52.520
-502.4	564.1	-6453169.0	-6453169.0	-6453169.0	42.499	5.074	52.427
-500.0	564.1	-6314220.0	-6314220.0	-6314220.0	42.267	5.338	52.395
-490.0	564.1	-6184264.0	-6184264.0	-6184264.0	41.191	6.564	52.245
-484.6	564.1	-5859177.0	-5859177.0	-5859177.0	40.555	7.289	52.156
-480.0	564.1	-6162520.0	-6162520.0	-6162520.0	39.976	7.949	52.075
-470.0	564.1	-6922887.0	-6922887.0	-6922887.0	38.665	9.443	51.892
-460.0	564.1	-7534995.0	-7534995.0	-7534995.0	37.299	10.999	51.702
-450.0	564.1	-7996087.0	-7996087.0	-7996087.0	35.916	12.576	51.508
-440.0	564.1	-8389617.0	-8389617.0	-8389617.0	34.546	14.137	51.317
-430.0	564.1	-8682727.0	-8682727.0	-8682727.0	33.213	15.656	51.131
-420.0	564.1	-8909711.0	-8909711.0	-8909711.0	31.937	17.110	50.953
-410.0	564.1	-9043981.0	-9043981.0	-9043981.0	30.732	18.483	50.785
-400.0	564.1	-9177048.0	-9177048.0	-9177048.0	29.608	19.764	50.628
-390.0	564.1	-9244900.0	-9244900.0	-9244900.0	28.569	20.948	50.483
-380.0	564.1	-9263492.0	-9263492.0	-9263492.0	27.620	22.029	50.350
-370.0	564.1	-9235696.0	-9235696.0	-9235696.0	26.762	23.008	50.231
-360.0	564.1	-9174860.0	-9174860.0	-9174860.0	25.993	23.883	50.123
-350.0	564.1	-9126136.0	-9126136.0	-9126136.0	25.314	24.658	50.029
-340.0	564.1	-9030477.0	-9030477.0	-9030477.0	24.721	25.333	49.946
-330.0	564.1	-8906449.0	-8906449.0	-8906449.0	24.214	25.911	49.875
-320.0	564.1	-8759863.0	-8759863.0	-8759863.0	23.788	26.396	49.816
-310.0	564.1	-8581991.0	-8581991.0	-8581991.0	23.444	26.789	49.767
-300.0	564.1	-8388633.0	-8388633.0	-8388633.0	23.178	27.092	49.730
-290.0	564.1	-8151094.0	-8151094.0	-8151094.0	22.989	27.307	49.704
-280.0	564.1	-7880011.0	-7880011.0	-7880011.0	22.876	27.436	49.688
-277.8	564.1	-7800075.0	-7800075.0	-7800075.0	22.862	27.452	49.686
-273.4	564.1	-7714181.0	-7714181.0	-7714181.0	22.843	27.473	49.684
-270.0	564.1	-7608773.0	-7608773.0	-7608773.0	22.839	27.478	49.683
-260.0	564.1	-7288764.0	-7288764.0	-7288764.0	22.876	27.436	49.688
-250.0	564.1	-6862094.0	-6862094.0	-6862094.0	22.989	27.307	49.704
-240.0	564.1	-6218656.0	-6218656.0	-6218656.0	23.178	27.092	49.730
-230.0	564.1	-5102061.0	-5102061.0	-5102061.0	23.444	26.789	49.767
-220.0	564.1	-3442358.0	-3442358.0	-3442358.0	23.789	26.396	49.816
-210.0	564.1	-1340632.0	-1340632.0	-1340632.0	24.214	25.911	49.875
-200.0	564.1	1224533.0	1224533.0	1224533.0	24.721	25.333	49.946
-190.0	564.1	4353339.0	4353339.0	4353339.0	25.314	24.658	50.028
-180.0	564.1	8179467.0	8179467.0	8179467.0	25.993	23.883	50.123

-170.0	564.1	12870825.0	12870825.0	12870825.0	26.762	23.008	50.231
-160.0	564.1	18643696.0	18643696.0	18643696.0	27.620	22.029	50.350
-150.0	564.1	25776912.0	25776912.0	25776912.0	28.569	20.948	50.483
-140.0	564.1	34629392.0	34629392.0	34629392.0	29.608	19.764	50.628
-130.0	564.1	45662688.0	45662688.0	45662688.0	30.732	18.483	50.785
-120.0	564.1	59461360.0	59461360.0	59461360.0	31.937	17.110	50.953
-110.0	564.1	76761136.0	76761136.0	76761136.0	33.213	15.656	51.131
-100.0	564.1	98462144.0	98462144.0	98462144.0	34.546	14.137	51.317
-90.0	564.1	125611712.0	125611712.0	125611712.0	35.916	12.576	51.508
-80.0	564.1	159318048.0	159318048.0	159318048.0	37.299	10.999	51.702
-70.0	564.1	200505376.0	200505376.0	200505376.0	38.665	9.443	51.892
-60.0	564.1	249661104.0	249661104.0	249661104.0	39.976	7.949	52.075
-50.0	564.1	305065216.0	305065216.0	305065216.0	41.191	6.564	52.245
-40.0	564.1	361635584.0	361635584.0	361635584.0	42.267	5.338	52.395
-30.0	564.1	409966080.0	409966080.0	409966080.0	43.160	4.320	52.520
-20.0	564.1	436080384.0	436080384.0	436080384.0	43.831	3.556	52.613
-10.0	564.1	426039552.0	426039552.0	426039552.0	44.247	3.082	52.671
-7.9	564.1	418727424.0	418727424.0	418727424.0	44.299	3.022	52.679
0.0	564.1	374782464.0	374782464.0	374782464.0	44.388	2.921	52.691
10.0	564.1	293739264.0	293739264.0	293739264.0	44.247	3.082	52.671
20.0	564.1	202832752.0	202832752.0	202832752.0	43.831	3.556	52.613
30.0	564.1	120465424.0	120465424.0	120465424.0	43.160	4.320	52.520
40.0	564.1	56096240.0	56096240.0	56096240.0	42.267	5.338	52.395
50.0	564.1	10708911.0	10708911.0	10708911.0	41.191	6.564	52.245
60.0	564.1	-19016128.0	-19016128.0	-19016128.0	39.976	7.949	52.075
70.0	564.1	-37354928.0	-37354928.0	-37354928.0	38.665	9.443	51.892
80.0	564.1	-47979280.0	-47979280.0	-47979280.0	37.299	10.999	51.702
90.0	564.1	-53587040.0	-53587040.0	-53587040.0	35.916	12.576	51.508
100.0	564.1	-56024208.0	-56024208.0	-56024208.0	34.546	14.137	51.317
110.0	564.1	-56503776.0	-56503776.0	-56503776.0	33.213	15.656	51.131
120.0	564.1	-55806512.0	-55806512.0	-55806512.0	31.937	17.110	50.953
130.0	564.1	-54430048.0	-54430048.0	-54430048.0	30.732	18.483	50.785
140.0	564.1	-52686608.0	-52686608.0	-52686608.0	29.608	19.764	50.628
150.0	564.1	-50765264.0	-50765264.0	-50765264.0	28.569	20.948	50.483
160.0	564.1	-48779344.0	-48779344.0	-48779344.0	27.620	22.029	50.350
170.0	564.1	-46814512.0	-46814512.0	-46814512.0	26.762	23.008	50.231
180.0	564.1	-44912752.0	-44912752.0	-44912752.0	25.993	23.883	50.123
190.0	564.1	-43095360.0	-43095360.0	-43095360.0	25.314	24.658	50.028

200.0	564.1	-41370096.0	-41370096.0	-41370096.0	24.721	25.333	49.946
210.0	564.1	-39704304.0	-39704304.0	-39704304.0	24.214	25.911	49.875
220.0	564.1	-37973472.0	-37973472.0	-37973472.0	23.789	26.396	49.816
230.0	564.1	-35949264.0	-35949264.0	-35949264.0	23.444	26.789	49.767
240.0	564.1	-33516192.0	-33516192.0	-33516192.0	23.178	27.092	49.730
250.0	564.1	-31117408.0	-31117408.0	-31117408.0	22.989	27.307	49.704
260.0	564.1	-29266384.0	-29266384.0	-29266384.0	22.876	27.436	49.688
270.0	564.1	-28153088.0	-28153088.0	-28153088.0	22.839	27.478	49.683
280.0	564.1	-27137568.0	-27137568.0	-27137568.0	22.876	27.436	49.688
290.0	564.1	-26124288.0	-26124288.0	-26124288.0	22.989	27.307	49.704
300.0	564.1	-25109072.0	-25109072.0	-25109072.0	23.178	27.092	49.730
310.0	564.1	-24088992.0	-24088992.0	-24088992.0	23.444	26.789	49.767
320.0	564.1	-23062368.0	-23062368.0	-23062368.0	23.788	26.396	49.816
330.0	564.1	-22029664.0	-22029664.0	-22029664.0	24.214	25.911	49.875
340.0	564.1	-20995408.0	-20995408.0	-20995408.0	24.721	25.333	49.946
350.0	564.1	-19968576.0	-19968576.0	-19968576.0	25.314	24.658	50.029
360.0	564.1	-18963264.0	-18963264.0	-18963264.0	25.993	23.883	50.123
370.0	564.1	-17994576.0	-17994576.0	-17994576.0	26.762	23.008	50.231
380.0	564.1	-17077344.0	-17077344.0	-17077344.0	27.620	22.029	50.350
390.0	564.1	-16219000.0	-16219000.0	-16219000.0	28.569	20.948	50.483
400.0	564.1	-15423043.0	-15423043.0	-15423043.0	29.608	19.764	50.628
410.0	564.1	-14691016.0	-14691016.0	-14691016.0	30.732	18.483	50.785
420.0	564.1	-14019017.0	-14019017.0	-14019017.0	31.937	17.110	50.953
430.0	564.1	-13399150.0	-13399150.0	-13399150.0	33.213	15.656	51.131
440.0	564.1	-12819410.0	-12819410.0	-12819410.0	34.546	14.137	51.317
450.0	564.1	-12260011.0	-12260011.0	-12260011.0	35.916	12.576	51.508
460.0	564.1	-11693709.0	-11693709.0	-11693709.0	37.299	10.999	51.702
470.0	564.1	-11092387.0	-11092387.0	-11092387.0	38.665	9.443	51.892
480.0	564.1	-10461035.0	-10461035.0	-10461035.0	39.976	7.949	52.075
490.0	564.1	-9754812.0	-9754812.0	-9754812.0	41.191	6.564	52.245
500.0	564.1	-8860437.0	-8860437.0	-8860437.0	42.267	5.338	52.395
510.0	564.1	-7973377.0	-7973377.0	-7973377.0	43.160	4.320	52.520
520.0	564.1	-7057017.0	-7057017.0	-7057017.0	43.831	3.556	52.613
530.0	564.1	-6119823.0	-6119823.0	-6119823.0	44.247	3.082	52.671
540.0	564.1	-5217056.0	-5217056.0	-5217056.0	44.388	2.921	52.691
550.0	564.1	-4458532.0	-4458532.0	-4458532.0	44.247	3.082	52.671

