THE UNIVERSITY OF ADELAIDE

DEPARTMENT OF MECHANICAL ENGINEERING

NOISE CHARACTERISTICS AND EXHAUST PROCESS GAS DYNAMICS OF A SMALL 2-STROKE ENGINE

by

Adrian David Jones, B.E. (Hons.)

Thesis for the Degree of Doctor of Philosophy

December, 1978

Awarded April 1980

```
      PROGRAM SPLMUF(INPUT,OUTPUT,TAPE1,TAPE75=INPUT,TAPE76=OUTPUT)
C
C        DECK FOR 2-STROKING ENGINE, TUNED EXHAUST SYSTEM WITH TWIN
C        CONES SEPARATED BY STRAIGHT SECTION, STRAIGHT INLET AND OUTLET
C        PIPES FROM TWIN CONES.
C
C        PROGRAM IS A STARTING PROGRAM USED TO SET DATA AND LIMITS ON
C        ARRAY SIZES.
C
C        AUTHOR   ADRIAN JONES.
C                 MECHANICAL ENGINEERING DEPARTMENT
C                 UNIVERSITY OF ADELAIDE.
C
C
C
C-----------------------------------------------------------------------
C
C
C           PROGRAM WILL CALCULATE WAVE DIAGRAM FOR A TUNED EXHAUST
C         SYSTEM.
C        THE PRESSURE AT A POINT IN THE EXHAUST SYSTEM, THE VELOCITY AT
C         THE TAILPIPE OUTLET AND THE RADIATED SOUND PRESSURE AT A POINT
C         IN THE FAR FIELD ARE ALL PLOTTED.
C        THE RMS SPL IS FOUND FOR BOTH PRESSURE TRACES.
C        PROGRAM ASSUMES NO FRICTION WITH PIPE WALLS, NO HEAT LOSS
C         THROUGH WALLS, BUT GAS IN PIPE HAS OTHERWISE CORRECT ENTROPY
C         VALUES.
C           PROGRAM ALLOWS FOR A TUNED SYSTEM WITH 5 SEGMENTS.
C         FIRST SEGMENT IS STRAIGHT, 2ND IS EXPANDING CONE SECTION,
C         3RD IS STRAIGHT, 4TH IS CONTRACTING CONE AND 5TH IS STRAIGHT
C         TAILPIPE.
C         THERE ARE THREE DIAMETERS AND FIVE LENGTHS.
C         THE DIAMETERS ARE D1 TO D3 AND LENGTHS ARE XL1 TO XL5.
C
C-----------------------------------------------------------------------
C
C
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /B/ B,BBQ,BC,BORE,BOREA
      COMMON /C/ CONLEN,CP,CTX
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /I/ I,II,IJ
      COMMON /J/ J,JA,JJ,JREV,JREX
      COMMON /K/ KJ,KK
      COMMON /M/ M,MM
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1           ,NWAVES,NWVDIS
      COMMON /O/ ONETHD
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1           ,PRATIC,PRATIO,PX(1200),PO
      COMMON /Q/ Q
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1           ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1           ,TLAMIN,TMEVNN(100),THEVNT(100),TRDUC,TRDUCE,TSTART
```

I G

```
      FUNCTION VOLEX ( FN,EX,EVS,ER )
C
C     FUNCTION CALCULATES EXACT (TO HIGH ACCURACY) VOLUME IN ENGINE
C      CYLINDER OR CRANKCASE, TAKING INTO ACCOUNT THE CON-ROD OBLIQUITY.
C      (SPECIFICALLY THEIR INITIAL VALUES AT E.P.O. AND T.P.O.)
C
C-----------------------------------------------------------------------
C
      COMMON /B/ B,BBQ,BC,BORE,BOREA
      COMMON /C/ CONLEN,CP,CTX
      COMMON /Q/ Q
C
C-----------------------------------------------------------------------
C
      AASIN = ( SIN( EX )*Q )**2/2.
      FITERM = AASIN*( 1.+ AASIN*( 1.+ AASIN*( 1.+ 5.*AASIN/4. ) )/2. )
      VOLEX = EVS*( 1./( ER-1. ) + .5 + .5*FN*COS( EX ) )
     1      + FN*CONLEN*BOREA*FITERM
      RETURN
      END
```

H.4

```
      FUNCTION RMSHRM ( M,P,N,B,T )
C
      DIMENSION P(M),T(M)
C
C     FUNCTION CALCULATES THE RMS VALUE OF THE NTH HARMONIC OF A CYCLIC
C      SET OF VALUES AVERAGED ABOUT ZERO. THERE ARE M VALUES IN ARRAY P.
C
C-----------------------------------------------------------------------
C
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
C
C-----------------------------------------------------------------------
C
      AREAA = 0.
      AREAB = 0.
      X = PI*RPMN/60.
      Y = 2.*FLOAT( N )*X
      AA = T(1) * Y
      CS = COS( AA )
      SN = SIN( AA )
      CSN = CS
      SNN = SN
      DO 1 NB=2,M
C     BELOW WE CONSIDER ALL POINTS OTHER THAN LAST.
      AA = T(NB) * Y
      CSNP = COS( AA )
      SNNP = SIN( AA )
      AREAA = AREAA + ( T(NB)-T(NB-1) ) * ( P(NB)*CSNP + P(NB-1)*CSN )
      AREAB = AREAB + ( T(NB)-T(NB-1) ) * ( P(NB)*SNNP + P(NB-1)*SNN )
      CSN = CSNP
      SNN = SNNP
    1 CONTINUE
      AREAA = AREAA + ( 60./RPMN - T(M) + T(1) )*( P(1)*CS + P(M)*CSN )
      AREAB = AREAB + ( 60./RPMN - T(M) + T(1) )*( P(1)*SN + P(M)*SNN )
      AREAA = AREAA*X/PI
      AREAB = AREAB*X/PI
C     NOW CALCULATE RMS VALUE OF NTH HARMONIC OF P.
C     B IS USED, SO THAT IF THERE IS A VALUE OF ZERO FOR THE SIGNAL AT
C      THE PARTICULAR FREQUENCY, THE VALUE OF RMSHRM AS CALCULATED WILL
C      GIVE 0.0 DB.
      RMSHRM  = SQRT( ( AREAA**2 + AREAB**2 )/2. + 1./B**2 )
      RETURN
      END
```

```
      SLFE(N) = FP
      SZI(N) = D
      ZI(N) = FLOAT( N )*ZLN/( FLOAT( JA ) + 1. )
      NTYPE(N) = 3
      TMEVNT(N) = 0.
    4 CONTINUE
      RETURN
      END
```

F4

```
      SUBROUTINE WVSTAT
C
C     THIS SUBROUTINE SETS THE VALUES FOR THE INITIAL C+ AND C-
C      CHARACTERISTICS (PRESSURE WAVES) WHICH ARE IN THE EXHAUST DUCT AT
C      TIME T=0 (THAT IS, AT E.P.O.) ON THE FIRST REVOLUTION. THE
C      VELOCITY IS EVERYWHERE SET TO ZERO, PRESSURE TO THE ATMOSPHERIC
C      VALUE, AND THE VALUE OF ENTROPY IS ASSUMED EQUAL (FOR CONVENIENCE
C      ) WITH THAT IN THE CYLINDER AT E.P.O. THIS CORRESPONDS TO A GAS
C      AT ROUGHLY THE AVERAGE TEMPERATURE OF THE SUBSEQUENT EXHAUST GAS.
C     ALL CALCULATED TERMS WILL BE NONDIMENSIONAL FORM.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /J/ J,JA,JJ,JREV,JREX
      COMMON /N/ NC,N[illegible],NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1         ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1         ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1         ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1         ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2         ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
C     B IS SIMPLIFYING EXPRESSION FOR A(N)
      B = EXP( GAMAB*( SCO/(GAMA*RGAS) - SD ) )
C     C IS GAS VELOCITY = 0.
      C = 0.
C     D IS SIMPLIFYING EXPRESSION FOR SZI(N)
      D = SCO/(GAMA*RGAS)
C     E IS SIMPLIFYING EXPRESSION FOR PQ(N),THE RIEMANN VALUE
      E = B/GAMAB
      FP = 1./B
      JB = JA - 1
      DO 1 N=1,JB,2
C     LOOP 1 WILL ADD Q WAVES
      A(N) = B
      U(N) = C
      PQ(N) = E
      SLPE(N) = -FP
      SZI(N) = D
      ZI(N) = FLOAT( N )*ZLN/( FLOAT( JA ) + 1. )
      NTYPE(N) = 2
      TMEVNT(N) = 0.
    1 CONTINUE
C     LOOP 4 WILL ADD P WAVES
      DO 4 N=2,JA,2
      A(N) = B
      U(N) = C
      PQ(N) = E
```

E4

```
      SUBROUTINE WVCLOS ( IX )
C
C     SUBROUTINE REGULARLY COMBINES TOGETHER ENTROPY CHARACTERISTICS.
C      THIS IS NECESSARY FOR THE PROGRAM TO RUN PROPERLY. WVCLOS
C      SEARCHES THE WHOLE DUCT FOR ADJACENT P CHARACTERISTICS WHICH ARE
C      CLOSER THAN A SPECIFIED DISTANCE. THIS SUBROUTINE IS CALLED AFTER
C      EVERY DTAU PERIOD OF TIME. ONLY P CHARACTERISTICS IX TO I ARE
C      CONSIDERED.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /I/ I,II,IJ
      COMMON /M/ M,MM
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1          ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
      XX = FLOAT(NWVDIS)*AO*DTAU/(FLOAT(MM)*ZLNM)
      N = IX
      IF ( IX.GT.I-1 ) STOP 500
C     PROGRAM STOPS AS WE ARE TRYING TO CONSIDER WAVES I, I+1
C      OR I+1, I+2, ETC.
    1 CONTINUE
C     BELOW IS FOR THE TWO WAVES TOO CLOSE TOGETHER. ARE THEY BOTH
C      ENTROPY WAVES.
      IF ( ZI(N+1)-ZI(N).GT.XX ) GO TO 3
      IF ( NTYPE(N+1).GT.1 .OR. NTYPE(N).GT.1 ) GO TO 3
C     BELOW IS FOR THE TWO WAVES BEING ENTROPY WAVES. THE TWO WAVES WILL
C      BE MERGED INTO ONE WITH AVERAGED VALUES.
      ZI(N) = ( ZI(N) + ZI(N+1) )/2.
      U(N) = ( U(N) + U(N+1) )/2.
      SLPE(N) = 1./U(N)
      SZI(N) = SZI(N)
      PQ(N) = PQ(N+1)
      NTYPE(N) = 1
C     WE NOW WISH TO REMOVE ONE WAVE, WAVE N+1.
      CALL REMOV1 ( N+2,.TRUE. )
      N = N - 1
    3 CONTINUE
      N = N + 1
      IF ( N.LT.I ) GO TO 1
      RETURN
      END
```

```
      SUBROUTINE WAREU ( ZI,D )
C
C     SUBROUTINE WORKS IN CONJUNCTION WITH PQADIF AND IS ONLY SUITABLE
C      FOR A SPECIFIC TYPE OF EXHAUST SYSTEM. THIS SUBROUTINE FINDS
C      WHICH SECTION OF THE EXHAUST SYSTEM A PARTICULAR C+ OR C-
C      CHARACTERISTIC IS IN, AND THEN RETURNS TO PQADIF THE
C      APPROPRIATE DUCT DIAMETER AT THE POINT AT WHICH THE
C      CHARACTERISTIC IS LOCATED.
C
C-----------------------------------------------------------------------
C
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /X/ X,XC,XC1,XE,XEC,XL1,XL3,XZ2,XZ4,X1,X2,X3,X4
C
C-----------------------------------------------------------------------
C
      IF ( ZI.GT.X2 ) GO TO 1
      IF ( ZI.GT.XL1 ) GO TO 2
C     BELOW ZI IS IN 1ST SECTION, WHICH IS STRAIGHT PIPE.
      D = D1
      RETURN
C     2 IS FOR ZI BETWEEN X1 AND X2, IN EXPANDING CONE SECTION.
    2 D = ( ZI - XL1 )*XZ2 + D1
      RETURN
C     CASE 1 IS FOR ZI IN REGIONS 3, 4 OR 5.
    1 IF ( ZI.GT.X3 ) GO TO 3
C     BELOW ZI IS IN 3RD SECTION, STRAIGHT PIPE.
      D = D2
      RETURN
C     CASE 3 IS FOR ZI IN REGIONS 4 OR 5.
    3 IF ( ZI.GT.X4 ) GO TO 4
C     BELOW ZI IS IN 4TH SECTION.
      D = ( ZI - X3 )*XZ4 + D2
      RETURN
C     CASE 4 IS FOR ZI IN 5TH REGION.
    4 D = D3
      RETURN
      END
```

```
      CALL QIKPLOT ( TIMEX,UX,-NTIM,48H*VELOCITY VALUES AT OPEN END OF P
     1IPE      MSECS*,22H*VELOCITY       MS/SEC* )
      IF ( NNN.LT.NNZ ) GO TO 6
      CALL AUTPLT ( TIMEX,UX,NTIM,1,48H*VELOCITY VALUES AT OPEN END OF P
     1IPE      MSECS*,22H*VELOCITY       MS/SEC* )
    6 CONTINUE
      RETURN
C
C-----------------------------------------------------------------------
C
    3 FORMAT (33X,55HVALUES IN ABOVE PRESSURE TRACE REFER TO MEASURING P
     1OINT,F5.1,15H MS FROM SOURCE)
      END
```