SAMPLE FIRING OUTPUT

M.I.T. ZERO-DIMENSIONAL WANKEL ENGINE CYCLE SIMULATION

>>>>> INPUT DATA <<<<<

>>>>> OPERATING MODE

FIRING CYCLE

SPECIFIED BURN RATE

BURN DURATION	=	90.000 DEG CA
WIEBE CONSTANT	=	5.000
WIEBE EXPONENT	=	1.500

>>>>> OPERATING CONDITIONS

FUEL USED IS ISOOCTANE

F/A EQUIVALENCE RATIO	=	1.000
SPARK TIMING	=	-30.00 DEG CA
ENGINE SPEED	=	3000.0 RPM

>>>>> MANIFOLD CONDITIONS

INTAKE MANIFOLD PRESSURE	=	0.9800	ATM
EXHAUST MANIFOLD PRESSURE	=	1.0200	ATM
FRESH CHARGE TEMPERATURE	=	300.00	K
EXHAUST GAS RECIRCULATION	=	0.00	%
EGR TEMPERATURE	=	300.00	K
INTAKE CHARGE TEMPERATURE	=	300.00	K
ATMOSPHERIC PRESSURE	=	1.0000	ATM
ATMOSPHERIC TEMPERATURE	=	300.00	K

>>>>> HEAT TRANSFER AND TURBULENCE PARAMETERS

HEAT TRANSFER CONSTANT	=	0.0350	
HEAT TRANSFER EXPONENT	=	0.8000	
ROTOR TEMPERATURE	=	370.00	K
SIDE WALL TEMPERATURE	=	370.00	K
HOUSING WALL TEMPERATURE	=	370.00	K

>>>>> ENGINE DESIGN PARAMETERS

ECCENTRICITY OF ROTOR	=	1.500	CM
RADIUS OF ROTOR	=	10.500	CM
DEPTH OF CHAMBER	=	7.000	CM
COMPRESSION RATIO	=	9.407	
DISPLACED VOLUME	=	572.875	CC

```
VOLUME OF ROTOR POCKET  =    35.000 CC

INTAKE PORT OPENS       =  -530.0   DEG CA

INTAKE PORT CLOSES      =  -180.0   DEG CA

EXHAUST PORT OPENS      =   199.0   DEG CA

EXHAUST PORT CLOSES     =   588.5   DEG CA
```

```
>>>>> LEAKAGE AND CREVICE VOLUME PARAMETERS

LEAK AREA PER APEX      =     0.010000  CM*CM

CREVICE VOLUME PER APEX=      0.875000  CC

CREVICE GAS TEMPERATURE=    370.000000  K
```

```
>>>>> COMPUTATIONAL PARAMETERS

MAXIMUM # OF ITERATIONS =      3

OUTPUT AT ITERATION #   =      3

TPRINT                  =     10.00
TPRINX                  =      1.00
XBZERO                  =      0.00030
XESTOP                  =      0.00000
XBSTOP                  =      0.99500
CIINTG                  =      0.000100
CCINTG                  =      0.000100
CBINTG                  =      0.000050
CEINTG                  =      0.000100
AREROT                  =      0.000200
REL                     =      0.000200
ERMAX                   =      0.000000
MAXERR                  =      0.030000
MAXTRY                  =      2
```

>>>>> START OF INTAKE PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MIN (G)	MEX (G)	VIV (CM/SEC)	VEV (CM/SEC)	X1 (-)	Q DOT (KJ/DEG)	WORK (KJ)	IMF	IFG
-530.0	1.0198	871.27	0.00000	0.00000	0.0		0.00000		0.000000		
-520.0	1.0092	851.10	-0.00125	-0.00336	-8820.2	-4728.4	0.01106	0.000029	0.000595	0	2
-510.0	0.9847	830.53	-0.00376	-0.00917	-3571.3	-8531.4	0.02064	0.000028	0.001559	0	2
-505.8	0.9798	823.76	-0.00408	-0.01147	702.7	-9076.9	0.02428	0.000028	0.002068	0	8
-500.0	0.9755	815.63	-0.00316	-0.01376	3523.3	-9527.2	0.02893	0.000028	0.002868	0	2
-492.9	0.9658	805.91	-0.00000	-0.01507	6222.7	-10493.0	0.03413	0.000028	0.003998	0	8
-490.0	0.9516	775.32	0.00326	-0.01511	5103.4	0.0	0.08742	0.000026	0.004501	1	2
-480.0	0.9371	635.90	0.02393	-0.01511	6217.7	0.0	0.35817	0.000020	0.006397	1	2
-470.0	0.9397	547.43	0.04991	-0.01511	6037.6	0.0	0.53119	0.000015	0.008570	1	2
-460.0	0.9432	490.67	0.07938	-0.01511	5778.4	0.0	0.64043	0.000011	0.011004	1	2
-450.0	0.9474	452.71	0.11177	-0.01511	5451.3	0.0	0.71344	0.000009	0.013670	1	2
-440.0	0.9510	426.05	0.14645	-0.01511	5152.4	0.0	0.76448	0.000006	0.016533	1	2
-430.0	0.9543	406.65	0.18296	-0.01511	4861.0	0.0	0.80159	0.000004	0.019554	1	2
-420.0	0.9579	392.15	0.22091	-0.01511	4518.3	0.0	0.82946	0.000003	0.022694	1	2
-410.0	0.9615	381.09	0.25966	-0.01511	4142.8	0.0	0.85079	0.000001	0.025910	1	2
-400.0	0.9619	372.43	0.29759	-0.01511	4091.8	0.0	0.86699	0.000000	0.029156	1	2
-390.0	0.9626	365.62	0.33504	-0.01511	4019.5	0.0	0.87981	-0.000001	0.032381	1	2
-380.0	0.9637	360.23	0.37159	-0.01511	3886.4	0.0	0.89011	-0.000002	0.035544	1	2
-370.0	0.9656	356.02	0.40653	-0.01511	3665.1	0.0	0.89841	-0.000002	0.038604	1	2
-360.0	0.9680	352.79	0.43892	-0.01511	3352.9	0.0	0.90486	-0.000003	0.041520	1	2
-350.0	0.9696	350.32	0.46756	-0.01511	3119.8	0.0	0.90931	-0.000003	0.044252	1	2
-340.0	0.9707	348.45	0.49241	-0.01511	2953.8	0.0	0.91193	-0.000004	0.046760	1	2
-330.0	0.9719	347.05	0.51402	-0.01511	2750.0	0.0	0.91327	-0.000004	0.049010	1	2
-320.0	0.9730	345.98	0.53241	-0.01511	2566.1	0.0	0.91392	-0.000004	0.050969	1	2
-310.0	0.9743	345.21	0.54781	-0.01511	2319.2	0.0	0.91431	-0.000005	0.052612	1	2
-300.0	0.9761	344.72	0.56019	-0.01511	1923.9	0.0	0.91453	-0.000005	0.053915	1	2
-290.0	0.9776	344.46	0.56884	-0.01511	1497.6	0.0	0.91449	-0.000005	0.054860	1	2
-280.0	0.9799	344.46	0.57409	-0.01511	234.6	0.0	0.91420	-0.000005	0.055433	1	2
-277.3	0.9801	344.50	0.57452	-0.01511	-389.2	0.0	0.91403	-0.000005	0.055523	1	8
-270.0	0.9804	344.66	0.57402	-0.01511	-625.9	0.0	0.91344	-0.000005	0.055626	0	2
-260.0	0.9828	345.01	0.57133	-0.01511	-1753.5	0.0	0.91276	-0.000005	0.055433	0	2
-250.0	0.9895	345.73	0.56695	-0.01511	-3191.0	0.0	0.91218	-0.000005	0.054855	0	2
-240.0	1.0030	347.05	0.56212	-0.01511	-4907.7	0.0	0.91168	-0.000004	0.053891	0	2
-230.0	1.0293	349.47	0.55949	-0.01511	-7020.5	0.0	0.91125	-0.000004	0.052534	0	2
-220.0	1.0690	352.96	0.55879	-0.01511	-9104.5	0.0	0.91090	-0.000003	0.050764	0	2
-210.0	1.1206	357.32	0.55862	-0.01511	-10950.4	0.0	0.91058	-0.000003	0.048558	0	2
-200.0	1.1850	362.52	0.55859	-0.01511	-12549.1	0.0	0.91028	-0.000002	0.045889	0	2
-190.0	1.2642	368.59	0.55858	-0.01511	-13904.5	0.0	0.90998	-0.000000	0.042725	0	2
-180.0	1.3610	375.59	0.55858	-0.01511	0.0	0.0	0.90968	0.000001	0.039026	0	2
-180.0	1.3610	375.59	0.55858	-0.01511	0.0	0.0	0.90968	0.000001	0.039026	0	2

>>>>> START OF COMPRESSION PROCESS

CA (DEG)	P (ATM)	TEMP (K)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
-180.0	1.3610	375.59	0.000001	0.039026	2
-170.0	1.4790	383.60	0.000003	0.034747	2
-160.0	1.6231	392.69	0.000005	0.029834	2
-150.0	1.7994	402.96	0.000008	0.024223	2
-140.0	2.0160	414.52	0.000011	0.017840	2
-130.0	2.2835	427.49	0.000015	0.010600	2
-120.0	2.6142	441.93	0.000019	0.002412	2
-110.0	3.0252	457.96	0.000025	-0.006820	2
-100.0	3.5418	475.75	0.000032	-0.017197	2
-90.0	4.1946	495.42	0.000041	-0.028815	2
-80.0	5.0235	517.19	0.000052	-0.041751	2
-70.0	6.0720	540.81	0.000066	-0.056021	2
-60.0	7.3831	565.81	0.000084	-0.071517	2
-50.0	8.9851	591.54	0.000105	-0.087921	2
-40.0	10.8563	616.84	0.000130	-0.104577	2
-30.0	12.8713	639.82	0.000156	-0.120372	2

>>>>> START OF COMBUSTION AND EXPANSION PROCESSES

CA (DEG)	P (ATM)	TB (K)	TU (K)	XBURND (-)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
-30.0	12.8716	2373.61	639.27	0.00030	0.000156	-0.120372	2
-29.0	13.0745	2353.29	641.38	0.00037	0.000168	-0.121846	2
-28.0	13.2872	2349.54	643.57	0.00067	0.000176	-0.123295	2
-27.0	13.5147	2353.19	645.89	0.00131	0.000188	-0.124719	2
-26.0	13.7606	2357.54	648.37	0.00238	0.000203	-0.126116	2
-25.0	14.0283	2361.65	651.04	0.00393	0.000221	-0.127488	2
-24.0	14.3206	2365.70	653.93	0.00602	0.000241	-0.128833	2
-23.0	14.6400	2369.88	657.04	0.00870	0.000264	-0.130154	2
-22.0	14.9886	2374.30	660.38	0.01201	0.000289	-0.131448	2
-21.0	15.3683	2378.99	663.97	0.01599	0.000317	-0.132717	2
-20.0	15.7807	2383.97	667.79	0.02067	0.000347	-0.133959	2
-19.0	16.2269	2389.24	671.83	0.02607	0.000379	-0.135174	2
-18.0	16.7077	2394.78	676.11	0.03224	0.000415	-0.136362	2
-17.0	17.2237	2400.56	680.58	0.03917	0.000452	-0.137520	2
-16.0	17.7750	2406.55	685.25	0.04690	0.000492	-0.138647	2
-15.0	18.3612	2412.72	690.08	0.05542	0.000536	-0.139741	2
-14.0	18.9819	2419.03	695.06	0.06476	0.000581	-0.140800	2
-13.0	19.6361	2425.43	700.17	0.07490	0.000630	-0.141820	2
-12.0	20.3221	2431.88	705.37	0.08586	0.000682	-0.142797	2
-11.0	21.0386	2438.35	710.65	0.09762	0.000736	-0.143728	2
-10.0	21.7831	2444.77	715.97	0.11018	0.000794	-0.144609	2
-9.0	22.5533	2451.13	721.31	0.12352	0.000854	-0.145435	2
-8.0	23.3467	2457.37	726.64	0.13762	0.000917	-0.146200	2
-7.0	24.1600	2463.47	731.94	0.15247	0.000983	-0.146898	2
-6.0	24.9900	2469.38	737.19	0.16804	0.001052	-0.147525	2
-5.0	25.8336	2475.10	742.36	0.18430	0.001123	-0.148074	2
-4.0	26.6866	2480.57	747.43	0.20121	0.001196	-0.148538	2
-3.0	27.5456	2485.79	752.38	0.21875	0.001272	-0.148910	2
-2.0	28.4066	2490.73	757.20	0.23687	0.001349	-0.149185	2
-1.0	29.2656	2495.37	761.86	0.25553	0.001429	-0.149354	2
0.0	30.1188	2499.69	766.35	0.27470	0.001509	-0.149412	2
1.0	30.9622	2503.69	770.66	0.29431	0.001591	-0.149352	2
2.0	31.7919	2507.35	774.77	0.31432	0.001674	-0.149167	2
3.0	32.6040	2510.66	778.68	0.33469	0.001757	-0.148850	2
4.0	33.3951	2513.62	782.38	0.35536	0.001840	-0.148396	2
5.0	34.1616	2516.22	785.85	0.37627	0.001923	-0.147799	2
6.0	34.9000	2518.46	789.09	0.39738	0.002006	-0.147053	2
7.0	35.6074	2520.33	792.10	0.41862	0.002088	-0.146153	2
8.0	36.2808	2521.84	794.87	0.43995	0.002168	-0.145095	2
9.0	36.9177	2522.98	797.40	0.46131	0.002247	-0.143874	2

10.0	37.5156	2523.75	799.68	0.48264	0.002324	-0.142488	2
11.0	38.0726	2524.16	801.72	0.50390	0.002398	-0.140932	2
12.0	38.5867	2524.20	803.51	0.52502	0.002470	-0.139205	2
13.0	39.0566	2523.89	805.06	0.54597	0.002540	-0.137304	2
14.0	39.4810	2523.23	806.36	0.56668	0.002605	-0.135230	2
15.0	39.8590	2522.22	807.43	0.58712	0.002668	-0.132980	2
16.0	40.1900	2520.86	808.25	0.60725	0.002727	-0.130555	2
17.0	40.4736	2519.16	808.84	0.62701	0.002782	-0.127956	2
18.0	40.7097	2517.13	809.19	0.64637	0.002833	-0.125184	2
19.0	40.8984	2514.78	809.31	0.66529	0.002880	-0.122240	2
20.0	41.0403	2512.10	809.21	0.68374	0.002922	-0.119127	2
21.0	41.1359	2509.10	808.89	0.70169	0.002960	-0.115849	2
22.0	41.1860	2505.80	808.36	0.71911	0.002994	-0.112407	2
23.0	41.1917	2502.20	807.61	0.73599	0.003023	-0.108807	2
24.0	41.1542	2498.30	806.66	0.75229	0.003047	-0.105052	2
25.0	41.0749	2494.11	805.52	0.76800	0.003067	-0.101147	2
26.0	40.9552	2489.64	804.18	0.78311	0.003082	-0.097097	2
27.0	40.7969	2484.90	802.65	0.79761	0.003093	-0.092908	2
28.0	40.6015	2479.89	800.94	0.81149	0.003099	-0.088584	2
29.0	40.3710	2474.62	799.06	0.82474	0.003102	-0.084133	2
30.0	40.1073	2469.09	797.01	0.83737	0.003100	-0.079559	2
31.0	39.8122	2463.33	794.80	0.84938	0.003094	-0.074870	2
32.0	39.4877	2457.32	792.43	0.86077	0.003084	-0.070072	2
33.0	39.1359	2451.09	789.92	0.87154	0.003070	-0.065170	2
34.0	38.7588	2444.64	787.26	0.88172	0.003053	-0.060172	2
35.0	38.3584	2437.97	784.46	0.89130	0.003033	-0.055084	2
36.0	37.9366	2431.09	781.54	0.90030	0.003009	-0.049913	2
37.0	37.4975	2424.12	779.24	0.90875	0.002983	-0.044665	2
38.0	37.0389	2416.86	776.07	0.91664	0.002954	-0.039346	2
39.0	36.5647	2409.41	772.75	0.92401	0.002922	-0.033963	2
40.0	36.0766	2401.78	769.27	0.93088	0.002888	-0.028523	2
40.3	35.9453	2399.73	768.32	0.93261	0.002879	-0.027073	8
41.0	35.5764	2393.98	765.60	0.93725	0.002852	-0.023030	2
42.0	35.0655	2386.00	761.69	0.94316	0.002814	-0.017492	2
43.0	34.5469	2377.87	757.49	0.94861	0.002775	-0.011914	2
44.0	34.0194	2369.55	752.92	0.95365	0.002734	-0.006302	2
45.0	33.4847	2361.04	747.91	0.95828	0.002691	-0.000660	2
46.0	32.9468	2352.42	742.50	0.96254	0.002647	0.005005	2
47.0	32.4051	2343.65	736.65	0.96643	0.002602	0.010689	2
48.0	31.8626	2334.78	730.38	0.96999	0.002557	0.016385	2
49.0	31.3188	2325.78	723.60	0.97323	0.002511	0.022091	2
50.0	30.7761	2316.69	716.34	0.97618	0.002465	0.027801	2
51.0	30.2346	2307.50	708.54	0.97885	0.002418	0.033512	2
52.0	29.6961	2298.24	700.19	0.98127	0.002372	0.039219	2
53.0	29.1607	2288.90	691.22	0.98346	0.002326	0.044919	2
54.0	28.6296	2279.51	681.65	0.98542	0.002279	0.050608	2
55.0	28.1033	2270.06	671.41	0.98719	0.002234	0.056284	2
56.0	27.5825	2260.56	660.48	0.98877	0.002188	0.061942	2
57.0	27.0678	2251.04	648.84	0.99018	0.002143	0.067581	2
58.0	26.5597	2241.48	636.46	0.99145	0.002099	0.073197	2
59.0	26.0585	2231.91	623.32	0.99257	0.002055	0.078787	2

60.0	25.5648	2222.33	609.40	0.99356	0.002012	0.084351	2
61.0	25.0786	2212.74	594.71	0.99444	0.001970	0.089885	2
62.0	24.5988	2203.11	579.14	0.99515	0.001928	0.095388	2
63.0	24.1236	2193.41	563.00	0.99561	0.001886	0.100857	2
64.0	23.6575	2183.75	547.00	0.99601	0.001846	0.106290	2
65.0	23.2005	2174.12	531.27	0.99636	0.001806	0.111686	2
66.0	22.7528	2164.53	515.96	0.99667	0.001767	0.117043	2
67.0	22.3142	2154.98	501.19	0.99693	0.001729	0.122360	2
68.0	21.8847	2145.48	487.19	0.99717	0.001693	0.127637	2
69.0	21.4642	2136.04	474.02	0.99737	0.001657	0.132872	2
70.0	21.0528	2126.64	461.72	0.99754	0.001622	0.138065	2
71.0	20.6503	2117.31	450.36	0.99769	0.001589	0.143215	2
72.0	20.2566	2108.04	439.97	0.99782	0.001556	0.148320	2
73.0	19.8717	2098.82	430.53	0.99793	0.001524	0.153382	2
74.0	19.4953	2089.67	422.03	0.99803	0.001493	0.158398	2
75.0	19.1268	2080.57	414.43	0.99808	0.001463	0.163369	2
76.0	18.7668	2071.54	407.74	0.99813	0.001434	0.168294	2
77.0	18.4151	2062.59	401.85	0.99817	0.001406	0.173173	2
78.0	18.0715	2053.70	396.67	0.99820	0.001378	0.178005	2
79.0	17.7359	2044.88	392.12	0.99823	0.001352	0.182791	2
80.0	17.4081	2036.14	388.11	0.99825	0.001326	0.187530	2
81.0	17.0879	2027.47	384.58	0.99827	0.001301	0.192221	2
82.0	16.7752	2018.88	381.46	0.99829	0.001277	0.196867	2
83.0	16.4699	2010.36	378.70	0.99830	0.001253	0.201465	2
84.0	16.1718	2001.92	376.24	0.99832	0.001230	0.206016	2
85.0	15.8806	1993.55	374.05	0.99833	0.001208	0.210520	2
86.0	15.5963	1985.26	372.08	0.99833	0.001187	0.214977	2
87.0	15.3187	1977.04	370.30	0.99834	0.001166	0.219387	2
88.0	15.0476	1968.90	368.68	0.99835	0.001145	0.223751	2
89.0	14.7829	1960.83	367.21	0.99835	0.001126	0.228068	2
90.0	14.5244	1952.84	365.86	0.99835	0.001107	0.232339	2
91.0	14.2720	1944.93	364.61	0.99836	0.001088	0.236564	2
92.0	14.0256	1937.08	363.45	0.99836	0.001070	0.240743	2
93.0	13.7849	1929.32	362.36	0.99836	0.001053	0.244876	2
94.0	13.5498	1921.63	361.34	0.99836	0.001036	0.248963	2
95.0	13.3203	1914.01	360.37	0.99836	0.001019	0.253006	2
96.0	13.0962	1906.46	359.44	0.99837	0.001003	0.257003	2
97.0	12.8772	1898.99	358.55	0.99837	0.000987	0.260956	2
98.0	12.6634	1891.60	357.69	0.99837	0.000972	0.264864	2
99.0	12.4546	1884.27	356.86	0.99837	0.000957	0.268728	2
100.0	12.2506	1877.02	356.05	0.99837	0.000943	0.272549	2
101.0	12.0514	1869.85	355.26	0.99837	0.000929	0.276326	2
102.0	11.8568	1862.74	354.48	0.99837	0.000916	0.280060	2
103.0	11.6666	1855.70	353.71	0.99837	0.000902	0.283751	2
104.0	11.4809	1848.74	352.94	0.99837	0.000889	0.287399	2
105.0	11.2994	1841.84	352.18	0.99837	0.000877	0.291006	2
106.0	11.1221	1835.02	351.42	0.99837	0.000865	0.294571	2
107.0	10.9489	1828.27	350.66	0.99837	0.000853	0.298094	2
108.0	10.7796	1821.58	349.90	0.99837	0.000841	0.301577	2
109.0	10.6142	1814.96	349.13	0.99837	0.000830	0.305019	2
110.0	10.4525	1808.41	348.35	0.99837	0.000819	0.308420	2
111.0	10.2945	1801.93	347.57	0.99837	0.000808	0.311782	2
112.0	10.1401	1795.52	346.77	0.99837	0.000798	0.315104	2
113.0	9.9891	1789.17	345.97	0.99837	0.000787	0.318387	2
114.0	9.8416	1782.88	345.15	0.99837	0.000777	0.321631	2