B4.

```
      SUBROUTINE VLDIFF
C
C     SUBROUTINE DIFFERENTIATES THE CYCLE OF VALUES OF VELOCITY AT THE
C      TAILPIPE OUTLET AND OBTAINS THE CYCLE OF PRESSURE VALUES IN THE
C       FAR FIELD (GREATER THAN ONE WAVELENGTH AWAY). THESE VALUES, LEFT
C      AVERAGED ABOUT ATMOSPHERIC PRESSURE, ARE PLOTTED, AND RETURNED.
C     THERE ARE M VALUES IN THE CYCLE.
C     THE TIME DIFFERENCES BETWEEN THE 1ST AND 2ND VALUES, ETC. ARE HELD
C      IN T(1) ETC. TO T(M).
C     WE WILL HAVE A DO LOOP TO WORK OUT THE PRESSURE CYCLE AT A
C      DISTANCE DIST.
C     WE ARE ASSUMING HEMISPHERICAL RADIATION. AS THE ENGINE IS VERY
C      NEAR TO THE GROUND IN RELATION TO THE DISTANCE WE ARE FROM IT, WE
C      MUST THEN INCREASE THE FAR FIELD SOUND PRESSURE, AS FOR SPHERICAL
C       RADIATION, BY A FACTOR OF 2.**(1./2.) TO ACCOUNT FOR THIS.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1          ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2          ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
C
C-----------------------------------------------------------------------
C
C     XXX IS A SIMPLIFYING EXPRESSION.
      XXX = 0.424264068711928 * AE / ( PI*DIST )
      MMM = NTIM - 1
      DO 1 N=1,MMM
C     BELOW WE CONSIDER ALL POINTS OTHER THAN LAST.
      PX(N) = XXX * ( UX(N+1) - UX(N) )/( TIMEX(N+1) - TIMEX(N) )
    1 CONTINUE
      PX(NTIM) = XXX * ( UX(1) - UX(NTIM) )
     1         / ( 60./RPMN - TIMEX(NTIM) + TIMEX(1) )
C     WE WILL ONLY PLOT OUT M - 1 VALUES OF RADIATED PRESSURE AS THE
C      LAST VALUE IS FOUND FROM THE FIRST AND LAST VELOCITY VALUES OF
C       THE SAME CYCLE.
C     IF WE ARE ON THE FIRST OR SECOND CYCLE, THE LAST PRESSURE VALUE
C      WILL OFTEN BE ODD, AS THE CYCLE OF VALUES USED IS NOT STEADY
C       STATE.
      J = NTIM - 1
      CALL QIKPLOT ( TIMEX,PX,-J,37H*PRESSURE CYCLE R METRES FROM OUTLET
     1*,35H*PRESSURE FLUCTUATIONS      PASCALS* )
      WRITE (76,3) DIST
      IF ( NNN.LT.NNZ ) GO TO 5
      CALL AUTPLT ( TIMEX,PX,J,1,37H*PRESSURE CYCLE R METRES FROM OUTLET
     1*,35H*PRESSURE FLUCTUATIONS      PASCALS* )
    5 CONTINUE
C     VELOCITY VALUES AT TAILPIPE OUTLET WILL ALSO BE PLOTTED.
C     WE WILL PLOT ALL VELOCITY VALUES.
```

A4

```
      END
```

```
      SUBROUTINE TRNPRS
C
C     SUBROUTINE CALCULATES THE PRESSURE VALUE AT THE POINT IN THE DUCT
C      TRDUC METRES FROM THE EXHAUST PORT, AT A PARTICULAR TIME. THIS IS
C      DONE BY CALCULATING THE PRESSURE VALUES BEHIND THE NEAREST
C      CHARACTERISTICS TO EITHER SIDE OF THE POINT AND THEN USING
C      LINEAR INTERPOLATION.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /I/ I,II,IJ
      COMMON /K/ KJ,KK
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2          ,TSTOP
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
      NA = 1
    1 IF ( ZI(NA).GE.TRDUCE ) GO TO 2
      IF ( NA.GT.I-1 ) GO TO 3
      NA = NA + 1
      GO TO 1
C     AT 2 WAVE NA IS FIRST WAVE TO RIGHT OF TRANSDUCER. WE MUST FIRST
C      SEE WHETHER THERE IS A WAVE NA-1.
    2 IF (NA.GT.1) GO TO 6
C     CASE BELOW IS FOR WAVE NA TO RIGHT OF TRANSDUCER, BUT NO WAVE TO
C      LEFT.
C     MUST FIND AVERAGE PRESSURE BETWEEN Q WAVE NA AND L.H. BOUNDARY.
      B = TRDUCE/ZI(NA) * ( A(NA) - A(II-1) ) + A(II-1)
      C = TRDUCE/ZI(NA) * ( SZI(NA) - SZI(II-1) ) + SZI(II-1)
      GO TO 7
C     CASE 3 IS FOR NO WAVE TO RIGHT OF TRANSDUCER. WE WILL ASSUME THAT
C      THERE IS A WAVE TO THE LEFT OF THE TRANSDUCER.
C     CASE IS FOR NO WAVE TO RIGHT AND WAVE TO LEFT OF TRANSDUCER, WE
C      MUST FIND AVERAGE PRESSURE BETWEEN THE WAVE AND R.H. BOUNDARY.
    3 B = ( TRDUCE - ZI(NA) )/( ZLN - ZI(NA) ) * ( A(II) - A(NA) )
     1  + A(NA)
      C = ( TRDUCE - ZI(NA) )/( ZLN - ZI(NA) ) * ( SZI(II) - SZI(NA) )
     1  + SZI(NA)
      GO TO 7
C     IN 6 WE MUST FIND AN AVERAGE VALUE BETWEEN THE TWO WAVES.
    6 B = ( TRDUCE - ZI(NA-1) )/( ZI(NA) - ZI(NA-1) )
     1  * ( A(NA) - A(NA-1) ) + A(NA-1)
      C = ( TRDUCE - ZI(NA-1) )/( ZI(NA) - ZI(NA-1) )
     1  * ( SZI(NA) - SZI(NA-1) ) + SZI(NA-1)
C          NOW CALCULATE PRESSURE IN PSI FROM SPEED OF SOUND AND
C      ENTROPY, IN METRIC UNITS.
    7 PE(KK) = PO * B**ZB * EXP( -GAMA*( C - SD ) ) / 6900.
      RETURN
```

23

```
      SUBROUTINE SPLCLC ( M,P,T,B )
C
      DIMENSION P(M),T(M),A(150),C(150)
C
C     SUBROUTINE CALCULATES THE RMS SOUND PRESSURE LEVEL HARMONICS FROM
C      A GIVEN CYCLE OF PRESSURE VALUES. THERE ARE M VALUES OF PRESSURE
C      P (AVERAGED ABOUT ZERO). CALCULATION GOES TO IJ HARMONICS AND
C      PRINTS OUT ANSWERS. THE TIME DIFFERENCE BETWEEN THE PRESSURE
C      VALUES NEED NOT BE CONSTANT BUT, OF COURSE, MUST BE SPECIFIED IN
C      AN ARRAY. THE OVERALL RMS LEVEL IS ALSO CALCULATED.
C
C-----------------------------------------------------------------------
C
      COMMON /I/ I,II,IJ
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1          ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
C
C-----------------------------------------------------------------------
C
      Y = 0.
      DO 1 MA=1,IJ
C     CALC. THE RMS SPL
      X = RMSHRM ( M,P,MA,B,T )
      Y = Y + X**2
      RM = 20.*ALOG10( X*B )
C     RMS SPL IS NOW PRINTED AS DB, WITH THE VALUE OF THE COEFF. MA
      A(MA) = RM
      C(MA) = FLOAT( MA )*RPMN/60.
      WRITE (76,990) MA,RM,C(MA)
    1 CONTINUE
C     Y IS THE SUM OF THE MEAN SQUARE PRESSURES FOR ALL THE SUB-BANDS.
C      WE MAY NOW FIND THE ROOT OF Y, WHICH WILL BE THE OVERALL RMS
C      PRESSURE.
      Y = SQRT( Y )
C     CALC. THE OVERALL RMS SPL.
      RM = 20.*ALOG10(Y*B)
      WRITE (76,5) RM
      CALL QIKPLOT ( C,A,-IJ,66H*FREQUENCY SPECTRUM OF ABOVE RMS SPL VAL
     1UES                      HZ*,15H*RMS   SPL    DB* )
      IF ( NNN.LT.NNZ ) GO TO 4
      CALL AUTPLT ( C,A,IJ,1,66H*FREQUENCY SPECTRUM OF ABOVE RMS SPL VAL
     1UES                      HZ*,15H*RMS   SPL    DB* )
    4 CONTINUE
      RETURN
C
C-----------------------------------------------------------------------
C
  990 FORMAT (10X,I3,F36.3,F27.1)
    5 FORMAT (20H  OVERALL RMS  SPL =,F7.2,3H DB)
      END
```

```
      AA = AA/FLOAT( NETA )
C     A IS NOW THE AVERAGE OF N VALUES OF P.
      DO 7 K=1,NETA
      PX(K) = PX(K) - AA
    7 CONTINUE
C     THE VALUES OF P ARE NOW ALL AVERAGED ABOUT ZERO. THAT IS, THEIR
C      SUM IS ZERO.
C     AS THE LAST PRESSURE VALUE IS NOT SENT TO AVERAG, IT IS NOT
C      EXPRESSED RELATIVE TO THE AVERAGE VALUE.
      WRITE (76,52)
      DO 3 J=1,NREV
      PE(J) = PE(J)/6900.
    3 CONTINUE
      CALL SPLCLC ( NREV,PE,TIME,344700000. )
C     SUBROUTINE CALCULATES AND PRINTS THE PEAK SPL IN DB. WE MUST FIND
C      THE MINIMUM AND MAXIMUM VALUES IN THE N VALUES OF ARRAY P.
C     WE NOW HAVE THE MINIMUM VALUE OF P IN RMIN.
C     WE NOW HAVE THE MAXIMUM VALUE OF P IN RMAX. WE WILL CALL THE PEAK
C      SPL RMAX (TO SAVE SPACE).
C     WE WILL CALCULATE THE PEAK SPL IN DB AND CALL IT RMAX.
      PMIN = 1.E10
      PMAX = -1.E10
      DO 15 K=1,NREV
      IF ( PE(K).LT.PMIN ) PMIN = PE(K)
      IF ( PE(K).GT.PMAX ) PMAX = PE(K)
   15 CONTINUE
C     WE NOW HAVE THE MINIMUM VALUE OF PE IN PMIN
C      AND THE MAXIMUM VALUE IN PMAX.
C     WE WILL CALCULATE THE PEAK SPL IN DB.
      PSPL = 20. * ALOG10( 172350000.*( PMAX - PMIN ) )
      WRITE (76,12) PSPL
      WRITE (76,53)
      CALL SPLCLC ( NETA,PX,TIMEX,50000. )
      RETURN
C
C-----------------------------------------------------------------------
C
   52 FORMAT (5X,18HNUMBER OF HARMONIC,15X,42HRMS SPL IN EXHAUST IN DB
     1 FREQUENCY IN HZ)
   12 FORMAT (20H  OVERALL PEAK SPL =,F7.2,3H DB)
   53 FORMAT (5X,18HNUMBER OF HARMONIC,12X,44HRMS SPL IN FAR FIELD IN DB
     1   FREQUENCY IN HZ)
      END
```