115.0	9.6973	1776.67	344.31	0.99837	0.000768	0.324837	2
116.0	9.5563	1770.51	343.46	0.99837	0.000758	0.328004	2
117.0	9.4184	1764.42	342.60	0.99837	0.000749	0.331134	2
118.0	9.2836	1758.40	341.71	0.99837	0.000740	0.334227	2
119.0	9.1518	1752.43	340.81	0.99837	0.000731	0.337282	2
120.0	9.0228	1746.53	339.88	0.99837	0.000722	0.340301	2
121.0	8.8967	1740.70	338.94	0.99837	0.000714	0.343284	2
122.0	8.7734	1734.92	337.97	0.99837	0.000706	0.346230	2
122.5	8.7123	1732.03	337.46	0.99837	0.000702	0.347701	8
123.0	8.6528	1729.20	336.96	0.99837	0.000698	0.349142	2
124.0	8.5348	1723.55	335.98	0.99837	0.000690	0.352018	2
125.0	8.4194	1717.95	335.01	0.99837	0.000682	0.354859	2
126.0	8.3066	1712.42	334.07	0.99837	0.000675	0.357666	2
127.0	8.1961	1706.94	333.14	0.99837	0.000667	0.360439	2
128.0	8.0880	1701.52	332.23	0.99837	0.000660	0.363178	2
129.0	7.9823	1696.15	331.33	0.99837	0.000653	0.365883	2
130.0	7.8788	1690.85	330.44	0.99837	0.000646	0.368556	2
131.0	7.7775	1685.60	329.57	0.99837	0.000640	0.371196	2
132.0	7.6784	1680.41	328.72	0.99837	0.000633	0.373803	2
133.0	7.5814	1675.27	327.88	0.99837	0.000627	0.376379	2
134.0	7.4865	1670.19	327.05	0.99837	0.000620	0.378923	2
135.0	7.3935	1665.16	326.24	0.99837	0.000614	0.381435	2
136.0	7.3025	1660.18	325.44	0.99837	0.000608	0.383916	2
137.0	7.2134	1655.26	324.66	0.99837	0.000602	0.386367	2
138.0	7.1262	1650.39	323.89	0.99837	0.000597	0.388787	2
139.0	7.0408	1645.58	323.13	0.99837	0.000591	0.391177	2
140.0	6.9572	1640.81	322.39	0.99837	0.000585	0.393537	2
141.0	6.8753	1636.10	321.65	0.99837	0.000580	0.395868	2
142.0	6.7951	1631.44	320.94	0.99837	0.000575	0.398170	2
143.0	6.7165	1626.83	320.23	0.99837	0.000569	0.400443	2
144.0	6.6396	1622.26	319.53	0.99837	0.000564	0.402687	2
145.0	6.5642	1617.75	318.85	0.99837	0.000559	0.404902	2
146.0	6.4904	1613.29	318.18	0.99837	0.000555	0.407090	2
147.0	6.4181	1608.87	317.53	0.99837	0.000550	0.409250	2
148.0	6.3472	1604.51	316.88	0.99837	0.000545	0.411383	2
149.0	6.2778	1600.19	316.25	0.99837	0.000540	0.413488	2
150.0	6.2098	1595.92	315.62	0.99837	0.000536	0.415567	2
151.0	6.1432	1591.69	315.01	0.99837	0.000531	0.417619	2
152.0	6.0779	1587.52	314.41	0.99837	0.000527	0.419644	2
153.0	6.0139	1583.38	313.82	0.99837	0.000523	0.421644	2
154.0	5.9512	1579.30	313.24	0.99837	0.000519	0.423618	2
155.0	5.8898	1575.25	312.68	0.99837	0.000515	0.425566	2
156.0	5.8296	1571.26	312.12	0.99837	0.000511	0.427488	2
157.0	5.7705	1567.30	311.57	0.99837	0.000507	0.429386	2
158.0	5.7127	1563.39	311.04	0.99837	0.000503	0.431259	2
159.0	5.6560	1559.53	310.51	0.99837	0.000499	0.433107	2
160.0	5.6004	1555.71	310.00	0.99837	0.000495	0.434931	2
161.0	5.5459	1551.93	309.49	0.99837	0.000492	0.436731	2
162.0	5.4925	1548.19	309.00	0.99837	0.000488	0.438507	2
163.0	5.4402	1544.49	308.51	0.99837	0.000485	0.440259	2
164.0	5.3888	1540.84	308.04	0.99837	0.000481	0.441988	2

165.0	5.3385	1537.22	307.57	0.99837	0.000478	0.443694	2
166.0	5.2892	1533.65	307.12	0.99837	0.000474	0.445376	2
167.0	5.2408	1530.12	306.67	0.99837	0.000471	0.447036	2
168.0	5.1934	1526.62	306.23	0.99837	0.000468	0.448673	2
169.0	5.1468	1523.17	305.80	0.99837	0.000465	0.450288	2
170.0	5.1012	1519.76	305.38	0.99837	0.000462	0.451881	2
171.0	5.0565	1516.38	304.97	0.99837	0.000459	0.453452	2
172.0	5.0127	1513.04	304.57	0.99837	0.000456	0.455001	2
173.0	4.9697	1509.75	304.18	0.99837	0.000453	0.456529	2
174.0	4.9275	1506.49	303.79	0.99837	0.000450	0.458035	2
175.0	4.8862	1503.26	303.42	0.99837	0.000447	0.459520	2
176.0	4.8456	1500.08	303.05	0.99837	0.000444	0.460984	2
177.0	4.8059	1496.93	302.69	0.99837	0.000442	0.462427	2
178.0	4.7669	1493.82	302.34	0.99837	0.000439	0.463850	2
179.0	4.7286	1490.74	302.00	0.99837	0.000436	0.465252	2
180.0	4.6911	1487.70	301.67	0.99837	0.000434	0.466634	2
181.0	4.6543	1484.69	301.34	0.99837	0.000431	0.467996	2
182.0	4.6183	1481.73	301.02	0.99837	0.000429	0.469338	2
183.0	4.5829	1478.79	300.71	0.99837	0.000426	0.470661	2
184.0	4.5482	1475.89	300.41	0.99837	0.000424	0.471963	2
185.0	4.5142	1473.03	300.12	0.99837	0.000421	0.473247	2
186.0	4.4808	1470.20	299.83	0.99837	0.000419	0.474511	2
187.0	4.4481	1467.40	299.55	0.99837	0.000417	0.475756	2
188.0	4.4160	1464.64	299.28	0.99837	0.000415	0.476982	2
189.0	4.3846	1461.91	299.02	0.99837	0.000412	0.478189	2
190.0	4.3537	1459.21	298.76	0.99837	0.000410	0.479378	2
191.0	4.3235	1456.55	298.52	0.99837	0.000408	0.480548	2
192.0	4.2938	1453.92	298.27	0.99837	0.000406	0.481699	2
193.0	4.2648	1451.32	298.04	0.99837	0.000404	0.482833	2
194.0	4.2363	1448.75	297.81	0.99837	0.000402	0.483948	2
195.0	4.2083	1446.22	297.59	0.99837	0.000400	0.485046	2
196.0	4.1809	1443.71	297.38	0.99837	0.000398	0.486126	2
197.0	4.1540	1441.24	297.18	0.99837	0.000396	0.487188	2
198.0	4.1277	1438.80	296.98	0.99837	0.000394	0.488233	2
199.0	4.1019	1436.39	296.79	0.99837	0.000392	0.489260	2

>>>>> START OF EXHAUST PROCESS

CA (DEG)	P (ATM)	TEMP (K)	MEX (G)	VEV (M/SEC)	Q DOT (KJ/DEG)	WORK (KJ)	IFG
199.0	4.1019	1436.39	-0.01511	0.0	0.000392	0.489260	2
200.0	4.0758	1434.07	-0.01499	27739.8	0.000306	0.490270	2
210.0	3.7284	1403.59	-0.00078	27452.6	0.000287	0.499344	2
220.0	3.2678	1361.34	0.03378	27049.3	0.000257	0.506420	2
230.0	2.7555	1309.09	0.08299	26541.7	0.000221	0.511520	2
240.0	2.2441	1248.89	0.14070	25944.4	0.000183	0.514875	2
250.0	1.8197	1189.71	0.19421	25337.1	0.000151	0.516845	2
260.0	1.5002	1136.80	0.23840	23776.6	0.000125	0.517824	2
270.0	1.2704	1092.13	0.27298	20207.8	0.000106	0.518102	2
280.0	1.1213	1058.25	0.29744	14498.3	0.000093	0.517873	2
290.0	1.0446	1037.36	0.31182	7638.1	0.000086	0.517244	2
300.0	1.0246	1028.32	0.31819	3371.7	0.000083	0.516247	2
310.0	1.0257	1023.65	0.32245	3725.8	0.000082	0.514877	2
320.0	1.0290	1019.10	0.32774	4653.2	0.000081	0.513143	2
330.0	1.0331	1014.29	0.33428	5574.3	0.000079	0.511065	2
340.0	1.0379	1009.15	0.34203	6471.9	0.000078	0.508667	2
350.0	1.0438	1003.33	0.35106	7403.6	0.000076	0.505976	2
360.0	1.0514	996.33	0.36147	8397.8	0.000074	0.503022	2
370.0	1.0601	988.00	0.37337	9369.2	0.000072	0.499838	2
380.0	1.0686	978.75	0.38672	10183.1	0.000069	0.496462	2
390.0	1.0754	969.23	0.40124	10737.5	0.000067	0.492941	2
400.0	1.0791	960.00	0.41655	10998.6	0.000064	0.489329	2
410.0	1.0795	951.35	0.43221	10986.1	0.000061	0.485685	2
420.0	1.0771	943.25	0.44780	10742.4	0.000058	0.482069	2
430.0	1.0727	935.61	0.46299	10320.7	0.000055	0.478539	2
440.0	1.0669	928.27	0.47751	9757.0	0.000052	0.475145	2
450.0	1.0607	921.18	0.49117	9108.3	0.000050	0.471937	2
460.0	1.0546	914.27	0.50381	8418.5	0.000047	0.468955	2
470.0	1.0477	907.18	0.51535	7557.5	0.000045	0.466239	2
480.0	1.0408	899.96	0.52565	6569.2	0.000043	0.463820	2
490.0	1.0343	892.56	0.53461	5459.6	0.000041	0.461727	2
500.0	1.0305	885.33	0.54212	4679.7	0.000040	0.459984	2
510.0	1.0262	877.62	0.54820	3602.5	0.000038	0.458610	2
520.0	1.0231	869.84	0.55273	2549.5	0.000037	0.457618	2
530.0	1.0210	862.14	0.55566	1403.8	0.000037	0.457019	2
540.0	1.0203	854.92	0.55695	821.9	0.000036	0.456818	2
550.0	1.0190	847.97	0.55660	-1428.8	0.000036	0.457019	2

```
>--------------+--------------<
>                             <
>     CALCULATION RESULTS     <
>                             <
>--------------+--------------<
```

--> VOLUMETRIC EFFICIENCY; (%) BASED ON: INTAKE / ATM	---->	75.9	74.4
--> PUMPING MEAN EFFECTIVE PRESSURE; (KPA) : PEMP	---->	-10.	
--> GROSS INDICATED MEAN EFFECTIVE PRESSURE; (KPA) : IMEP	---->	807.	
--> GROSS INDICATED SPECIFIC FUEL CONSUMPTION; (G/IKW-HR) : ISFC	---->	270.	
--> GROSS INDICATED THERMAL EFFICIENCY; (%)	---->	30.0	
--> NET INDICATED THERMAL EFFICIENCY; (%)	---->	29.7	
--> (HEAT TRANSFER PER CYCLE)/ (MASS OF FUEL TIMES LHV); (%)	---->	21.2	
--> IGNITION DELAY (0 - 10%) (CRANK ANGLE) / (MS)	---->	19.19	1.07
--> BURN DURATION (10 - 90%) (CRANK ANGLE) / (MS)	---->	46.78	2.60
--> MEAN EXHAUST TEMPERATURE; (K)	---->	1124.4	

```
MASS IN CYLINDER AT TIVO =  0.02858 G

MASS IN CYLINDER AT TIVC =  0.66149 G

MASS OF FUEL INDUCTED    =  0.03467 G

RESIDUAL FRACTION        =  0.09032
```

```
HEATI   =    0.001518 KJ  (TIPO   -   -270)
WORKI   =    0.055626 KJ

HEATC   =    0.006440 KJ  (-270   - TSPARK)
WORKC   =   -0.175998 KJ

HEATCE  =    0.308159 KJ  (TIPC   -    +270
WORKCE  =    0.462476 KJ

HEATE   =    0.017150 KJ  (+270   -   TIPO)
WORKE   =   -0.061083 KJ
```

```
TOTAL ENTHALPY IN / CYCLE    =  0.09030 KJ

TOTAL ENTHALPY OUT / CYCLE   = -0.88656 KJ

TOTAL HEAT LOSS / CYCLE      =  0.32683 KJ

TOTAL WORK OUTPUT / CYCLE    =  0.45702 KJ

HEAT LOSS TO CREVICE/CYCLE   =  0.11964 KJ

"LOST" FUEL ENERGY           =  0.09587 KJ

NET ENERGY GAIN / CYCLE      =  0.02250 KJ

(ENERGY GAIN)/(ENTHALPY IN) = 24.91203 %

(ENERGY GAIN)/(MFUEL*LHV)    =  1.46151 %
```

LEAKAGE AND CREVICE VOLUME DATA

CA (DEG)	CHAMBER MASS (G)	LEAD CREVICE MASS (G)	LAG CREVICE MASS (G)	LEAD LEAKAGE MASS (G)	LAG LEAKAGE MASS (G)	LEAD CREVICE COMPOSITION ( )	LAG CREVICE COMPOSITION ( )
-520.0	0.0314	0.000000	0.000000	-0.000177	-0.000496	0.000000	0.000000
-510.0	0.0353	0.000000	0.000000	-0.000379	-0.000956	0.000000	0.000000
-505.8	0.0376	0.000000	0.000000	-0.000471	-0.001136	0.000000	0.000000
-500.0	0.0412	0.000000	0.000000	-0.000605	-0.001368	0.000000	0.000000
-492.9	0.0461	0.000000	0.000000	-0.000782	-0.001626	0.000000	0.000000
-490.0	0.0495	0.000000	0.000000	-0.000859	-0.001722	0.000000	0.000000
-480.0	0.0708	0.000000	0.000000	-0.001150	-0.002016	0.000000	0.000000
-470.0	0.0973	0.000000	0.000000	-0.001485	-0.002253	0.000000	0.000000
-460.0	0.1274	0.000000	0.000000	-0.001875	-0.002447	0.000000	0.000000
-450.0	0.1604	0.000000	0.000000	-0.002335	-0.002599	0.000000	0.000000
-440.0	0.1957	0.000000	0.000000	-0.002883	-0.002715	0.000000	0.000000
-430.0	0.2330	0.000000	0.000000	-0.003542	-0.002798	0.000000	0.000000
-420.0	0.2718	0.000000	0.000000	-0.004342	-0.002864	0.000000	0.000000
-410.0	0.3116	0.000000	0.000000	-0.005315	-0.002925	0.000000	0.000000
-400.0	0.3508	0.000000	0.000000	-0.006496	-0.002986	0.000000	0.000000
-390.0	0.3897	0.000000	0.000000	-0.007910	-0.003049	0.000000	0.000000
-380.0	0.4280	0.000000	0.000000	-0.009596	-0.003113	0.000000	0.000000
-370.0	0.4652	0.000000	0.000000	-0.011802	-0.003179	0.000000	0.000000
-360.0	0.5007	0.000000	0.000000	-0.014881	-0.003247	0.000000	0.000000
-350.0	0.5335	0.000000	0.000000	-0.018924	-0.003317	0.000000	0.000000
-340.0	0.5631	0.000000	0.000000	-0.023625	-0.003391	0.000000	0.000000
-330.0	0.5896	0.000000	0.000000	-0.028464	-0.003467	0.000000	0.000000
-320.0	0.6126	0.000000	0.000000	-0.032988	-0.003544	0.000000	0.000000
-310.0	0.6321	0.000000	0.000000	-0.036943	-0.003623	0.000000	0.000000
-300.0	0.6479	0.000000	0.000000	-0.040268	-0.003700	0.000000	0.000000
-290.0	0.6593	0.000000	0.000000	-0.043014	-0.003776	0.000000	0.000000
-280.0	0.6669	0.000000	0.000000	-0.045280	-0.003849	0.000000	0.000000
-277.3	0.6679	0.000000	0.000000	-0.045823	-0.003868	0.000000	0.000000
-270.0	0.6688	0.000000	0.000000	-0.047160	-0.003919	0.000000	0.000000
-260.0	0.6677	0.000000	0.000000	-0.048738	-0.003985	0.000000	0.000000
-250.0	0.6648	0.000000	0.000000	-0.050076	-0.004046	0.000000	0.000000
-240.0	0.6611	0.000000	0.000000	-0.051224	-0.004100	0.000000	0.000000
-230.0	0.6595	0.000000	0.000000	-0.052220	-0.004136	0.000000	0.000000
-220.0	0.6597	0.000000	0.000892	-0.053095	-0.004112	0.000000	0.234322
-210.0	0.6602	0.000000	0.000939	-0.053872	-0.004049	0.000000	0.308123
-200.0	0.6607	0.000000	0.000998	-0.054568	-0.003962	0.000000	0.390065
-190.0	0.6611	0.000000	0.001069	-0.055199	-0.003852	0.000000	0.471789
-180.0	0.6615	0.000000	0.001156	-0.055776	-0.003720	0.000000	0.548449

-170.0	0.6618	0.000000	0.001261	-0.056309	-0.003566	0.000000	0.617016
-160.0	0.6620	0.000000	0.001389	-0.056805	-0.003389	0.000000	0.676261
-150.0	0.6621	0.000000	0.001544	-0.057266	-0.003188	0.000000	0.725959
-140.0	0.6621	0.000000	0.001734	-0.057677	-0.002963	0.000000	0.766591
-130.0	0.6619	0.000000	0.001968	-0.058001	-0.002709	0.000000	0.799286
-120.0	0.6612	0.002215	0.002256	-0.058032	-0.002419	0.503044	0.825299
-110.0	0.6599	0.002577	0.002614	-0.057732	-0.002086	0.599186	0.845764
-100.0	0.6582	0.003031	0.003063	-0.057346	-0.001697	0.678318	0.861720
-90.0	0.6562	0.003602	0.003630	-0.056891	-0.001239	0.739108	0.873995
-80.0	0.6537	0.004326	0.004350	-0.056347	-0.000694	0.785179	0.883308
-70.0	0.6506	0.005239	0.005260	-0.055691	-0.000037	0.819510	0.890255
-60.0	0.6467	0.006379	0.006398	-0.054896	0.000759	0.844633	0.895342
-50.0	0.6420	0.007771	0.007787	-0.053928	0.001728	0.862677	0.898998
-40.0	0.6364	0.009396	0.009411	-0.052752	0.002905	0.875363	0.901569
-30.0	0.6301	0.011146	0.011158	-0.051344	0.004314	0.884063	0.903332
-29.0	0.6294	0.011322	0.011335	-0.051190	0.004468	0.884754	0.904023
-28.0	0.6287	0.011507	0.011520	-0.051033	0.004624	0.885431	0.904700
-27.0	0.6280	0.011704	0.011717	-0.050874	0.004784	0.886094	0.905363
-26.0	0.6273	0.011918	0.011931	-0.050712	0.004946	0.886734	0.906003
-25.0	0.6265	0.012150	0.012164	-0.050547	0.005111	0.887340	0.906608
-24.0	0.6256	0.012404	0.012417	-0.050379	0.005279	0.887893	0.907162
-23.0	0.6248	0.012681	0.012695	-0.050207	0.005451	0.888374	0.907642
-22.0	0.6238	0.012983	0.012997	-0.050031	0.005627	0.888757	0.908026
-21.0	0.6228	0.013312	0.013327	-0.049851	0.005808	0.889015	0.908283
-20.0	0.6217	0.013669	0.013685	-0.049666	0.005993	0.889118	0.908386
-19.0	0.6205	0.014056	0.014071	-0.049476	0.006183	0.889033	0.908301
-18.0	0.6193	0.014472	0.014488	-0.049280	0.006378	0.888728	0.907996
-17.0	0.6180	0.014918	0.014935	-0.049079	0.006580	0.888171	0.907439
-16.0	0.6166	0.015395	0.015412	-0.048871	0.006788	0.887331	0.906599
-15.0	0.6152	0.015902	0.015920	-0.048656	0.007002	0.886178	0.905447
-14.0	0.6137	0.016438	0.016456	-0.048434	0.007224	0.884687	0.903956
-13.0	0.6121	0.017003	0.017022	-0.048205	0.007454	0.882834	0.902104
-12.0	0.6104	0.017595	0.017614	-0.047968	0.007691	0.880601	0.899871
-11.0	0.6087	0.018212	0.018232	-0.047722	0.007937	0.877972	0.897242
-10.0	0.6069	0.018853	0.018874	-0.047468	0.008191	0.874938	0.894209
-9.0	0.6050	0.019516	0.019538	-0.047205	0.008454	0.871493	0.890764
-8.0	0.6031	0.020198	0.020221	-0.046933	0.008727	0.867633	0.886906
-7.0	0.6011	0.020897	0.020920	-0.046651	0.009009	0.863366	0.882639
-6.0	0.5991	0.021609	0.021633	-0.046359	0.009301	0.858696	0.877969
-5.0	0.5971	0.022332	0.022356	-0.046057	0.009602	0.853632	0.872907
-4.0	0.5950	0.023062	0.023087	-0.045746	0.009914	0.848192	0.867467
-3.0	0.5928	0.023796	0.023823	-0.045424	0.010236	0.842390	0.861667
-2.0	0.5907	0.024531	0.024559	-0.045092	0.010568	0.836249	0.855528
-1.0	0.5885	0.025264	0.025292	-0.044750	0.010910	0.829791	0.849071
0.0	0.5863	0.025990	0.026019	-0.044398	0.011263	0.823040	0.842321
1.0	0.5842	0.026707	0.026737	-0.044036	0.011625	0.816023	0.835306
2.0	0.5820	0.027412	0.027442	-0.043664	0.011997	0.808768	0.828052
3.0	0.5798	0.028100	0.028131	-0.043282	0.012379	0.801304	0.820590
4.0	0.5777	0.028769	0.028801	-0.042891	0.012770	0.793661	0.812949
5.0	0.5755	0.029416	0.029449	-0.042491	0.013171	0.785870	0.805160
6.0	0.5735	0.030039	0.030072	-0.042081	0.013580	0.777963	0.797255
7.0	0.5714	0.030634	0.030667	-0.041664	0.013998	0.769970	0.789264
8.0	0.5694	0.031199	0.031233	-0.041238	0.014424	0.761923	0.781219
9.0	0.5674	0.031732	0.031767	-0.040805	0.014858	0.753852	0.773151