53

```
      SUBROUTINE RMSCLC
C
C     SUBROUTINE FINDS RMS VALUES OF SPL IN EXHAUST SYSTEM AT A CERTAIN
C      POINT, AND AT A CERTAIN POINT IN THE FAR FIELD (ASSUMING
C      HEMISPHERICAL PROPAGATION).
C     THE PRESSURE CYCLE IN THE EXHAUST, PRESSURE CYCLE IN THE FAR FIELD
C      AND THE VELOCITY AT THE TAILPIPE OUTLET ARE ALL PLOTTED. FOR THE
C     TWO PRESSURE CYCLES THE RMS SPL FOR L HARMONICS ARE PRINTED, AS IS
C      THE OVERALL RMS SPL. THE OVERALL PEAK SPL IS PRINTED FOR THE FAR
C      FIELD PRESSURE CYCLE.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /I/ I,II,IJ
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1           ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1           ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1           ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2           ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
C
C-----------------------------------------------------------------------
C
      DO 2 J=1,NREV
      PE(J) = PE(J)*6900.
    2 CONTINUE
C     SUBROUTINE FINDS THE AVERAGE OF THE N VALUES OF P, AND SUBTRACTS
C      THIS AVERAGE FROM EACH VALUE. THE N VALUES OF P ARE THEN
C      DESCRIBED AROUND AN AVERAGE OF ZERO.
      AA = 0.
      DO 1 K=1,NREV
      AA = AA + PE(K)
    1 CONTINUE
      AA = AA/FLOAT( NREV )
C     A IS NOW THE AVERAGE OF N VALUES OF P.
      DO 5 K=1,NREV
      PE(K) = PE(K) - AA
    5 CONTINUE
C     THE VALUES OF P ARE NOW ALL AVERAGED ABOUT ZERO. THAT IS, THEIR
C      SUM IS ZERO.
      CALL OIKPLOT ( TIME,PE,-NREV,46H*PRESSURE TRACE TAKEN AT TRANSDUCE
     1R     MSECS*,41H*PRESSURE ABOUT AVERAGE           PASCALS* )
      IF (NNN.LT.NNZ) GO TO 4
      CALL AUTPLT ( TIME,PE,NREV,1,46H*PRESSURE TRACE TAKEN AT TRANSDUCE
     1R     MSECS*,41H*PRESSURE ABOUT AVERAGE           PASCALS* )
    4 CONTINUE
      CALL VLDIFF
      NETA = NTIM - 1
      AA = 0.
      DO 6 K=1,NETA
      AA = AA + PX(K)
    6 CONTINUE
```

```
      SUBROUTINE REMOV1 ( NX, DUN )
C
      LOGICAL DUN
C
C     SUBROUTINE REMOVES ONE WAVE, WAVE NX-1, FROM SYSTEM BY NUMBERING
C      WAVES NX TO I AS NX-1 TO I-1. I IS THEN REDUCED BY 1.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /I/ I,II,IJ
      COMMON /J/ J,JA,JJ,JREV,JREX
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1          ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TROUC,TROUCE,TSTART
     2          ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ Z8,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
      DO 3 NYX=NX,I
      ZI(NYX-1) = ZI(NYX)
      U(NYX-1) = U(NYX)
      A(NYX-1) = A(NYX)
      PQ(NYX-1) = PQ(NYX)
      SZI(NYX-1) = SZI(NYX)
      SLPE(NYX-1) = SLPE(NYX)
      NTYPE(NYX-1) = NTYPE(NYX)
      TMEVNT(NYX-1) = TMEVNT(NYX)
      IF ( DUN ) GO TO 3
      ZIN(NYX-1) = ZIN(NYX)
      UN(NYX-1) = UN(NYX)
      AN(NYX-1) = AN(NYX)
      PQN(NYX-1) = PQN(NYX)
      SZIN(NYX-1) = SZIN(NYX)
      SLPEN(NYX-1) = SLPEN(NYX)
      NTYPEN(NYX-1) = NTYPEN(NYX)
      TMEVNN(NYX-1) = TMEVNN(NYX)
    3 CONTINUE
      I = I - 1
      JA = JA - 1
      RETURN
      END
```

113

```
      SUBROUTINE PRSTOP ( K )
C
C
C     SUBROUTINE PRINTS OUT WAVE DIAGRAM INFORMATION FROM MAIN PROGRAM
C      WHEN EITHER AN ERROR OR A RARE CASE (WHICH THE PROGRAM WILL NOT
C      HANDLE) HAS BEEN ENCOUNTERED. THIS PRINTOUT ASSISTS IN LOCATING
C      THE SOURCE OF THE ERROR. IT WILL THEN BE ARRANGED FOR EXECUTION
C      OF JOB TO BE STOPPED IN THIS SUBROUTINE.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /I/ I,II,IJ
      COMMON /M/ M,MM
      COMMON /N/ NC,ND,NI,NNN,NNZ,MREV,NTIM,NTYPE(100),NTYPEN(100)
     1           ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1           ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1           ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1           ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2           ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
      WRITE (76,1) K,M,NI,T
      IF ( T.LT.0.0 .OR. T.GT.0.2 ) GO TO 5
      WRITE (76,42)
      DO 44 K=1,I
      WRITE (76,43) A(K),U(K),SLPE(K),PQ(K),SZI(K),ZI(K),NTYPE(K),K
     1,TMEVNT(K)
   44 CONTINUE
    5 WRITE (76,4)
      IF ( T.LT.0.0 .OR. T.GT.0.2 ) GO TO 6
      WRITE (76,42)
      DO 45 K=1,I
      WRITE (76,43) AN(K),UN(K),SLPEN(K),PQN(K),SZIN(K),ZIN(K)
     1,NTYPEN(K),K,TMEVNN(K)
   45 CONTINUE
    6 STOP
C
C-----------------------------------------------------------------------
C
    1 FORMAT (5X,46HPROGRAM STOPS DUE TO AN ERROR OR RARE EVENT AT,I9//5
     1X,2HM=,I3,10X,3HNI=,I3,10X,2HT=,F12.10//10X,34HWAVE DIAGRAM FOR PR
     2ESENT TIME LINE)
   42 FORMAT (5X,6HA(NI)=,7X,6HU(NI)=,8X,9HSLPE(NI)=,8X,7HPQ(NI)=,8X,8HS
     1ZI(NI)=,8X,7HZI(NI)=,5X,10HNTYPE(NI)=,5X,11HNI=  TMEVNT)
   43 FORMAT (2X,F10.5,5F16.5,2I8,F12.7)
    4 FORMAT (//10X,31HWAVE DIAGRAM FOR NEXT TIME LINE)
      END
```

```
      SUBROUTINE PQADIF ( N )
C
C
C     SUBROUTINE CALCULATES THE CHANGE IN THE VALUE OF THE RIEMANN
C      INVARIANT (HELD IN PQ) FOR THE C+ AND C- CHARACTERISTICS AS THEY
C      TRAVERSE A VARIABLE AREA SECTION OF THE EXHAUST SYSTEM.
C     SUBROUTINE RESETS CLOCK ON PRESSURE WAVE TO ZERO AFTER CHANGING PQ
C      VALUE OF PRESSURE WAVE, BEFORE SOME EVENT IN THE WAVE DIAGRAM.
C     PROGRAM IS FOR EXPANSION CHAMBER WITH TWIN CONES SEPARATED BY
C      STRAIGHT SECTION, AND WITH STRAIGHT INLET AND OUTLET SECTIONS.
C     ALL UNITS ARE NON-DIMENSIONAL.
C     WE NEED TO FIND DNEW, THE DIAMETER AT THE CURRENT POSITION OF THE
C      WAVE AND DOLD, THE DIAMETER AT POSITION FOR WAVE FOR LAST TIME
C      PQADIF WAS CALLED.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2          ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ Z0,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
C     DISTANCE TRAVELLED WILL BE CALLED DST.
      DST = TMEVNT(N)/SLPE(N)
C     ZIOLD IS THE PSITION OF THE WAVE THS LAST TIME PQADIF WAS CALLED.
      ZIOLD = ZI(N) - DST
C     WE MUST NOW FIND THE DIAMETERS CORRESPONDING WITH ZI,ZIOLD.
      CALL WAREU ( ZI(N),DNEW )
      CALL WAREU ( ZIOLD,DOLD )
C     PROGRAM NOW HAS VALUES OF DNEW, DOLD.
      PQ(N) = PQ(N) - 2.*A(N)*U(N)*TMEVNT(N)*ALOG( DNEW/DOLD )/DST
      TMEVNT(N) = 0.
      RETURN
      END
```

```
      GO TO 43
   40 IF ( 1./PRATC.LE.PRATIC ) GO TO 46
C     CASE BELOW IS FOR SONIC OUTFLOW FROM CRANKCASE.
C     TLAMDA, TLAMIN FOR CYLINDER WILL BE FOUND
C     TLAMDC,TLAMIC FOR CRANKCASE WILL BE FOUND
      TLAMDA=-CTX*ACTC*ATC*PCTC*ACT**2/(VCT*PCT*ACTC**2)
      TLAMDC = CTX*ACTC*ATC/VCTC
      GO TO 47
C     CASE 46 IS FOR SUBSONIC OUTFLOW FROM CRANKCASE.
   46 CT = PRATC**(GAMAP/(2.*GAMA))
     1   * SQRT( 2./GAMAM * ( PRATC**( -GAMAM/GAMA) -1. ) )
      CT = CT*B80
      TLAMDA=-CT *ACTC*ATC*PCTC*ACT**2/(VCT*PCT*ACTC**2)
      TLAMDC = CT*ACTC*ATC/VCTC
   47 TLAMIN = TLAMIN + TLAMDA*TINT
      TLAMIC = TLAMIC + TLAMDC*TINT
C     PCTC,PRESSURE IN CRANKCASE WILL BE FOUND
      PCTC = PCOC * (VCOC/VCTC)**GAMA
     1     * EXP( -GAMA*TLAMIC - GAMAM*(SCCO - SCTC)/RGAS )
C     TCTC,TEMP.OF GAS IN CRANKCASE AT START
      TCTC = ACTC**2/(GAMA*RGAS)
C      ACTC,SPEED OF SOUND IN CRANKCASE WILL BE FOUND
      ACTC = ACOC * (PCTC/PCOC)**(1./ZB) * EXP(-(SCCO - SCTC)/(ZB*RGAS))
C     DLAMDA,ELAMDA ARE SIMPLIFYING EXPRESSIONS FOR TLAMDA
      DLAMDA =  - TLAMDA*TINT*(1. + TLAMDA*TINT)
      ELAMDA = 1. + TLAMDA*TINT
C     SCT,VALUE OF ENTROPY IN CYLINDER WILL BE UPDATED
      SCT = DLAMDA * ( SCTC - RGAS*ALOG( PRATC )
     1                 + CP*ALOG( DLAMDA + ELAMDA*TCT/TCTC ) )
     2    + ELAMDA * ( SCT + CP*ALOG( DLAMDA*TCTC/TCT + ELAMDA ) )
C     PCT,PRESSURE IN CYLINDER WILL BE UPDATED
      PCT = PCO * (VCO/VCT)**GAMA
     1    * EXP( -GAMA*TLAMIN - GAMAM*(SCO - SCT)/RGAS )
C     ACT WILL BE UPDATED
      ACT = ACO * (PCT/PCO)**(1./ZB) * EXP( -(SCO - SCT)/(ZB*RGAS) )
C     TCT,TEMP.IN CYLINDER WILL BE UPDATED
      TCT = ACT**2/(GAMA*RGAS)
   43 CONTINUE
      RETURN
      END
```

```
C     PCT,PRESS IN CYLINDER AT T, WILL BE CALCULATED
      PCT = PCJ * (VCO/VCT)**GAMA
     1   * EXP( -GAMA*TLAMIN - GAMAM*(SCO - SCT)/RGAS )
      ACT = ACO * (PCT/PCJ)**(1./Z8) * EXP( -(SCO - SCT)/(Z8*RGAS))
      TCT = ACT**2/(GAMA*RGAS)
      GO TO 132
C     CASE 1 IS FOR INFLOW TO CYLINDER
    1 IF ( PRATIC.LT.PRATIC ) GO TO 504
C     CASE BELOW IS FOR SONIC INFLOW TO CYLINDER
      TLAMDA =  - BBQ*CTX*ACT*AT/VCT
      GO TO 2
C     CASE 504 IS FOR SUBSONIC INFLOW TO CYLINDER, NON - ISENTROPIC
  504 CT = -PRATIO**( -GAMAP/(2.*GAMA) )
     1   * SQRT( 2./GAMAM * ( PRATIO**(GAMAM/GAMA) - 1. ) )
      GO TO 3
  132 CONTINUE
C     MUST DETERMINE WHETHER TRANSFER PORTS OPEN
      IF ( ABS( 6.*RPMN*T -X ) .GT. XC ) GO TO 43
C     CASE BELOW IS FOR TRANSFER PORTS OPEN
      PRATC = PCT/PCTC
      IF ( PRATC.LE.1. ) GO TO 40
      IF ( PRATC.LE.PRATIC ) GO TO 42
C     CASE BELOW IS FOR SONIC INFLOW TO CRANKCASE.
C     TLAMDA,TLAMIN FOR CYLINDER WILL BE FOUND
C     TLAMDC,TLAMIC FOR CRANKCASE WILL FOUND
      TLAMDA = CTX*ACT*ATC/VCT
      TLAMDC =-CTX*ACT*ATC*PCT*ACTC**2/(VCTC*PCTC*ACT**2)
      GO TO 44
C     CASE 42 IS FOR SUBSONIC INFLOW TO CRANKCASE.
   42 CT = PRATC**( -GAMAP/(2.*GAMA) )
     1   * SQRT( 2./GAMAM * ( PRATC**(GAMAM/GAMA) -1. ) )
      CT = CT*BBQ
      TLAMDA = CT*ACT*ATC/VCT
      TLAMDC =-CT *ACT*ATC*PCT*ACTC**2/(VCTC*PCTC*ACT**2)
   44 TLAMIN = TLAMIN + TLAMDA*TINT
      TLAMIC = TLAMIC + TLAMDC*TINT
C     PCT,PRESSURE IN CYLINDER WILL BE FOUND
      PCT = PCJ * (VCO/VCT)**GAMA
     1   * EXP( -GAMA*TLAMIN - GAMAM*(SCO - SCT)/RGAS )
C     TCT,TEMP.OF GAS IN CYLINDER AT START
      TCT = ACT**2/(GAMA*RGAS)
C     ACT,SPEED OF SOUND IN CYLINDER WILL BE FOUND
      ACT = ACO * (PCT/PCJ)**(1./Z8) * EXP( -(SCO - SCT)/(Z8*RGAS) )
C     DLAMDC,ELAMDC ARE SIMPLIFYING EXPRESSIONS FOR TLAMDC
      DLAMDC =  - TLAMDC*TINT*(1. + TLAMDC*TINT)
      ELAMDC = 1. + TLAMDC*TINT
C     SCTC,VALUE OF ENTROPY IN CRANKCASE WILL BE UPDATED
      SCTC = DLAMDC * ( SCT - RGAS*ALOG( 1./PRATC )
     1                  + CP * ALOG( DLAMDC + ELAMDC*TCTC/TCT ) )
     2     + ELAMDC * ( SCTC + CP * ALOG( DLAMDC*TCT/TCTC + ELAMDC ) )
C     PCTC,PRESS.IN CRANKCASE WILL BE UPDATED
      PCTC = PCOC * (VCOC/VCTC)**GAMA
     1     * EXP( -GAMA*TLAMIC - GAMAM*(SCCO - SCTC)/RGAS )
C     ACTC WILL BE UPDATED
      ACTC = ACOC * (PCTC/PCOC)**(1./Z8) * EXP(-(SCCO - SCTC)/(Z8*RGAS))
C     TCTC TEMP.IN CRANKCASE WILL BE UPDATED
      TCTC = ACTC**2/(GAMA*RGAS)
```

```
      SUBROUTINE MASSEX
C
C
C     SUBROUTINE PERFORMS MASS EXCHANGE CALCULATIONS BETWEEN CRANKCASE,
C     CYLINDER AND EXHAUST PIPE. NEW VALUES OF CONDITIONS IN CYLINDER
C      AND CRANKCASE ARE CALCULATED. ALL FLOWS BETWEEN THE CYLINDER AND
C      THE EXHAUST PIPE ARE ASSUMED TO BE THOSE CALCULATED IN FLOWIN AND
C      FLWOUT. THE VALUES OF THE CYLINDER CONDITIONS ARE NOT CALCULATED
C      PRECISELY FOR FLOWS FROM THE EXHAUST PIPE INTO THE CYLINDER.
C     WILL FIRST PERFORM MASS TRANSFER CALCULATIONS FOR FLOW BETWEEN
C      CYLINDER AND EXHAUST PORT.
C     WILL DETERMINE WHETHER CYLINDER INFLOW OR OUTFLOW EXISTS.
C
C----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /B/ B,BBQ,BC,BORE,BOREA
      COMMON /C/ CONLEN,CP,CTX
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1         ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1         ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1         ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2         ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /V/ VCT,VCTC,VCTCO,VCO,VCOC,VS,VSC
      COMMON /X/ X,XC,XCI,XE,XEC,XL1,XL3,XZ2,XZ4,X1,X2,X3,X4
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C----------------------------------------------------------------------
C
      IF ( UDLAST.LT.0. ) GO TO 1
C     WILL CONSIDER CYLINDER OUTFLOW FIRST
C     MUST DETERMINE WHETHER SONIC OR SUBSONIC
      IF ( PRATIO.LT.PRATIC ) GO TO 503
C     CASE BELOW IS FOR SONIC OUTFLOW FROM CYLINDER.
C     AN EXPRESSION FOR TLAMDA WILL BE FOUND
      TLAMDA = BBQ*CTX*ACT*AT/VCT
C     THE INTEGRAL OF TLAMDA*TIME WILL BE FOUND
    2 TLAMIN = TLAMIN + TLAMDA*TINT
C     PCT, PRESS IN CYLINDER AT T, WILL BE CALCULATED.
      PCT = PCJ * (VCO/VCT)**GAMA
     1    * EXP( -GAMA*TLAMIN - GAMAM*(SCO - SCT)/RGAS )
      ACT = ACO * (PCT/PCJ)**(1./ZB) * EXP( -(SCO - SCT)/(ZB*RGAS))
      TCT = ACT**2/(GAMA*RGAS)
      GO TO 132
C     CASE 503 IS FOR SUBSONIC OUTFLOW FROM CYLINDER, NON-ISENTROPIC.
  503 CT = PRATIO**( -GAMAP/(2.*GAMA) )
     1   * SQRT( 2./GAMAM * ( PRATIO**( GAMAM/GAMA ) - 1. ) )
    3 CT = CT*BBQ
C     AN EXPRESSION FOR TLAMDA WILL BE FOUND
      TLAMDA = CT*ACT*AT/VCT
C     THE INTEGRAL OF TLAMDA*TIME WILL BE FOUND
      TLAMIN = TLAMIN + TLAMDA*TINT
```

```
C       BE SWEPT D/S
   43 CONTINUE
C     WILL ENSURE THAT AA2 IS WITHIN POSSIBLE RANGE
      KKK = 1
      YI = SQRT(GAMAB)*AACT + GAMAB*QR
      YB = AMIN1(YI,XA)
      YI = YB/GAMAB - QR
      UU2MAX = AMIN1( AACT/SQRT( GAMAB ),YI,UMAX )
      UU2MIN = 0.
      UU2LST = UU2MAX - 1.E-10
      AA2 = GAMAB*QR
      YA = AMAX1( YA,AA2 )
      AA2 = YA + 1.E-10
      AA2LST = AA2 + 0.01
      IF ( AA2LST.LE.YA .OR. AA2LST.GE.YB ) AA2LST = (YA + YB)/2.
C     UU2MAX IS MAXIMUM VALUE OF UU2
C     UU2MIN IS MINIMUM VALUE OF UU2
C     YB IS MAXIMUM VALUE OF AA2
C     YA IS MINIMUM VALUE OF AA2
   17 UU2 = AA2/GAMAB - QR
      IF ( UU2.GT.UU2MAX ) UU2 = UU2MAX - 1.E-10
      IF ( UU2.LE.UU2MIN ) UU2 = UU2MIN + 1.E-10
      YC = AACT**2 - GAMAB*UU2**2
      UU2 = YC*Z*CTX*AACT**(GAMAP/GAMAM)/(ARATIO*AA2**(GAMA/GAMAB))
      IF ( UU2.GE.UU2MAX ) UU2 = UU2MAX - 1.E-10
      IF ( UU2.LE.UU2MIN ) UU2 = UU2MIN + 1.E-10
      UU2 = (UU2LST + UU2)/2.
      DIFUU2 = ABS( UU2 - UU2LST )
      UU2LST = UU2
      AA2 = GAMAB * ( QR + UU2 )
      IF ( AA2.LE.YA ) AA2 = YA + 1.E-10
      IF ( AA2.GE.YB ) AA2 = YB - 1.E-10
      AA2 = (AA2 + AA2LST*(1.-FLOAT(KKK)**(-10)))/(2.-FLOAT(KKK)**(-10))
      DIFAA2 = ABS( AA2 - AA2LST )
      AA2LST = AA2
      IF ( DIFUU2.LT.1.E-4 .AND. DIFAA2.LT.1.E-4 ) KL = KKK
      KKK = KKK + 1
      IF ( KKK.LT.KL ) GO TO 17
      KL = KJ
    3 UU2 = AA2/GAMAB - QR
C     CORRECT VALUE OF AA2 HAS BEEN FOUND, OTHER FLOW VALUES MUST BE
C       FOUND.
C     UU2, VEL. IN OLD FLOW GAS, IS EQUAL TO UU1
      UU1 = UU2
C     AA1 IS SPEED OF SOUND IN NEW FLOW
      AA1 = SQRT(AACT**2 - GAMAB*UU1**2)
C     S1 IS ENTROPY OF NEW FLOW, WILL MAKE NEW SDT EQUAL TO S1
      S1 = SDT + ALOG(AA1/AA2)/GAMAB
C     P1 IS STATIC PRESSURE IN NEW FLOW, AND AT PORT, NEW PDT IS P1
      PDT=PCT*(AA2/AACT)**(GAMA/GAMAB)*EXP(-GAMA*(SDT-SCT/(GAMA*RGAS)))
C     NEW SDT IS EQUAL TO CURRENT S1
      PR = AA2/GAMAB + UU2
   31 CONTINUE
      QR = PR
      SDT = S1
      RETURN
      END
```