10.0	0.5655	0.032231	0.032266	-0.040364	0.015298	0.745788	0.765089
11.0	0.5637	0.032694	0.032730	-0.039917	0.015746	0.737760	0.757064
12.0	0.5619	0.033121	0.033157	-0.039463	0.016200	0.729798	0.749104
13.0	0.5601	0.033509	0.033546	-0.039004	0.016660	0.721928	0.741237
14.0	0.5585	0.033858	0.033895	-0.038539	0.017124	0.714178	0.733489
15.0	0.5569	0.034168	0.034205	-0.038070	0.017594	0.706572	0.725887
16.0	0.5554	0.034437	0.034475	-0.037596	0.018067	0.699136	0.718454
17.0	0.5539	0.034665	0.034704	-0.037119	0.018545	0.691892	0.711212
18.0	0.5526	0.034854	0.034892	-0.036639	0.019025	0.684860	0.704184
19.0	0.5513	0.035002	0.035040	-0.036157	0.019507	0.678060	0.697387
20.0	0.5501	0.035110	0.035149	-0.035673	0.019992	0.671511	0.690841
21.0	0.5489	0.035179	0.035218	-0.035187	0.020478	0.665228	0.684562
22.0	0.5479	0.035210	0.035249	-0.034701	0.020964	0.659226	0.678564
23.0	0.5469	0.035204	0.035242	-0.034214	0.021451	0.653518	0.672860
24.0	0.5460	0.035161	0.035200	-0.033728	0.021938	0.648115	0.667461
25.0	0.5452	0.035083	0.035122	-0.033242	0.022423	0.643028	0.662377
26.0	0.5444	0.034971	0.035010	-0.032758	0.022908	0.638262	0.657615
27.0	0.5437	0.034827	0.034866	-0.032275	0.023391	0.633826	0.653183
28.0	0.5431	0.034653	0.034691	-0.031794	0.023872	0.629723	0.649085
29.0	0.5425	0.034449	0.034486	-0.031316	0.024350	0.625958	0.645323
30.0	0.5420	0.034217	0.034254	-0.030841	0.024825	0.622530	0.641900
31.0	0.5416	0.033959	0.033996	-0.030370	0.025297	0.619442	0.638816
32.0	0.5412	0.033677	0.033714	-0.029902	0.025765	0.616691	0.636069
33.0	0.5409	0.033373	0.033409	-0.029438	0.026230	0.614275	0.633658
34.0	0.5406	0.033047	0.033083	-0.028978	0.026690	0.612192	0.631579
35.0	0.5404	0.032702	0.032738	-0.028523	0.027145	0.610435	0.629827
36.0	0.5402	0.032340	0.032376	-0.028073	0.027595	0.609001	0.628397
37.0	0.5400	0.031964	0.031999	-0.027628	0.028040	0.607881	0.627282
38.0	0.5399	0.031571	0.031606	-0.027188	0.028480	0.607060	0.626465
39.0	0.5399	0.031166	0.031201	-0.026754	0.028915	0.606509	0.625919
40.0	0.5399	0.030750	0.030784	-0.026325	0.029344	0.606192	0.625606
40.3	0.5399	0.030638	0.030671	-0.026212	0.029456	0.606141	0.625556
41.0	0.5399	0.030323	0.030356	-0.025902	0.029767	0.606057	0.625474
42.0	0.5399	0.029888	0.029920	-0.025485	0.030184	0.606032	0.625451
43.0	0.5400	0.029446	0.029478	-0.025075	0.030594	0.606046	0.625465
44.0	0.5401	0.028996	0.029028	-0.024670	0.030999	0.606075	0.625494
45.0	0.5402	0.028540	0.028572	-0.024272	0.031398	0.606074	0.625492
46.0	0.5403	0.028082	0.028113	-0.023880	0.031790	0.606053	0.625471
47.0	0.5405	0.027620	0.027650	-0.023494	0.032176	0.606048	0.625467
48.0	0.5406	0.027158	0.027188	-0.023115	0.032555	0.606057	0.625476
49.0	0.5408	0.026694	0.026724	-0.022742	0.032928	0.606061	0.625480
50.0	0.5410	0.026232	0.026261	-0.022376	0.033294	0.606059	0.625478
51.0	0.5412	0.025770	0.025798	-0.022016	0.033655	0.606059	0.625477
52.0	0.5414	0.025311	0.025339	-0.021662	0.034008	0.606059	0.625477
53.0	0.5416	0.024855	0.024882	-0.021315	0.034356	0.606059	0.625477
54.0	0.5419	0.024402	0.024429	-0.020974	0.034697	0.606059	0.625477
55.0	0.5421	0.023954	0.023980	-0.020640	0.035032	0.606059	0.625477
56.0	0.5423	0.023510	0.023536	-0.020311	0.035360	0.606059	0.625477
57.0	0.5425	0.023071	0.023096	-0.019989	0.035683	0.606059	0.625477
58.0	0.5428	0.022638	0.022663	-0.019672	0.035999	0.606059	0.625477
59.0	0.5430	0.022211	0.022235	-0.019362	0.036310	0.606059	0.625477

60.0	0.5432	0.021790	0.021814	-0.019057	0.036615	0.606059	0.625477
61.0	0.5435	0.021375	0.021399	-0.018758	0.036914	0.606059	0.625477
62.0	0.5437	0.020967	0.020990	-0.018465	0.037207	0.606059	0.625477
63.0	0.5439	0.020561	0.020584	-0.018178	0.037494	0.606059	0.625477
64.0	0.5442	0.020164	0.020186	-0.017896	0.037777	0.606059	0.625477
65.0	0.5444	0.019775	0.019796	-0.017619	0.038053	0.606059	0.625477
66.0	0.5446	0.019393	0.019414	-0.017348	0.038324	0.606059	0.625477
67.0	0.5448	0.019019	0.019040	-0.017082	0.038590	0.606059	0.625477
68.0	0.5450	0.018653	0.018674	-0.016822	0.038851	0.606059	0.625477
69.0	0.5453	0.018295	0.018315	-0.016566	0.039107	0.606059	0.625477
70.0	0.5455	0.017944	0.017964	-0.016315	0.039358	0.606059	0.625477
71.0	0.5456	0.017601	0.017620	-0.016069	0.039604	0.606059	0.625477
72.0	0.5458	0.017266	0.017285	-0.015828	0.039846	0.606059	0.625477
73.0	0.5460	0.016937	0.016956	-0.015591	0.040082	0.606059	0.625477
74.0	0.5462	0.016617	0.016635	-0.015359	0.040315	0.606059	0.625477
75.0	0.5464	0.016303	0.016320	-0.015131	0.040543	0.606059	0.625477
76.0	0.5465	0.015996	0.016013	-0.014907	0.040766	0.606059	0.625477
77.0	0.5467	0.015696	0.015713	-0.014688	0.040986	0.606059	0.625477
78.0	0.5469	0.015403	0.015420	-0.014473	0.041201	0.606059	0.625477
79.0	0.5470	0.015117	0.015134	-0.014261	0.041413	0.606059	0.625477
80.0	0.5471	0.014838	0.014854	-0.014054	0.041620	0.606059	0.625477
81.0	0.5473	0.014565	0.014581	-0.013851	0.041824	0.606059	0.625477
82.0	0.5474	0.014298	0.014314	-0.013651	0.042024	0.606059	0.625477
83.0	0.5475	0.014038	0.014053	-0.013455	0.042220	0.606059	0.625477
84.0	0.5477	0.013784	0.013799	-0.013262	0.042412	0.606059	0.625477
85.0	0.5478	0.013536	0.013551	-0.013073	0.042602	0.606059	0.625477
86.0	0.5479	0.013293	0.013308	-0.012887	0.042787	0.606059	0.625477
87.0	0.5480	0.013057	0.013071	-0.012705	0.042970	0.606059	0.625477
88.0	0.5481	0.012826	0.012840	-0.012526	0.043149	0.606059	0.625477
89.0	0.5482	0.012600	0.012614	-0.012350	0.043325	0.606059	0.625477
90.0	0.5483	0.012380	0.012393	-0.012177	0.043498	0.606059	0.625477
91.0	0.5484	0.012165	0.012178	-0.012007	0.043668	0.606059	0.625477
92.0	0.5485	0.011955	0.011968	-0.011840	0.043835	0.606059	0.625477
93.0	0.5486	0.011749	0.011762	-0.011676	0.043999	0.606059	0.625477
94.0	0.5486	0.011549	0.011562	-0.011515	0.044161	0.606059	0.625477
95.0	0.5487	0.011353	0.011366	-0.011356	0.044319	0.606059	0.625477
96.0	0.5488	0.011162	0.011175	-0.011200	0.044475	0.606059	0.625477
97.0	0.5489	0.010976	0.010988	-0.011047	0.044628	0.606059	0.625477
98.0	0.5489	0.010794	0.010805	-0.010896	0.044779	0.606059	0.625477
99.0	0.5490	0.010616	0.010627	-0.010748	0.044928	0.606059	0.625477

100.0	0.5490	0.010442	0.010453	-0.010602	0.045073	0.606059	0.625477
101.0	0.5491	0.010272	0.010283	-0.010459	0.045217	0.606059	0.625477
102.0	0.5491	0.010106	0.010117	-0.010318	0.045358	0.606059	0.625477
103.0	0.5492	0.009944	0.009955	-0.010179	0.045497	0.606059	0.625477
104.0	0.5492	0.009786	0.009796	-0.010043	0.045633	0.606059	0.625477
105.0	0.5493	0.009631	0.009642	-0.009908	0.045768	0.606059	0.625477
106.0	0.5493	0.009480	0.009490	-0.009776	0.045900	0.606059	0.625477
107.0	0.5493	0.009332	0.009342	-0.009646	0.046030	0.606059	0.625477
108.0	0.5494	0.009188	0.009198	-0.009517	0.046159	0.606059	0.625477
109.0	0.5494	0.009047	0.009057	-0.009391	0.046285	0.606059	0.625477
110.0	0.5494	0.008909	0.008919	-0.009267	0.046409	0.606059	0.625477
111.0	0.5495	0.008774	0.008784	-0.009145	0.046532	0.606059	0.625477
112.0	0.5495	0.008643	0.008652	-0.009024	0.046652	0.606059	0.625477
113.0	0.5495	0.008514	0.008523	-0.008905	0.046771	0.606059	0.625477
114.0	0.5495	0.008388	0.008398	-0.008788	0.046888	0.606059	0.625477
115.0	0.5495	0.008265	0.008274	-0.008673	0.047004	0.606059	0.625477
116.0	0.5495	0.008145	0.008154	-0.008559	0.047117	0.606059	0.625477
117.0	0.5496	0.008028	0.008036	-0.008447	0.047229	0.606059	0.625477
118.0	0.5496	0.007913	0.007921	-0.008337	0.047340	0.606059	0.625477
119.0	0.5496	0.007800	0.007809	-0.008228	0.047448	0.606059	0.625477
120.0	0.5496	0.007691	0.007699	-0.008121	0.047556	0.606059	0.625477
121.0	0.5496	0.007583	0.007591	-0.008015	0.047661	0.606059	0.625477
122.0	0.5496	0.007478	0.007486	-0.007911	0.047766	0.606059	0.625477
122.5	0.5496	0.007426	0.007434	-0.007859	0.047818	0.606070	0.625488
123.0	0.5496	0.007375	0.007383	-0.007808	0.047869	0.606091	0.625508
124.0	0.5496	0.007275	0.007283	-0.007707	0.047970	0.606008	0.625433

125.0	0.5496	0.007176	0.007184	-0.007607	0.048070	0.605888	0.625315
126.0	0.5496	0.007080	0.007088	-0.007508	0.048169	0.605668	0.625103
127.0	0.5496	0.006986	0.006993	-0.007411	0.048266	0.605392	0.624831
128.0	0.5496	0.006893	0.006901	-0.007315	0.048362	0.605047	0.624492
129.0	0.5496	0.006803	0.006811	-0.007220	0.048457	0.604634	0.624084
130.0	0.5495	0.006715	0.006722	-0.007126	0.048551	0.604153	0.623609
131.0	0.5495	0.006628	0.006635	-0.007034	0.048643	0.603606	0.623067
132.0	0.5495	0.006543	0.006551	-0.006943	0.048734	0.602992	0.622459
133.0	0.5495	0.006461	0.006468	-0.006853	0.048825	0.602312	0.621784
134.0	0.5495	0.006379	0.006386	-0.006764	0.048913	0.601566	0.621044
135.0	0.5495	0.006300	0.006307	-0.006676	0.049001	0.600755	0.620238
136.0	0.5495	0.006222	0.006229	-0.006589	0.049088	0.599879	0.619368
137.0	0.5494	0.006146	0.006153	-0.006504	0.049174	0.598939	0.618433
138.0	0.5494	0.006071	0.006078	-0.006419	0.049258	0.597935	0.617434
139.0	0.5494	0.005998	0.006005	-0.006336	0.049342	0.596868	0.616373
140.0	0.5494	0.005926	0.005933	-0.006253	0.049425	0.595738	0.615248
141.0	0.5493	0.005856	0.005863	-0.006171	0.049506	0.594546	0.614062
142.0	0.5493	0.005788	0.005794	-0.006091	0.049587	0.593292	0.612814
143.0	0.5493	0.005720	0.005726	-0.006011	0.049667	0.591978	0.611505
144.0	0.5493	0.005654	0.005660	-0.005932	0.049745	0.590604	0.610136
145.0	0.5492	0.005590	0.005596	-0.005855	0.049823	0.589170	0.608708
146.0	0.5492	0.005526	0.005532	-0.005778	0.049900	0.587677	0.607220
147.0	0.5492	0.005464	0.005470	-0.005701	0.049976	0.586127	0.605675
148.0	0.5492	0.005403	0.005409	-0.005626	0.050052	0.584519	0.604072
149.0	0.5491	0.005344	0.005350	-0.005552	0.050126	0.582854	0.602412

150.0	0.5491	0.005285	0.005291	-0.005478	0.050200	0.581133	0.600697
151.0	0.5491	0.005228	0.005234	-0.005405	0.050273	0.579357	0.598926
152.0	0.5490	0.005172	0.005178	-0.005333	0.050345	0.577527	0.597101
153.0	0.5490	0.005117	0.005123	-0.005262	0.050416	0.575643	0.595222
154.0	0.5490	0.005063	0.005069	-0.005191	0.050487	0.573706	0.593291
155.0	0.5489	0.005010	0.005016	-0.005121	0.050557	0.571717	0.591307
156.0	0.5489	0.004959	0.004964	-0.005052	0.050626	0.569677	0.589272
157.0	0.5488	0.004908	0.004913	-0.004984	0.050694	0.567586	0.587187
158.0	0.5488	0.004858	0.004863	-0.004916	0.050762	0.565446	0.585052
159.0	0.5488	0.004809	0.004815	-0.004849	0.050829	0.563257	0.582868
160.0	0.5487	0.004761	0.004767	-0.004783	0.050895	0.561020	0.580636
161.0	0.5487	0.004714	0.004720	-0.004717	0.050961	0.558737	0.578358
162.0	0.5487	0.004668	0.004674	-0.004652	0.051026	0.556407	0.576033
163.0	0.5486	0.004623	0.004628	-0.004588	0.051091	0.554031	0.573663
164.0	0.5486	0.004579	0.004584	-0.004524	0.051154	0.551612	0.571248
165.0	0.5485	0.004536	0.004541	-0.004461	0.051218	0.549149	0.568790
166.0	0.5485	0.004493	0.004498	-0.004398	0.051280	0.546643	0.566289
167.0	0.5485	0.004451	0.004456	-0.004336	0.051342	0.544095	0.563746
168.0	0.5484	0.004410	0.004415	-0.004275	0.051404	0.541507	0.561163
169.0	0.5484	0.004370	0.004375	-0.004214	0.051465	0.538878	0.558539
170.0	0.5483	0.004331	0.004336	-0.004153	0.051525	0.536210	0.555877
171.0	0.5483	0.004292	0.004297	-0.004094	0.051585	0.533505	0.553176
172.0	0.5482	0.004254	0.004259	-0.004034	0.051644	0.530762	0.550438
173.0	0.5482	0.004217	0.004222	-0.003975	0.051703	0.527982	0.547663
174.0	0.5481	0.004181	0.004185	-0.003917	0.051762	0.525168	0.544853
175.0	0.5481	0.004145	0.004150	-0.003859	0.051819	0.522318	0.542008
176.0	0.5481	0.004110	0.004115	-0.003802	0.051877	0.519435	0.539130
177.0	0.5480	0.004076	0.004080	-0.003745	0.051934	0.516519	0.536219
178.0	0.5480	0.004042	0.004046	-0.003689	0.051990	0.513572	0.533276
179.0	0.5479	0.004009	0.004013	-0.003633	0.052046	0.510593	0.530303
180.0	0.5479	0.003976	0.003981	-0.003577	0.052101	0.507585	0.527299
181.0	0.5478	0.003944	0.003949	-0.003522	0.052156	0.504547	0.524266
182.0	0.5478	0.003913	0.003918	-0.003468	0.052211	0.501482	0.521205
183.0	0.5477	0.003883	0.003887	-0.003414	0.052265	0.498388	0.518116
184.0	0.5477	0.003852	0.003857	-0.003360	0.052319	0.495269	0.515001
185.0	0.5476	0.003823	0.003827	-0.003307	0.052372	0.492124	0.511861
186.0	0.5476	0.003794	0.003798	-0.003254	0.052425	0.488954	0.508696
187.0	0.5475	0.003766	0.003770	-0.003201	0.052478	0.485761	0.505507
188.0	0.5475	0.003738	0.003742	-0.003149	0.052530	0.482545	0.502295
189.0	0.5474	0.003711	0.003715	-0.003097	0.052582	0.479307	0.499061
190.0	0.5474	0.003684	0.003688	-0.003046	0.052633	0.476048	0.495807
191.0	0.5473	0.003658	0.003662	-0.002995	0.052684	0.472768	0.492532
192.0	0.5473	0.003632	0.003636	-0.002944	0.052735	0.469470	0.489237
193.0	0.5472	0.003606	0.003610	-0.002894	0.052785	0.466152	0.485924
194.0	0.5472	0.003582	0.003586	-0.002844	0.052835	0.462817	0.482594
195.0	0.5471	0.003557	0.003561	-0.002794	0.052885	0.459466	0.479246
196.0	0.5471	0.003534	0.003537	-0.002745	0.052934	0.456098	0.475883
197.0	0.5470	0.003510	0.003514	-0.002696	0.052983	0.452715	0.472504
198.0	0.5470	0.003487	0.003491	-0.002647	0.053032	0.449318	0.469111
199.0	0.5469	0.003465	0.003469	-0.002599	0.053080	0.445908	0.465705

200.0	0.5467	0.003442	0.003446	-0.002551	0.053129	0.442556	0.462208
210.0	0.5322	0.003145	0.003148	-0.002091	0.053589	0.420906	0.439635
220.0	0.4976	0.002756	0.002759	-0.001679	0.054000	0.416944	0.435533
230.0	0.4486	0.002324	0.002326	-0.001325	0.054324	0.416944	0.435533
240.0	0.3912	0.001893	0.000000	-0.001031	0.054356	0.416944	0.000000
250.0	0.3381	0.001535	0.000000	-0.000794	0.054057	0.416944	0.000000
260.0	0.2944	0.001265	0.000000	-0.000601	0.053670	0.416944	0.000000
270.0	0.2603	0.001071	0.000000	-0.000448	0.053214	0.416944	0.000000
280.0	0.2364	0.000946	0.000000	-0.000332	0.052671	0.416739	0.000000
290.0	0.2227	0.000880	0.000000	-0.000249	0.052015	0.408153	0.000000
300.0	0.2170	0.000863	0.000000	-0.000183	0.051220	0.386253	0.000000
310.0	0.2137	0.000862	0.000000	-0.000122	0.050252	0.360034	0.000000
320.0	0.2095	0.000864	0.000000	-0.000061	0.049076	0.335242	0.000000
330.0	0.2043	0.000866	0.000000	0.000002	0.047667	0.311968	0.000000
340.0	0.1982	0.000869	0.000000	0.000066	0.045989	0.290260	0.000000
350.0	0.1913	0.000873	0.000000	0.000132	0.043792	0.270183	0.000000
360.0	0.1838	0.000879	0.000000	0.000200	0.040721	0.251882	0.000000
370.0	0.1759	0.000885	0.000000	0.000271	0.036687	0.235688	0.000000
380.0	0.1672	0.000891	0.000000	0.000344	0.031997	0.221917	0.000000
390.0	0.1574	0.000897	0.000000	0.000420	0.027166	0.210741	0.000000
400.0	0.1465	0.000899	0.000000	0.000498	0.022650	0.202029	0.000000
410.0	0.1347	0.000899	0.000000	0.000576	0.018701	0.195487	0.000000
420.0	0.1224	0.000897	0.000000	0.000653	0.015382	0.190746	0.000000
430.0	0.1099	0.000893	0.000000	0.000728	0.012639	0.187452	0.000000
440.0	0.0975	0.000888	0.000000	0.000801	0.010378	0.185315	0.000000
450.0	0.0857	0.000883	0.000000	0.000870	0.008500	0.184063	0.000000
460.0	0.0746	0.000878	0.000000	0.000937	0.006926	0.183537	0.000000
470.0	0.0643	0.000872	0.000000	0.000998	0.005590	0.183593	0.000000
480.0	0.0551	0.000866	0.000000	0.001052	0.004444	0.184061	0.000000
490.0	0.0471	0.000861	0.000000	0.001089	0.003450	0.184627	0.000000
500.0	0.0405	0.000000	0.000000	0.001063	0.002576	0.000000	0.000000
510.0	0.0353	0.000000	0.000000	0.001000	0.001801	0.000000	0.000000
520.0	0.0315	0.000000	0.000000	0.000913	0.001106	0.000000	0.000000
530.0	0.0293	0.000000	0.000000	0.000802	0.000477	0.000000	0.000000
540.0	0.0287	0.000000	0.000000	0.000671	-0.000099	0.000000	0.000000
550.0	0.0298	0.000000	0.000000	0.000517	-0.000631	0.000000	0.000000

HEAT TRANSFER DATA

CA (DEG)	VELHTX (CM/SEC)	HTPARO (KW/M**2)	HTPASI (KW/M**2)	HTPAHO (KW/M**2)	Q% ROTOR (%)	Q% SIDE (%)	Q% HOUSING (%)
-520.0	824.7	15979107.0	15979107.0	15979107.0	43.831	3.556	52.613
-510.0	824.7	15218337.0	15218337.0	15218337.0	43.160	4.320	52.520
-505.8	824.7	15009003.0	15009003.0	15009003.0	42.810	4.720	52.471
-500.0	824.7	14775061.0	14775061.0	14775061.0	42.267	5.338	52.395
-492.9	824.7	14440517.0	14440517.0	14440517.0	41.519	6.190	52.291
-490.0	824.7	13599877.0	13599877.0	13599877.0	41.191	6.564	52.245
-480.0	824.7	9980390.0	9980390.0	9980390.0	39.976	7.949	52.075
-470.0	824.7	7309748.0	7309748.0	7309748.0	38.665	9.443	51.892
-460.0	824.7	5320346.0	5320346.0	5320346.0	37.299	10.999	51.702
-450.0	824.7	3835557.0	3835557.0	3835557.0	35.916	12.576	51.508
-440.0	824.7	2700186.0	2700186.0	2700186.0	34.546	14.137	51.317
-430.0	824.7	1818480.0	1818480.0	1818480.0	33.213	15.656	51.131
-420.0	824.7	1125338.0	1125338.0	1125338.0	31.937	17.110	50.953
-410.0	824.7	574305.7	574305.7	574305.7	30.732	18.483	50.785
-400.0	824.7	127642.1	127642.1	127642.1	29.608	19.764	50.628
-390.0	824.7	-232376.7	-232376.7	-232376.7	28.569	20.948	50.483
-380.0	824.7	-522968.9	-522968.9	-522968.9	27.620	22.029	50.350
-370.0	824.7	-754987.0	-754987.0	-754987.0	26.762	23.008	50.231
-360.0	824.7	-936047.4	-936047.4	-936047.4	25.993	23.883	50.123
-350.0	824.7	-1075939.0	-1075939.0	-1075939.0	25.314	24.658	50.029
-340.0	824.7	-1182815.0	-1182815.0	-1182815.0	24.721	25.333	49.946
-330.0	824.7	-1263735.0	-1263735.0	-1263735.0	24.214	25.911	49.875
-320.0	824.7	-1325647.0	-1325647.0	-1325647.0	23.788	26.396	49.816
-310.0	824.7	-1371296.0	-1371296.0	-1371296.0	23.444	26.789	49.767
-300.0	824.7	-1401694.0	-1401694.0	-1401694.0	23.178	27.092	49.730
-290.0	824.7	-1418573.0	-1418573.0	-1418573.0	22.989	27.307	49.704
-280.0	824.7	-1421271.0	-1421271.0	-1421271.0	22.876	27.436	49.688
-277.3	824.7	-1419089.0	-1419089.0	-1419089.0	22.859	27.456	49.686
-270.0	824.7	-1410017.0	-1410017.0	-1410017.0	22.839	27.478	49.683
-260.0	824.7	-1392293.0	-1392293.0	-1392293.0	22.876	27.436	49.688
-250.0	824.7	-1357968.0	-1357968.0	-1357968.0	22.989	27.307	49.704
-240.0	824.7	-1295586.0	-1295586.0	-1295586.0	23.178	27.092	49.730
-230.0	824.7	-1178865.0	-1178865.0	-1178865.0	23.444	26.789	49.767
-220.0	824.7	-1002928.9	-1002928.9	-1002928.9	23.789	26.396	49.816
-210.0	824.7	-770007.9	-770007.9	-770007.9	24.214	25.911	49.875
-200.0	824.7	-471504.7	-471504.7	-471504.7	24.721	25.333	49.946
-190.0	824.7	-92780.1	-92780.1	-92780.1	25.314	24.658	50.029
-180.0	824.7	386404.4	386404.4	386404.4	25.993	23.883	50.123