D3

```
      IF ( UU2.LE.UU2MIN ) UU2 = UU2MIN + 1.E-10
      UU2 = ( UU2LST + UU2 )/2.
      DIFUU2 = ABS( UU2 - UU2LST )
      UU2LST = UU2
      AA2 = GAMAB * ( QR + UU2 )
      IF ( AA2.LE.YA ) AA2 = YA + 1.E-10
      IF ( AA2.GE.YB ) AA2 = YB - 1.E-10
      AA2 =( AA2 + AA2LST * (1.-FLOAT(K)**(-10)) )/(2.-FLOAT(K)**(-10))
      DIFAA2 = ABS( AA2 - AA2LST )
      AA2LST = AA2
      IF( DIFAA2.LT.1.E-4 .AND. DIFUU2.LT.1.E-4 .AND. K.GT.2 ) KL = K
      K = K + 1
      IF ( K.LT.KL ) GO TO 7
      KL = KJ
      GO TO 3
C     CASE 4 IS FOR SONIC OUTFLOW, AS IN DIFFUSER
    4 FM1 = 1.1
      FM1LST = 1.
C     FM1 IS FLOW MACH NO. JUST UPSTREAM OF SHOCK WHEN SHOCK IS LOCATED
C      JUST BEFORE THE FULL DUCT AREA, AREAT. THIS TYPE OF FLOW
C      DETERMINES THE MIN. QR
      KA = 1
   11 IF ( FM1.LE.1. ) FM1 = 1.0000000001
      FM1 = SQRT (
     1    GAMAP * (FM1*ARATIO)**(2.*GAMAM/GAMAP) / GAMAM - 1./GAMAB )
      DIFFM1 = ABS( FM1 - FM1LST )
      FM1LST = FM1
      IF ( DIFFM1.LT.1.E-4 ) KL = KA
      KA = KA + 1
      IF ( KA.LT.KL ) GO TO 11
      KL = KJ
C     VALUE OF FM1 HAS BEEN FOUND MUST FIND CORRESPONDING VALUE OF QR
C     FM11 IS FLOW MACH NO. JUST D/S OF SHOCK
      FM11 = SQRT((1. + GAMAB*FM1**2)/(GAMA*FM1**2 - GAMAB))
C     AA1 IS STATIC SPEED OF SOUND D/S OF SHOCK IN NEW FLOW
      AA1 = AACT/SQRT(GAMAB*FM11**2 + 1.)
C     UU1 IS GAS VEL. IN NEW FLOW FOR LIMITING CASE
      UU1 = FM11*AA1
      UU2 = UU1
      UMAX = UU2
C     STERM IS ENTROPY RISE ACROSS SHOCK , FOR LIMITING FLOW CASE
      STERM = ALOG((1. + 2. GAMA*(FM1**2 - 1.)/GAMAP)**(1./GAMAM)
     1 *(GAMAP*FM1**2/(GAMAM*FM1**2 + 2.))**(-GAMA/GAMAM))/GAMA
C     S1 IS ENTROPY OF NEW FLOW, FOR LIMITING CASE
      S1 = STERM + SCT/(GAMA*RGAS)
C     AA2 IS STATIC SPEED OF SOUND IN OLD FLOW GAS, FOR LIMITING CASE
      AA2 = AA1*EXP(GAMAB*(SOT - S1))
      IF (UU2.GT.AA2 .AND. ARATIO.LT.500.) STOP 1001
C     PROGRAM STOPS AS EXTREME FLOW CASE HAS SONIC FLOW IN OLD FLOW GAS
C      BEFORE SHOCK IS SWEPT DOWNSTREAM, AND AREA RATIO IS NOT HIGH
C      ENOUGH TO BE CONSIDERED VERY HIGH.
      YA = AA2
      IF (UU2.GT.AA2) GO TO 43
C     QRLOW IS LOWEST POSSIBLE VALUE OF QR FOR SHOCK TO STAY NEAR PORT
      QRLOW = AA2/GAMAB - UU2
      IF (QR.LE.QRLOW) GO TO 31
C     PROGRAM STOPS AS QR VALUE IS LOW ENOUGH TO CAUSE SHOCK AT PORT TO
```

```
      IF ( AA2.GE.YB .OR. AA2.LE.AA2MIN ) AA2 = YB - 1.E-10
C     YI IS MAXIMUM VALUE OF AA1
C     AA2MIN IS MINIMUM VALUE OF AA2
C     YB IS MAXIMUM VALUE OF AA2
      AA1LST = AA1
      AA2LST = AA2  - 0.01
      IF ( AA2LST.GE.YB .OR. AA2LST.LE.AA2MIN ) AA2LST =(YB + AA2MIN)/2.
   39 AA2 = ( AA1**2/ARATIO * SQRT( X * ( X*AACT**2 - AA2**2)/GAMAB ) )
     1      **ONETHD
      IF ( AA2.GE.YB ) AA2 = YB - 1.E-10
      IF ( AA2.LE.AA2MIN ) AA2 = AA2MIN + 1.E-10
      AA2 = (AA2 + AA2LST)/2.
      DIFAA2 = ABS( AA2 - AA2LST )
      AA2LST = AA2
      AA1 = SQRT( AACT**2 - GAMAB*AA2**2 )
      IF ( AA1.GE.YI ) AA1 = YI - 1.E-10
      IF ( AA1.LE.AA2 ) AA1 = AA2 + 1.E-10
      AA1 = ( AA1 + AA1LST )/2.
      DIFAA1 = ABS( AA1 - AA1LST )
      AA1LST = AA1
      IF ( DIFAA1.LT.1.E-4 .AND. DIFAA2.LT.1.E-4 ) KL = KB
      KB = KB + 1
      IF ( KB.LT.KL ) GO TO 39
      KL = KJ
      UU2 = AA2
      YA = AA2
      UMAX = UU2
      IF ( QR.LE.AA2/GAMAB - UU2 ) GO TO 31
C     CASE 5 IS FOR SUBSONIC OUTFLOW, AS FROM SQUARE EDGED OUTLET TO
C      PIPE.
    5 CONTINUE
C     WILL ITERATE TO GET AA2, THE NONDIMENSIONAL SPEED OF SOUND IN THE
C     PREVIOUS FLOW GAS. SIMPLIFYING EXPRESSIONS WILL BE USED.
C     WILL ENSURE THAT AA2 IS WITHIN POSSIBLE RANGE.
      K = 1
      YI = AACT*SQRT( X )
      YB = AMIN1( YI,XA )
      AA2 = GAMAB*QR
      YA = AMAX1( YA,AA2 )
      AA2 = YA + 1.E-10
      AA2LST = AA2 + 0.01
      IF ( AA2LST.LE.YA .OR. AA2LST.GE.YB ) AA2LST = (YA + YB)/2.
      YI = YB/GAMAB - QR
      UU2MAX = AMIN1( AACT/SQRT( GAMAB ),YI,UMAX )
      UU2MIN = 0.
      UU2LST = UU2MAX - 1.E-10
C     UU2MAX IS MAXIMUM VALUE OF UU2
C     UU2MIN IS MINIMUM VALUE OF UU2
C     YB IS MAXIMUM VALUE OF AA2
C     YA IS MINIMUM VALUE OF AA2
    7 UU2 = AA2/GAMAB - QR
      IF ( UU2.GT.UU2MAX ) UU2 = UU2MAX - 1.E-10
      IF ( UU2.LE.UU2MIN ) UU2 = UU2MIN + 1.E-10
      YD = ( AACT**2*X - AA2**2 )*X
      YC = AACT**2 - GAMAB*UU2**2
      UU2 = YC/( ARATIO*AA2**2 ) * SQRT( YD/GAMAB )
      IF ( UU2.GE.UU2MAX ) UU2 = UU2MAX - 1.E-10
```

```
      SUBROUTINE FLWOUT ( AA2,QR,UU2,AA1 )
C
C
C     SUBROUTINE PERFORMS THE CALCULATIONS TO CORRECTLY ALLOW FOR A C-
C      CHARACTERISTIC, WHICH REACHES THE OPEN EXHAUST PORT, TO BE
C      REFLECTED AS A C+ CHARACTERISTIC, WHEN THE RESULTANT FLOW IS OUT
C      OF THE CYLINDER. ZERO FLOW IS ALSO ALLOWED.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /C/ CONLEN,CP,CTX
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /K/ KJ,KK
      COMMON /O/ ONETHD
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
C
C-----------------------------------------------------------------------
C
      KL = KJ
      TMP = SDT - SCT/(GAMA*RGAS)
      X = EXP( GAMAM*TMP )
      Z = EXP( GAMA*TMP )
      ARATIO = AD/AT
      AACT = ACT/AO
      XA = GAMAM*QR/(3. - GAMA)
      IF( QR - AL/GAMAB ) 2,1,6
C     PROGRAM STOPS AS QR IS SUCH AS TO CAUSE INFLOW.
    6 STOP 1000
C     CASE 1 WILL CONSIDER NO FLOW
    1 UU2 = 0.
      AA2 = GAMAB*QR
      PR = QR
      GO TO 31
C     CASE 2 WILL CONSIDER CYLINDER OUTFLOW
    2 TERM = 2. * ( ARATIO/GAMAM )**2 / GAMAP
      UU2 = AACT * ( SQRT( 1./GAMAB + TERM ) - SQRT( TERM ) )
      UMAX = UU2
      AA2 = AACT * SQRT( 2./GAMAP ) * EXP( GAMAB*TMP )
      YA = AA2
      IF ( UU2.GT.AA2 ) GO TO 35
      IF ( QR .LE. AA2/GAMAB - UU2 ) 4,5
C     35 IS LIMITING FLOW CASE FOR SONIC FLOW AT 2 BUT SUBSONIC FLOW IN
C      THROAT.
   35 KB = 1
      YE = AACT/SQRT( GAMAB )
      YI = AACT*SQRT( X )
      XB = 2. * AL / (3. - GAMA)
      YB = AMIN1( YE,YI,XB )
      AA2MIN = AACT * X / SQRT( X + GAMAB*ARATIO**2 )
      YI = AACT
      AA1 = YI - 1.E-10
      AA2 = ( AA1**2 / ARATIO )**ONETHD
```

```
      URMAX = 0.
      ARMAX = GAMAB*QR
      ARMIN = AL
      AR = ARMAX - 1.E-10
      ARLAST = AR - 0.01
      IF (ARLAST.LT.ARMIN .OR. ARLAST.GT.ARMAX) ARLAST =(ARMAX+ARMIN)/2.
      URLAST = URMIN + 1.E-10
C     URMAX IS MAXIMUM ALLOWABLE VALUE OF UR
C     URMIN IS MINIMUM ALLOWABLE VALUE OF UR
C     ARMAX IS MAXIMUM ALLOWABLE VALUE OF AR
C     ARMIN IS MINIMUM ALLOWABLE VALUE OF AR
    1 IF ( AR.LT.ARMIN ) AR = ARMIN + 1.E-10
      IF ( AR.GT.ARMAX ) AR = ARMAX - 1.E-10
      AR = ( AR + ARLAST * ( 1. - FLOAT(K)**(-10) ) )
     1   / ( 2. - FLOAT(K)**(-10) )
      DIFFAR = ABS( AR - ARLAST )
      ARLAST = AR
      UR =  - SQRT(  (AR**2 - AL**2)
     1            / (GAMAB*( ARATIO**2 * (AR/AL)**(4./GAMAM) - 1.) ) )
      IF ( UR.LT.URMIN ) UR = URMIN + 1.E-10
      IF ( UR.GT.URMAX ) UR = URMAX - 1.E-10
      UR =( UR + URLAST * (1. -FLOAT(K)**(-10)))/(2. -FLOAT(K)**(-10))
      DIFFUR = ABS( UR - URLAST )
      URLAST = UR
      AR = GAMAB * ( QR + UR )
      PR = QR + 2.*UR
      IF ( DIFFUR.LT.1.E-4 .AND. DIFFAR.LT.1.E-4 ) KL = K
      K = K + 1
      IF ( K.LE.KL ) GO TO 1
      KL = KJ
C     STOP 3000 IS FOR SONIC FLOW AT START OF DUCT, AT FULL DUCT AREA
      IF ( -UR.GE.AR) STOP 3000
      GO TO 25
   15 UR =  - QR/(1. + 1./(FMR*GAMAB))
      AR =  - UR/FMR
      PR = QR + 2.*UR
      AL = AR**(2./GAMAP) * ( - ARATIO*UR )**(GAMAM/GAMAP)
   25 CONTINUE
      PDT = PD*AR**(GAMA/GAMAB)*EXP( - GAMA*(SDT - SD))
      QR = PR
      RETURN
      END
```

```
      SUBROUTINE FLOWIN ( AR,QR,UR )
C
C
C     SUBROUTINE PERFORMS THE CALCULATIONS TO CORRECTLY ALLOW FOR A C-
C      CHARACTERISTIC, WHICH REACHES THE OPEN EXHAUST PORT, TO BE
C      REFLECTED AS A C+ CHARACTERISTIC. WHEN THE RESULTANT FLOW AT THE
C      EXHAUST PORT IS INTO THE CYLINDER FROM THE EXHAUST PIPE.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /C/ CONLEN,CP,CTX
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /K/ KJ,KK
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(300),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMCN /S/ SCCO,SCT,SCTC,SCO,SD,SOT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
C
C-----------------------------------------------------------------------
C
      KL = KJ
      PDT = AL/GAMAB
      ARATIO = AD/AT
C     MUST DETERMINE WHETHER INFLOW IS SONIC.
C     MUST FIRST MAKE AN ESTIMATE FOR FMR, FLOW MACH NO. AT LARGE AREA,
C      IF INFLOW IS SONIC.
      KKK = 1
      IF ( ARATIO.LT.2. ) GO TO 9
C     CASE 7 IS FOR SHORT ITERATION
      FMR = CTX/ARATIO
      FMRLST = FMR - 0.01
   12 FMR = CTX * ( 1. + GAMAB * FMR**2 )**( 0.5*GAMAP/GAMAM ) / ARATIO
      DIFFMR = ABS( FMR - FMRLST )
      FMRLST = FMR
      IF ( DIFFMR.LT.1.E-4 ) KL = KKK
      KKK = KKK + 1
      IF ( KKK.GT.KL ) 11,12
C     CASE 9 IS FOR LONG ITERATION
    9 FMR =  -2.* SQRT( ( ARATIO/(CTX*GAMAP) )**2 - 1./GAMAP)
      FMRLST = FMR - 0.01
      FMR = FMR + 2.*ARATIO/(CTX*GAMAP)
   14 FMR = CTX * ( 1. + GAMAB * FMR**2 )**( .5*GAMAP/GAMAM )/ARATIO
      DIFFMR = ABS( FMR - FMRLST )
      FMRLST = FMR
      IF ( DIFFMR.LT.1.E-4 ) KL = KKK
      KKK = KKK + 1
      IF ( KKK.LT.KL ) GO TO 14
   11 UR =  - AL * FMR**(2./GAMAP) / ARATIO**(GAMAM/GAMAP)
      KL = KJ
      AR =  - UR/FMR
      URMIN = UR
      PDT = AR/GAMAB - UR
      IF ( QR.GT.PDT ) GO TO 15
      K = 1
      YI = AL/GAMAB - QR
      URMIN = AMAX1( YI,URMIN )
```

```
      SUBROUTINE AREAVL
C
C
C     SUBROUTINE CALCULATES THE PRECISE EXHAUST AND TRANSFER PORT AREAS,
C      CYLINDER AND CRANKCASE VOLUMES AT A PARTICULAR TIME.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /B/ B,BBQ,BC,BORE,BOREA
      COMMON /C/ CONLEN,CP,CTX
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /Q/ Q
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2          ,TSTOP
      COMMON /V/ VCT,VCTC,VCTCO,VCO,VCOC,VS,VSC
      COMMON /X/ X,XC,XCI,XE,XEC,XL1,XL3,XZ2,XZ4,X1,X2,X3,X4
C
C-----------------------------------------------------------------------
C
      THETA = PI*T*RPMN/30. - XE
      AASIN = ( Q * SIN( THETA ) )**2 / 2.
      ABSIN = ( Q * SIN( XE ) )**2 / 2.
      ACSIN = ( Q * SIN( XEC ) )**2 / 2.
      FITERM =  AASIN*(1.+ AASIN*(1.+ AASIN*(1.+ 5.*AASIN/4.))/2.)
      FITRTH = -ABSIN*(1.+ ABSIN*(1.+ ABSIN*(1.+ 5.*ABSIN/4.))/2.) + 1.
      FITRTR = -ACSIN*(1.+ ACSIN*(1.+ ACSIN*(1.+ 5.*ACSIN/4.))/2.) + 1.
C     AT, AREA OF PORT AT TIME T, WILL BE FOUND.
      CSTHE = COS( THETA )
      CSXE = COS( XE )
      AT = B * ( CSTHE - CSXE + ( FITRTH - 1. + FITERM )/Q )
     1   / (1. - CSXE + ( FITRTH - 1. )/Q )
C     VCT WILL NOW BE CALCULATED
      TMP = CONLEN*BOREA*FITERM
      VCT = VS*( 1./( R - 1. ) + 0.5 + 0.5*CSTHE ) + TMP
      WRITE (76,1) T
      IF ( ABS( T*6.*RPMN - X ).GT.XC ) RETURN
      VCTC = VSC*( 1./( RC - 1. ) + 0.5 - 0.5*CSTHE ) - TMP
      CSXE = COS( XEC )
      ATC = BC*( CSTHE - CSXE + ( FITRTR - 1. + FITERM )/Q )
     1    / ( 1. - CSXE + ( FITRTR - 1. )/Q )
      RETURN
C
C-----------------------------------------------------------------------
C
    1 FORMAT(//5X,2HT=,F12.10)
      END
```

I-2

```
      SUBROUTINE ADD1LH
C
C
C     SUBROUTINE PREPARES THE PROGRAM FOR THE ADDITION OF AN EXTRA P
C      CHARACTERISTIC BY RENUMBERING ALL THE EXISTING CHARACTERISTICS
C      1 TO I AS 2 TO I+1. I IS THEN INCREASED BY 1. THE NEW P
C      CHARACTERISTIC IS TO BE ADDED AT THE LEFT HAND BOUNDARY, THAT
C      IS, AT THE EXHAUST PORT.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /I/ I,II,IJ
      COMMON /J/ J,JA,JJ,JREV,JREX
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1           ,NWAVES,NWVDIS
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1           ,PRATIC,PRATIO,PX(1200),PO
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1           ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1           ,TLAMIN,TMEVNN(100),TMEVNT(100),TROUC,TROUCE,TSTART
     2           ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
      N = I
    1 ZIN(N+1) = ZIN(N)
      UN(N+1) = UN(N)
      AN(N+1) = AN(N)
      SLPEN(N+1) = SLPEN(N)
      PQN(N+1) = PQN(N)
      SZIN(N+1) = SZIN(N)
      NTYPEN(N+1) = NTYPEN(N)
      TMEVNN(N+1) = TMEVNN(N)
      ZI(N+1) = ZI(N)
      U(N+1) = U(N)
      A(N+1) = A(N)
      SLPE(N+1) = SLPE(N)
      PQ(N+1) = PQ(N)
      SZI(N+1) = SZI(N)
      NTYPE(N+1) = NTYPE(N)
      TMEVNT(N+1) = TMEVNT(N)
      N = N - 1
      IF ( N.GT.0 ) GO TO 1
      I = I + 1
      JA = JA + 1
      RETURN
      END
```

```
      349 FORMAT (95H  P WAVE NI.LE.I-2, P WAVE NI+1.LE.I-1. WAVE NI DOES NO
     1T REACH R.H. END BUT OVERTAKES WAVE NI+1)
      315 FORMAT(10X,67HA P WAVE NI-1, OR NI HAS CAUGHT UP ANOTHER P WAVE NI
     1, OR NI+1 AT M=,I3,3HNI=,I3,5X,2HI=,I3)
      329 FORMAT (94H  TWO RIGHTWARD WAVES, AND AT LEAST ONE MORE WAVE, REAC
     1H R.H. BOUNDARY IN FINE TIME LINE AT M=,I3,5X,3HNI=,I3)
      343 FORMAT (97H  RW WAVE NI MEETS LW WAVE NI+1, RESULTANT WAVE NI PUT
     1AT L.H. BOUNDARY BUT THERE IS WAVE NI-1 M=,I3,5X,3HNI=,I3)
       42 FORMAT (5X,6HA(NI)=,7X,6HU(NI)=,8X,9HSLPE(NI)=,8X,7HPQ(NI)=,8X,8HS
     1ZI(NI)=,8X,7HZI(NI)=,5X,10HNTYPE(NI)=,5X,11HNI=   TMEVNT)
       43 FORMAT (2X,F10.5,5F16.5,2I8,F12.7)
        5 FORMAT (5X,7HPE(KK)=,F8.4,5X,4HPCT=,F8.4,5X,5HPCTC=,F8.4,5X,4HPDT=
     1,F8.4,15X,2HT=,F10.8)
          END
```