-170.0	824.7	992982.4	992982.4	992982.4	26.762	23.008	50.231
-160.0	824.7	1762636.0	1762636.0	1762636.0	27.620	22.029	50.350
-150.0	824.7	2742778.0	2742778.0	2742778.0	28.569	20.948	50.483
-140.0	824.7	3996364.0	3996364.0	3996364.0	29.608	19.764	50.628
-130.0	824.7	5607554.0	5607554.0	5607554.0	30.732	18.483	50.785
-120.0	824.7	7680330.0	7680330.0	7680330.0	31.937	17.110	50.953
-110.0	824.7	10353726.0	10353726.0	10353726.0	33.213	15.656	51.131
-100.0	824.7	13833265.0	13833265.0	13833265.0	34.546	14.137	51.317
-90.0	824.7	18374720.0	18374720.0	18374720.0	35.916	12.576	51.508
-80.0	824.7	24332560.0	24332560.0	24332560.0	37.299	10.999	51.702
-70.0	824.7	32066320.0	32066320.0	32066320.0	38.665	9.443	51.892
-60.0	824.7	41925712.0	41925712.0	41925712.0	39.976	7.949	52.075
-50.0	824.7	54154512.0	54154512.0	54154512.0	41.191	6.564	52.245
-40.0	824.7	68577840.0	68577840.0	68577840.0	42.267	5.338	52.395
-30.0	824.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-30.0	824.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-29.0	824.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-28.0	824.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-27.0	824.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-26.0	825.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-25.0	825.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-24.0	827.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-23.0	828.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-22.0	830.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-21.0	833.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-20.0	836.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-19.0	839.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-18.0	843.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-17.0	848.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-16.0	853.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-15.0	858.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-14.0	864.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-13.0	871.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-12.0	878.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-11.0	885.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-10.0	893.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-9.0	901.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-8.0	910.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-7.0	919.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-6.0	929.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-5.0	938.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-4.0	948.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-3.0	959.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-2.0	969.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
-1.0	980.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
0.0	991.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
1.0	1002.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
2.0	1013.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
3.0	1024.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
4.0	1035.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
5.0	1046.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
6.0	1057.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
7.0	1068.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
8.0	1079.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
9.0	1090.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520

10.0	1101.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
11.0	1112.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
12.0	1122.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
13.0	1132.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
14.0	1142.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
15.0	1152.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
16.0	1162.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
17.0	1171.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
18.0	1180.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
19.0	1189.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
20.0	1197.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
21.0	1205.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
22.0	1213.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
23.0	1220.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
24.0	1227.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
25.0	1234.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
26.0	1241.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
27.0	1247.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
28.0	1252.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
29.0	1258.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
30.0	1263.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
31.0	1268.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
32.0	1272.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
33.0	1276.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
34.0	1280.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
35.0	1283.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
36.0	1286.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
37.0	1289.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
38.0	1292.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
39.0	1294.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
40.0	1296.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
40.3	1296.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
41.0	1297.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
42.0	1299.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
43.0	1300.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
44.0	1301.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
45.0	1302.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
46.0	1302.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
47.0	1302.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
48.0	1302.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
49.0	1302.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
50.0	1302.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
51.0	1302.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
52.0	1301.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
53.0	1301.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
54.0	1300.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
55.0	1299.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
56.0	1298.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
57.0	1297.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
58.0	1296.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
59.0	1294.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520

60.0	1293.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
61.0	1292.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
62.0	1290.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
63.0	1288.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
64.0	1287.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
65.0	1285.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
66.0	1283.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
67.0	1282.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
68.0	1280.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
69.0	1278.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
70.0	1276.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
71.0	1275.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
72.0	1273.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
73.0	1271.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
74.0	1269.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
75.0	1268.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
76.0	1266.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
77.0	1264.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
78.0	1262.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
79.0	1260.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
80.0	1259.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
81.0	1257.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
82.0	1255.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
83.0	1253.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
84.0	1252.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
85.0	1250.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
86.0	1248.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
87.0	1246.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
88.0	1245.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
89.0	1243.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
90.0	1241.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
91.0	1240.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
92.0	1238.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
93.0	1236.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
94.0	1235.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
95.0	1233.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
96.0	1232.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
97.0	1230.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
98.0	1228.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
99.0	1227.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520

100.0	1225.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
101.0	1224.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
102.0	1222.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
103.0	1221.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
104.0	1219.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
105.0	1218.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
106.0	1216.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
107.0	1215.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
108.0	1213.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
109.0	1212.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
110.0	1210.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
111.0	1209.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
112.0	1208.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
113.0	1206.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
114.0	1205.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
115.0	1203.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
116.0	1202.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
117.0	1201.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
118.0	1199.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
119.0	1198.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
120.0	1197.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
121.0	1195.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
122.0	1194.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
122.5	1193.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
123.0	1193.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
124.0	1192.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
125.0	1190.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
126.0	1189.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
127.0	1188.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
128.0	1187.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
129.0	1185.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
130.0	1184.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
131.0	1183.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
132.0	1182.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
133.0	1181.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
134.0	1179.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
135.0	1178.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
136.0	1177.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
137.0	1176.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
138.0	1175.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
139.0	1174.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
140.0	1173.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
141.0	1172.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
142.0	1171.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
143.0	1170.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
144.0	1168.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
145.0	1167.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
146.0	1166.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
147.0	1165.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
148.0	1164.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
149.0	1163.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520

150.0	1162.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
151.0	1161.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
152.0	1160.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
153.0	1159.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
154.0	1158.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
155.0	1158.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
156.0	1157.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
157.0	1156.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
158.0	1155.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
159.0	1154.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
160.0	1153.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
161.0	1152.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
162.0	1151.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
163.0	1150.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
164.0	1149.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
165.0	1149.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
166.0	1148.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
167.0	1147.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
168.0	1146.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
169.0	1145.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
170.0	1144.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
171.0	1144.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
172.0	1143.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
173.0	1142.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
174.0	1141.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
175.0	1141.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
176.0	1140.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
177.0	1139.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
178.0	1138.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
179.0	1138.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
180.0	1137.2	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
181.0	1136.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
182.0	1135.8	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
183.0	1135.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
184.0	1134.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
185.0	1133.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
186.0	1133.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
187.0	1132.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
188.0	1131.6	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
189.0	1131.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
190.0	1130.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
191.0	1129.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
192.0	1129.0	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
193.0	1128.4	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
194.0	1127.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
195.0	1127.1	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
196.0	1126.5	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
197.0	1125.9	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
198.0	1125.3	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520
199.0	1124.7	84162576.0	84162576.0	84162576.0	43.160	4.320	52.520

200.0	824.7	94743088.0	94743088.0	94743088.0	24.721	25.333	49.946
210.0	824.7	86922304.0	86922304.0	86922304.0	24.214	25.911	49.875
220.0	824.7	76551712.0	76551712.0	76551712.0	23.788	26.396	49.816
230.0	824.7	64925184.0	64925184.0	64925184.0	23.444	26.789	49.767
240.0	824.7	53187584.0	53187584.0	53187584.0	23.178	27.092	49.730
250.0	824.7	43313024.0	43313024.0	43313024.0	22.989	27.307	49.704
260.0	824.7	35775392.0	35775392.0	35775392.0	22.876	27.436	49.688
270.0	824.7	30287088.0	30287088.0	30287088.0	22.839	27.478	49.683
280.0	824.7	26669328.0	26669328.0	26669328.0	22.876	27.436	49.688
290.0	824.7	24758288.0	24758288.0	24758288.0	22.989	27.307	49.704
300.0	824.7	24188000.0	24188000.0	24188000.0	23.178	27.092	49.730
310.0	824.7	24108656.0	24108656.0	24108656.0	23.444	26.789	49.767
320.0	824.7	24072784.0	24072784.0	24072784.0	23.788	26.396	49.816
330.0	824.7	24045152.0	24045152.0	24045152.0	24.214	25.911	49.875
340.0	824.7	24022608.0	24022608.0	24022608.0	24.721	25.333	49.946
350.0	824.7	24004288.0	24004288.0	24004288.0	25.314	24.658	50.029
360.0	824.7	23986336.0	23986336.0	23986336.0	25.993	23.883	50.123
370.0	824.7	23956560.0	23956560.0	23956560.0	26.762	23.008	50.231
380.0	824.7	23897984.0	23897984.0	23897984.0	27.620	22.029	50.350
390.0	824.7	23795440.0	23795440.0	23795440.0	28.569	20.948	50.483
400.0	824.7	23642640.0	23642640.0	23642640.0	29.608	19.764	50.628
410.0	824.7	23443472.0	23443472.0	23443472.0	30.732	18.483	50.785
420.0	824.7	23206464.0	23206464.0	23206464.0	31.937	17.110	50.953
430.0	824.7	22944304.0	22944304.0	22944304.0	33.213	15.656	51.131
440.0	824.7	22666560.0	22666560.0	22666560.0	34.546	14.137	51.317
450.0	824.7	22387040.0	22387040.0	22387040.0	35.916	12.576	51.508
460.0	824.7	22114080.0	22114080.0	22114080.0	37.299	10.999	51.702
470.0	824.7	21823904.0	21823904.0	21823904.0	38.665	9.443	51.892
480.0	824.7	21530160.0	21530160.0	21530160.0	39.976	7.949	52.075
490.0	824.7	21238768.0	21238768.0	21238768.0	41.191	6.564	52.245
500.0	824.7	20995952.0	20995952.0	20995952.0	42.267	5.338	52.395
510.0	824.7	20732416.0	20732416.0	20732416.0	43.160	4.320	52.520
520.0	824.7	20485440.0	20485440.0	20485440.0	43.831	3.556	52.613
530.0	824.7	20254048.0	20254048.0	20254048.0	44.247	3.082	52.671
540.0	824.7	20058160.0	20058160.0	20058160.0	44.388	2.921	52.691
550.0	824.7	19856864.0	19856864.0	19856864.0	44.247	3.082	52.671