- H2

```
            CALL TRNPRS
            TIME(KK) = T
      C
      C
      C     SUBROUTINE PRINTS OUT VALUES OF PRESSURES AT TRANSDUCER (PE(KK)),
      C      IN CYLINDER (PCT), IN CRANKCASE (PCTC), AT PORT (PDT), AND TIME
      C      AFTER EPO (T).
      C
      C-----------------------------------------------------------------------
      C
            AA = PCT/SXN
            BB = PCTC/SXN
            C = PDT/SXN
            WRITE (76,5) PE(KK),AA,BB-C,T
            GO TO 120
        998 CALL RMSCLC
      C     WE MUST PLOT OUT ENTROPY AND POSITION OF WAVEFRONTS IN PIPE.
            CALL QIKPLOT(ZI,SZI,-I,27H*ENTROPY VALUES ALONG PIPE*,9H*ENTROPY*)
            IF ( NNN.GE.J ) GO TO 999
      C     WE MUST RESET INITIAL CYLINDER AND CRANKCASE CONDITIONS FOR NEXT
      C      REV.
            KK = 0
            NC = 0
            NTIM = 0
            ACT = ACO
            TLAMIN = 0.
            PCT = PCO*SXN
            TLAMIC = 0.
            TCT = TCO
            T = DIT
            VCTC = VCTCO
            ATC = 1.E-10
            PCTC = PCOC * ( VCOC/VCTCO )**GAMA
            SCT = SCO
            SCTC = SCCO
            TCTC = TCOC * ( VCOC/VCTCO )**GAMAM
            ACTC = SQRT( GAMA*RGAS*TCTC )
            GO TO 120
        999 CONTINUE
            RETURN
      C
      C-----------------------------------------------------------------------
      C
        345 FORMAT (65H  Q WAVE NI.GT.1 DOES NOT STRIKE BOUNDARY BUT CATCHES Q
           1 WAVE NI-1)
        353 FORMAT (12H  ZIN(NI-1)=,F10.5,9H ZIN(NI)=,F10.5,13H SLPEN(NI-1)=
           1,F10.5,11H SLPEN(NI)=,F10.5,10H UN(NI-1)=,F10.5,8H UN(NI)=
           1,F10.5)
        712 FORMAT (10X,48HA Q WAVE HAS CAUGHT UP WITH ANOTHER Q WAVE AT M=,I3
           1,3HNI=,I3)
        291 FORMAT (67H  MORE THAN TWO Q WAVES REACH L.H. BOUNDARY IN FINE TIM
           1E LINE AT M=,I3,5X,3HNI=,I3)
        317 FORMAT (97H  LW WAVE NI MEETS RW WAVE NI-1, RESULTANT WAVE NI PUT
           1AT R.H. BOUNDARY BUT THERE IS WAVE NI+1 M=,I3,5X,3HNI=,I3)
        608 FORMAT (8H  A(NI)=,F10.3,6HU(NI)=,F10.3/5X,26HSONIC OUTFLOW TO AIR
           1 AT M=,I3,3HNI=,I3)
         12 FORMAT (22H  SONIC INFLOW TO PIPE)
```

```
  111 IF ( NI.LT.I ) GO TO 27
  630 CONTINUE
      NI = 0
      DO 640 NL=1,I
      PO(NL) = PON(NL)
      U(NL) = UN(NL)
      A(NL) = AN(NL)
      SLPE(NL) = SLPEN(NL)
      SZI(NL) = SZIN(NL)
      ZI(NL) = ZIN(NL)
      NTYPE(NL) = NTYPEN(NL)
      TMEVNT(NL) = TMEVNN(NL)
  640 CONTINUE
      T = T + DTT
      TINT = TINT + DTT
C
C
C     SUBROUTINE PRINTS OUT WAVE DIAGRAM CALCULATIONS AT DESIRED
C      INTERVALS.
C     THE WAVE DIAGRAM CALCULATIONS WILL BE PRINTED EVERY  NTH TIME
C      SUBROUTINE IS CALLED.
C
C---------------------------------------------------------------------
C
      IF ( JJ.LT.ND ) GO TO 2
      IF ( T.LT.TSTART .OR. T.GT.TSTOP ) GO TO 2
      JJ = 0
      WRITE (76,42)
      DO 44 K=1,I
      WRITE (76,43) A(K),U(K),SLPE(K),PO(K),SZI(K),ZI(K),NTYPE(K),K
     1,TMEVNT(K)
   44 CONTINUE
    2 JJ = JJ + 1
      IF ( M.LT.MM ) GO TO 120
      M = 0
      CALL WVCLOS ( 1 )
C     TIME FOR 1 REV. IS 60./RPMN SEC.
      IF ( T.LE.60./RPMN ) GO TO 866
C     CASE BELOW IS FOR ENGINE COMPLETING ONE REV.
C     IF KK IS NOT EQUAL TO NREV, WE MUST GO TO 866 ONE LAST TIME.
      IF ( KK.LT.NREV ) GO TO 866
C
C
C     SUBROUTINE SEARCHES ALL VALUES IN THE TIME ARRAY AND IF ANY ARE
C      GREATER THAN THE PERIOD OF ONE ENGINE REVOLUTION, THEN THAT AND
C      ALL SUBSEQUENT VALUES ARE REMOVED.
C
C---------------------------------------------------------------------
C
      N = 0
    6 N = N + 1
      IF ( TIMEX(N).GE.60./RPMN ) NTIM = N - 1
      IF ( N.LT.NTIM ) GO TO 6
      NNN = NNN + 1
      GO TO 998
C     CASE 866 IS FOR ENGINE NOT COMPLETING ONE REV.
  866 KK = KK + 1
```

E-2

```
      TMEVNN(NI) = DT - DARIER
      PQN(NI) = PQ(NI+1)
      PQN(NI+1) = PQ(NI)
      AN(NI) = ( GAMAM/4.0 ) * ( PQN(NI+1) + PQN(NI) )
      UN(NI) = ( PQN(NI+1) - PQN(NI) )/2.
      SLPEN(NI) = 1./( UN(NI) - AN(NI) )
      ZIN(NI) = ZI(NI) + ( DT - DARIER )/SLPEN(NI) + DARIER/SLPE(NI)
      SZIN(NI) = ( SZI(NI) + SZI(NI+1) )/2.
      NTYPEN(NI) = 2
   35 IF ( NI.LT.2 ) GO TO 277
C     BELOW ASKS WHETHER NI HAS BEEN CAUGHT UP TO BY NI-1 AT NEXT TIME
C      LINE.
      IF ( ZIN(NI-1).LT.ZIN(NI) ) GO TO 277
C     BELOW IS FOR NI-1 CROSSING NI.
      ZIN(NI) = ZIN(NI-1) + 1.E-6
  277 IF ( ZIN(NI).GT.0.0 ) GO TO 111
      IF ( NI.GT.1 ) GO TO 339
      ZIN(NI) = 1.E-6
      GO TO 111
  339 WRITE (70,348) M,NI
      CALL PRSTOP ( 4014 )
   34 IF ( NTYPE(NI)*NTYPE(NI+1)-2 ) 37,38,36
   37 CALL PRSTOP ( 4013 )
C     PROGRAM STOPS AS TWO ENTROPY WAVES MEET.
C     CASE BELOW IS FOR A RIGHTWARD ENTROPY WAVE NI MEETING A (LEFTWARD)
C      Q WAVE NI + 1.
C     CARIER IS TIME OF TRAVEL BEFORE WAVES MEET.
   38 CARIER = ( ZI(NI+1) - ZI(NI) ) / ( 1./SLPE(NI) - 1./SLPE(NI+1) )
      TMEVNT(NI+1) = TMEVNT(NI+1) + CARIER
      CALL PQADIF ( NI+1 )
      TMEVNN(NI) = DT - CARIER
      DARIER = EXP( GAMAB * ( ( PQ(NI) + SZI(NI+1) )/2. - SZI(NI) ) )
      AN(NI) = ( PQ(NI+1) + A(NI)/GAMAB + U(NI) )*GAMAB/( DARIER + 1. )
      UN(NI) = AN(NI)*DARIER/GAMAB - PQ(NI+1)
      SZIN(NI) = SZI(NI)
      SLPEN(NI) = 1./( UN(NI) - AN(NI) )
      PQN(NI) = AN(NI)/GAMAB - UN(NI)
      NTYPEN(NI) = 2
      ZIN(NI) = CARIER/SLPE(NI+1) + ( DT - CARIER )/SLPEN(NI) + ZI(NI+1)
      GO TO 35
C     CASE 36 IS FOR A (RIGHTWARD) P WAVE NI MEETING A LEFTWARD ENTROPY
C      WAVE NI + 1.
C     CARIER IS TIME OF TRAVEL BEFORE WAVES MEET.
   36 CARIER = ( ZI(NI+1) - ZI(NI) ) / ( 1./SLPE(NI) - 1./SLPE(NI+1) )
      TMEVNT(NI) = TMEVNT(NI) + CARIER
      CALL PQADIF ( NI )
      TMEVNN(NI+1) = DT - CARIER
      DARIER = EXP( GAMAB*( ( SZI(NI) + SZI(NI+1) )/2. - PQ(NI+1) ) )
      AN(NI) = ( PQ(NI) + A(NI+1)/( DARIER*GAMAB ) - U(NI+1) )
     1        * DARIER*GAMAB / ( DARIER + 1. )
      UN(NI) = PQ(NI) - AN(NI)/GAMAB
      SLPEN(NI) = 1./UN(NI)
      PQN(NI) = PQ(NI+1)
      SZIN(NI) = ( SZI(NI) + SZI(NI+1) )/2.
      NTYPEN(NI) = 1
      ZIN(NI) = ZI(NI+1) + CARIER/SLPE(NI+1) + ( DT - CARIER )/SLPEN(NI)
      GO TO 35
```

D2

```
C      OR THE BOUNDARY. IF SO, WE MUST REPOSITION WAVE NI+1.
   31 IF ( ZIN(NI+1).GE.ZLN ) ZIN(NI+1) = ZLN - 1.E-6
      IF ( ZIN(NI+1) .GE. ZI(NI+2) + DT/SLPE(NI+2) )
     1         ZI(NI+2) = ZIN(NI+1) - DT/SLPE(NI+2) + 1.E-7
      NI = NI + 1
      GO TO 111
C     CASE 311 IS FOR RIGHTWARD WAVE NI.LE.I-2 REACHING R.H. BOUNDARY IN
C      DT.
C     THIS IS NOT ALLOWABLE AS THERE ARE AT LEAST TWO WAVES BETWEEN NI
C      AND THE BOUNDARY. IF NI=I-2 AND NI+1, NI+2 ARE BOTH S WAVES, WITH
C      NI A P WAVE, WE WILL CALL WVCLOS  AND REPEAT CALCULATIONS.
C     OTHERWISE WE WILL STOP PROGRAM.
  311 IF ( NI.LE.I-3 .OR. NTYPE(NI+1)+NTYPE(NI+2).GE.2 .OR.
     1    NTYPE(NI).LE.2) GO TO 141
C     BELOW IS FOR NI=I-2, WAVES NI+1, NI+2 BOTH ENTROPY WAVES, WAVE NI
C      A P WAVE.
      IF (FLOAT(NWVDIS)*AO*DTAU/(FLOAT(MM)*ZLNM).LT.ZLN - ZI(NI))
     1  STOP 6000
C     THIS STOPS PROGRAM AS NWVDIS IS TOO LARGE TO ALLOW ADJACENT
C      ENTROPY WAVES TO MERGE. BELOW WE ARE SURE THAT THEY WILL MERGE.
      CALL WVCLOS ( NI+1 )
      GO TO 3
  141 IF ( NI.NE.I-3 ) GO TO 142
      IF ( NTYPE(NI+1)+NTYPE(NI+2)+NTYPE(NI+3).GT.1 .OR. NTYPE(NI).LT.3)
     1    GO TO 142
      IF (FLOAT(NWVDIS)*AO*DTAU/(FLOAT(MM)*ZLNM).LT.ZLN - ZI(NI))
     1    STOP 6001
C     THIS STOPS PROGRAM AS NWVDIS IS TOO LARGE TO ALLOW ADJACENT
C      ENTROPY WAVES TO MERGE. BELOW WE ARE SURE THAT THEY WILL MERGE.
      CALL WVCLOS ( NI+1 )
      GO TO 3
  142 WRITE (76,329) M,NI
      CALL PRSTOP ( 4007 )
C     CASE 309 IS FOR LEFTWARD WAVE NI+1.LE.I, RIGHTWARD WAVE NI, LESS
C      THAN I.
  309 IF ( ( ZI(NI+1) - ZI(NI) )/( 1./SLPE(NI) - 1./SLPE(NI+1) ) .LT.DT)
     1    GO TO 321
C     CASE BELOW IS FOR WAVES NOT MEETING.
      ZIN(NI) = ZI(NI) + DT/SLPE(NI)
      SZIN(NI) = SZI(NI)
      UN(NI) = U(NI)
      AN(NI) = A(NI)
      SLPEN(NI) = SLPE(NI)
      NTYPEN(NI) = NTYPE(NI)
      TMEVNN(NI) = TMEVNT(NI) + DT
      PQN(NI) = PQ(NI)
      GO TO 111
C     CASE 321 IS FOR RIGHTWARD WAVE NI.LE.I MEETING LEFTWARD WAVE NI+1,
C      IN DT.
  321 IF ( NTYPE(NI)+NTYPE(NI+1).LT.3 ) GO TO 34
C     CASE BELOW IS FOR P WAVE NI MEETING Q WAVE NI+1.
C     DARIER IS TIME OF TRAVEL BEFORE WAVES MEET.
      DARIER = ( ZI(NI+1) - ZI(NI) ) / ( 1./SLPE(NI) - 1./SLPE(NI+1) )
      TMEVNT(NI) = TMEVNT(NI) + DARIER
      TMEVNT(NI+1) = TMEVNT(NI+1) + DARIER
      CALL PQADIF ( NI )
      CALL PQADIF ( NI+1 )
```

C2

```
C      CATCHING A P WAVE.
    28 IF ( NTYPE(NI)*NTYPE(NI+1).LT.8 ) GO TO 29
       WRITE (76,349)
C      CASE 313 IS TO FOUL PROGRAM
   313 WRITE (76,315) M,NI,I
       IF ( NI.GT.I-1 ) GO TO 24
       CALL REMOVI ( NI+2,.FALSE. )
C      WE WILL NOW REWORK THE SAME FINE TIME LINE, SO THAT THE MERGED
C       WAVE MAY BE ADVANCED.
       NI = 1
       GO TO 10
C      CASE 24 IS FOR P WAVE NI=I-1 CATCHING P WAVE NI=I, FORMING A SHOCK
C       WAVE.
C      WE WILL SIMPLY REMOVE WAVE I IN PART 24
    24 I = I - 1
       JA = JA - 1
C      WE WILL NOW REWORK THE SAME FINE TIME LINE, SO THAT THE MERGED
C       WAVE MAY BE ADVANCED.
       NI = 1
       GO TO 10
C      CASE 29 IS FOR RIGHTWARD WAVE NI.LE.I-2 CATCHING UP ANOTHER
C       RIGHTWARD GOING WAVE, NI + 1.LE.I - 1, WHICH MUST BE AN ENTROPY
C       WAVE.
C     IF WAVE NI IS ALSO AN ENTROPY WAVE, WE HAVE AN ERROR AND MUST STOP
C      PROGRAM.
    29 IF ( NTYPE(NI)*NTYPE(NI+1).LT.2 ) CALL PRSTOP ( 4015 )
C      BELOW IS FOR A P WAVE NI.LE.I-2 MEETING S WAVE NI+1.LE.I-1. WE
C       MUST FIX THE VALUES FOR THE NEW S WAVE NI AND THE NEW P WAVE
C       NI+1, AND ALSO WE MUST SEE WHETHER NEW P WAVE MEETS ANY MORE
C       WAVES, OR THE BOUNDARY IN DT.
C      NI WILL THEN BE ADVANCED BY 1.
C      CARIER IS THE TIME OF TRAVEL BEFORE WAVES MEET.
       CARIER = ( ZI(NI+1) - ZI(NI) ) / ( 1./SLPE(NI) - 1./SLPE(NI+1) )
       TMEVNT(NI) = TMEVNT(NI) + CARIER
       CALL PQADIF ( NI )
       TMEVNN(NI+1) = DT - CARIER
       DARIER = EXP( GAMAB*( ( SZI(NI) + SZI(NI+1) )/2. - PQ(NI+1) ) )
       AN(NI+1) = ( PQ(NI) + A(NI+1)/( DARIER*GAMAB ) - U(NI+1) ) * GAMAB
      1          / ( DARIER + 1. )
       AN(NI) = AN(NI+1)*DARIER
       UN(NI+1) = PQ(NI) - AN(NI)/GAMAB
       UN(NI) = UN(NI+1)
       SLPEN(NI+1) = 1./( UN(NI+1) + AN(NI+1) )
       SLPEN(NI) = 1./UN(NI)
       PQN(NI+1) = AN(NI+1)/GAMAB + UN(NI+1)
       PQN(NI) = PQ(NI+1)
       SZIN(NI+1) = PQ(NI+1)
       SZIN(NI) = ( SZI(NI+1) + SZI(NI) )/2.
       NTYPEN(NI+1) = 3
       NTYPEN(NI) = 1
       ZIN(NI+1) = ZI(NI) + CARIER/SLPE(NI) + ( DT - CARIER )/SLPEN(NI+1)
       ZIN(NI) = ZI(NI+1) + CARIER/SLPE(NI+1) + ( DT - CARIER )/SLPEN(NI)
C      IF THERE IS A WAVE NI-1, WE MUST DETERMINE WHETHER ZIN(NI-1) IS
C       LESS THAN ZIN(NI). IF NOT, WE MUST REPOSITION WAVE NI.
       IF ( NI.LT.2 ) GO TO 31
       IF ( ZIN(NI-1).GE.ZIN(NI) ) ZIN(NI) = ZIN(NI-1) + 1.E-6
C      WE NOW MUST DETERMINE WHETHER WAVE NI+1 MEETS EITHER ANOTHER WAVE
```

B-2

```
      F = CARIER**2 * EXP( GAMAM*( SZIN(NI) - SZI(II) ) ) / GAMAB**2
     1  - PQ(NI)**2
      UN(NI) = ( PQ(NI) - SQRT( PQ(NI)**2 + E*F ) ) / E
      UN(NI+1) = UN(NI)
      AN(NI) = GAMAB*( PQ(NI) - UN(NI) )
      SLPEN(NI) = 1./( UN(NI) - AN(NI) )
      SLPEN(NI+1) = 1./UN(NI+1)
      AN(NI+1) = AN(NI)
      ZIN(NI+1) = ZLN
      NTYPEN(NI) = 2
      PQN(NI+1) = SZI(II)
      PQN(NI) = AN(NI)/GAMAB - UN(NI)
      NTYPEN(NI+1) = 1
      TMEVNN(NI) = .0
      TMEVNN(NI+1) = 0.
      SZI(II) = SZIN(NI+1)
      A(II) = AN(NI+1)
      U(II) = UN(NI+1)
      IF ( -UN(NI+1).LT.AN(NI+1) ) GO TO 635
      WRITE (76,12)
      CALL PRSTOP ( 4006 )
  635 NTIM = NTIM + 1
      UX(NTIM) = U(II)*A0
      TIMEX(NTIM) = T + DTT
      GO TO 630
C     CASE 25 IS FOR A (RIGHTWARD) ENTROPY WAVE TO REACH THE OUTLET.
C      WE NOW CHANGE R.H. BOUNDARY CONDITIONS AND REMOVE WAVE NI FROM
C      SYSTEM.
   25 SZI(II) = SZI(NI)
      A(II) = A(NI)
      U(II) = U(NI)
      I = I - 1
      JA = JA - 1
      GO TO 111
C     CASE 304 IS FOR RIGHTWARD WAVE NI, LESS THAN I
  304 IF ( SLPE(NI+1).LT.0.0 ) GO TO 309
C     CASE BELOW IS FOR RIGHTWARD WAVE WHICH DOES NOT MEET A LEFTWARD
C      WAVE. DOES WAVE STRIKE BOUNDARY, OR ANOTHER RIGHTWARD WAVE.
C     IF NI = I-1 WE MAY SEND PROGRAM TO SIMPLE ADVANCEMENT OF WAVE NI,
C      AS NEXT TIME AROUND LOOP NI WILL EQUAL I AND PROGRAM WILL BE SENT
C      TO 20 WHICH WILL CONSIDER RIGHTWARD WAVES I, I-1.
      IF ( NI.GT.I-2 ) GO TO 30
      IF ( DT/SLPE(NI) .GT. ( ZLN - ZI(NI) ) ) GO TO 311
      IF ( ZI(NI) + DT/SLPE(NI) .GE. ZI(NI+1) + DT/SLPE(NI+1) ) GO TO 28
C     CASE 30 IS FOR UNIMPEDED RIGHTWARD WAVE, STRIKING NEITHER
C      BOUNDARY, NOR ANOTHER RIGHTWARD WAVE.
   30 ZIN(NI) = ZI(NI) + DT/SLPE(NI)
      PQN(NI) = PQ(NI)
      SZIN(NI) = SZI(NI)
      UN(NI) = U(NI)
      AN(NI) = A(NI)
      SLPEN(NI) = SLPE(NI)
      NTYPEN(NI) = NTYPE(NI)
      TMEVNN(NI) = TMEVNT(NI) + DT
      GO TO 111
C     CASE 28 IS FOR RIGHTWARD WAVE NI.LE.I-2 TO MEET RIGHTWARD WAVE
C      NI+1.LE.I-1 IT WILL BE ASSUMED THAT WAVE NI IS NOT AN S WAVE
```

A-2

```
      SLPEN(NI) = 1./( UN(NI) + AN(NI) )
      SLPEN(NI-1) = 1./UN(NI-1)
      PQN(NI) = AN(NI)/GAMAB + UN(NI)
      PQN(NI-1) = PQ(NI)
      SZIN(NI) = PQ(NI)
      SZIN(NI-1) = ( SZI(NI-1) + SZI(NI) )/2.
      NTYPEN(NI) = 3
      NTYPEN(NI-1) = 1
      ZIN(NI) = ZI(NI-1) + CARIER/SLPE(NI-1) + ( DT - CARIER )/SLPEN(NI)
      ZIN(NI-1) = ZI(NI) + CARIER/SLPE(NI) + ( DT - CARIER )/SLPEN(NI-1)
      IF ( NI.LT.3 ) GO TO 22
C     CASE BELOW IS FOR WAVE NI-2 TO EXIST.
      IF ( ZIN(NI-1).LE.ZIN(NI-2) ) ZIN(NI-1) = ZIN(NI-2) + 1.E-6
      IF ( ZIN(NI-1).GE.ZLN ) CALL PRSTOP ( 4032 )
C     CASE 22 IS TO SEE WHETHER NEW P WAVE NI MEETS R.H. BOUNDARY IN DT.
   22 IF ( ZIN(NI).GE.ZLN ) ZIN(NI) = ZLN - 1.E-6
      GO TO 111
C     CASE 308 IS FOR LOCATING AT BOUNDARY AT TIME + DT.
  308 IF ( NTYPE(NI).LT.2 ) GO TO 25
      CALL PQADIF ( NI )
      ZIN(NI) = ZLN
      AN(NI) = EXP(GAMAM*(SZIN(NI) - SO)/2.)
      SZIN(NI) = (SZI(NI) + SZI(II))/2.
      UN(NI) = PQ(NI) - 2.*AN(NI)/GAMAM + 1.E-10
      IF ( UN(NI).LT.0.0 ) GO TO 397
      IF ( UN(NI).LT.AN(NI) ) GO TO 393
      WRITE (76,608) A(NI),U(NI),M,NI
      CALL PRSTOP ( 4012 )
C     CASE 393 IS FOR SUBSONIC OUTFLOW.
  393 PQN(NI) = 2.*AN(NI)/GAMAM - UN(NI)
      SLPEN(NI) = 1./( UN(NI) - AN(NI) )
      TMEVNN(NI) = 0.
      NTYPEN(NI) = 2
C     BELOW GIVES VALUES IN AN(NI), ETC. TO A(II), ETC. WHICH FIX THE
C      FLOW VALUES AT THE R.H. BOUNDARY.
      U(II) = UN(NI)
      A(II) = AN(NI)
      SZI(II) = SZIN(NI)
      NTIM = NTIM + 1
      UX(NTIM) = U(II)*AO
      TIMEX(NTIM) = T + DTT
      GO TO 111
C     CASE 397 IS FOR UN(NI) NEGATIVE GIVING INFLOW. WE MUST CALCULATE
C      THE Q WAVE REFLECTION AND ADD A NEW ENTROPY WAVE. IN DOING THIS
C      WE ASSUME THAT GAS COMES ISENTROPICALLY FROM A RESERVOIR AT
C      ATMOSPHERIC PRESSURE AND WITH AN ENTROPY EQUAL TO THAT AT THE
C      OPEN PIPE OUTLET FOR THE LAST PERIOD OF OUTFLOW.
C     CARIER HERE IS THE STAGNATION SPEED OF SOUND OF THE INFLOW.
  397 CARIER = EXP( GAMAB*( SZI(II) - SO ) )
C     WE WILL ADD AN ENTROPY WAVE TO THE SYSTEM, BUT FIRST WE MUST
C      INCREASE I, JA BY 1.
C     WE WILL USE THE REUSEABLE VARIABLES E AND F TO SIMPLIFY.
      I = I + 1
C     NEW ENTROPY VALUES WILL BE SET.
      JA = JA + 1
      SZIN(NI+1) = SZIN(NI)
      E = 1. + EXP( GAMAM*( SZIN(NI) - SZI(II) ) ) / GAMAB
```

M-1

```
            TMEVNI(NI) = TMEVNI(NI) + CARIER
            CALL PQADIF ( NI )
            TMEVNN(NI-1) = DT - CARIER
            DARIER = EXP( GAMAB*( ( PQ(NI-1) + SZI(NI) )/2. - SZI(NI-1) ) )
            AN(NI) = ( PQ(NI) + A(NI-1)/GAMAB + U(NI-1) )*GAMAB/( DARIER + 1.)
            UN(NI) = AN(NI)*DARIER/GAMAB - PQ(NI)
            SZIN(NI) = SZI(NI-1)
            SLPEN(NI) = 1./UN(NI)
            PQN(NI) = ( PQ(NI-1) + SZI(NI) )/2.
            NTYPEN(NI) = 1
            ZIN(NI) = CARIER/SLPE(NI-1) + ( DT - CARIER )/SLPEN(NI) + ZI(NI-1)
            GO TO 17
      C     MUST CONSIDER RIGHTWARD WAVE.
          3 IF ( NI.NE.I ) GO TO 304
      C     CASE BELOW IS A SPECIAL PROGRAM FOR RIGHTWARD WAVE AT R.H. END OF
      C      SYSTEM.
            IF ( DT/SLPE(NI) .GT. ( ZLN - ZI(NI) ) ) GO TO 308
            IF ( NI.GT.1 ) GO TO 20
      C     CASE BELOW IS FOR RIGHTWARD WAVE NI = I, WHICH DOES NOT MEET
      C      ANOTHER RIGHTWARD WAVE NI-1.
         21 ZIN(NI) = ZI(NI) + DT/SLPE(NI)
            UN(NI) = U(NI)
            AN(NI) = A(NI)
            SLPEN(NI) = SLPE(NI)
            SZIN(NI) = SZI(NI)
            NTYPEN(NI) = NTYPE(NI)
            TMEVNN(NI) = TMEVNI(NI) + DT
            PQN(NI) = PQ(NI)
            GO TO 111
      C     AT 20 WE MUST SEE WHETHER RIGHTWARD WAVE NI WILL BE CAUGHT UP BY
      C      WAVE NI-1 IN DT.
         20 IF ( ZI(NI) + DT/SLPE(NI) .GT. ZIN(NI-1) ) GO TO 21
      C     CASE BELOW IS FOR RIGHTWARD WAVE NI = I TO MEET WAVE NI-1 (WHICH
      C      MUST BE A RIGHTWARD WAVE). IT WILL BE ASSUMED THAT WAVE NI-1 IS
      C      NOT AN S WAVE CATCHING A P WAVE NI.
      C     PROGRAM WILL BE SENT TO 313 IF BOTH NI-1, NI ARE P WAVES MEETING.
            IF ( NTYPE(NI-1)*NTYPE(NI).GT.8 ) GO TO 313
      C     IF BOTH WAVES ARE S WAVES WE WILL STOP PROGRAM. WE NEED ONLY ASK
      C      WHETHER NI-1 IS S WAVE, AS WE CANNOT GET HERE IF NI-1 IS S AND NI
      C      IS P. (IF NI-1 IS S WAVE, NI IS P WAVE, PROGRAM WILL STOP HERE
      C      ANYWAY).
            IF ( NTYPE(NI-1).LT.2 ) CALL PRSTOP ( 4005 )
      C     CASE BELOW IS FOR S WAVE NI BEING CAUGHT BY P WAVE NI-1. WE MUST
      C      FIX THE VALUES FOR THE NEW S WAVE NI-1 AND FOR THE NEW P WAVE NI,
      C      AND ALSO MUST SEE WHETHER NEW P WAVE NI MEETS BOUNDARY, OR
      C      WHETHER NEW S WAVE NI-1 MEETS WAVE NI-2.
      C     CARIER IS THE TIME OF TRAVEL BEFORE WAVES MEET.
            CARIER = ( ZI(NI) - ZI(NI-1) ) / ( 1./SLPE(NI-1) - 1./SLPE(NI) )
            TMEVNI(NI-1) = TMEVNI(NI-1) + CARIER
            CALL PQADIF ( NI-1 )
            TMEVNN(NI) = DT - CARIER
            DARIER = EXP( GAMAB*( ( SZI(NI-1) + SZI(NI) )/2. - PQ(NI) ) )
            AN(NI) = ( PQ(NI-1) + A(NI)/( DARIER*GAMAB ) - U(NI) ) * GAMAB
          1          / ( DARIER + 1. )
            AN(NI-1) = AN(NI)*DARIER
            UN(NI) = PQ(NI-1) - AN(NI)*DARIER/GAMAB
            UN(NI-1) = UN(NI)
```

L-1

```
   33 IF ( ( ZI(NI) - ZI(NI-1) )/( 1./SLPE(NI-1) - 1./SLPE(NI) ) .GE.DT)
     1 GO TO 55
      IF ( NTYPE(NI)*NTYPE(NI-1).LT.5 ) GO TO 14
C     CASE BELOW IS FOR LEFTWARD Q WAVE MEETING RIGHTWARD P WAVE.
C     DARIER IS TIME OF TRAVEL BEFORE WAVES MEET.
      DARIER = ( ZI(NI) - ZI(NI-1) )/( 1./SLPE(NI-1) - 1./SLPE(NI) )
      TMEVNT(NI-1) = TMEVNT(NI-1) + DARIER
      TMEVNT(NI) = TMEVNT(NI) + DARIER
      CALL PQADIF ( NI )
      CALL PQADIF ( NI-1 )
      TMEVNN(NI) = DT - DARIER
      PQN(NI) = PQ(NI-1)
      PQN(NI-1) = PQ(NI)
      AN(NI) = ( GAMAM/4. ) * ( PQN(NI) + PQN(NI-1) )
      UN(NI) = ( PQN(NI) - PQN(NI-1) )/2.
      SLPEN(NI) = 1./( UN(NI) + AN(NI) )
      ZIN(NI) = ZI(NI) + ( DT - DARIER )/SLPEN(NI) + DARIER/SLPE(NI)
      SZIN(NI) = ( SZI(NI) + SZI(NI-1) )/2.
      NTYPEN(NI) = 3
   17 IF ( NI.EQ.I ) GO TO 271
C     BELOW ASKS WHETHER NI HAS CAUGHT UP WITH NI+1 AT NEXT TIME LINE.
      IF ( ZIN(NI) .LT. ZI(NI+1) + DT/SLPE(NI+1) ) GO TO 271
C     BELOW IS FOR NI+1 CROSSING NI, MUST RELOCATE ZIN(NI) SO THAT
C      LINES NI, NI+1 DO NOT CROSS.
      ZIN(NI) = ZI(NI+1) + DT/SLPE(NI+1) - 1.E-6
  271 IF ( ZIN(NI).LT.ZLN ) GO TO 111
C     BELOW IS TO LOCATE PT. NI TO NEAR BOUNDARY AT START OF NEXT FINE
C      TIME LINE.
      IF ( NI.GT.I-1 ) GO TO 302
      WRITE (76,317) M,NI
      CALL PRSTOP ( 4010 )
  302 ZIN(NI) = ZLN - 1.E-6
      GO TO 111
   14 IF ( NTYPE(NI)*NTYPE(NI-1)-2 ) 15,16,18
   15 CALL PRSTOP ( 4004 )
C     PROGRAM STOPS AS TWO ENTROPY WAVES MEET.
C     CASE BELOW IS FOR A LEFTWARD ENTROPY WAVE NI MEETING A (RIGHTWARD)
C      P WAVE NI-1.
C     CARIER IS TIME OF TRAVEL BEFORE WAVES MEET.
   18 CARIER = ( ZI(NI) - ZI(NI-1) )/( 1./SLPE(NI-1) - 1./SLPE(NI) )
      TMEVNT(NI-1) = TMEVNT(NI-1) + CARIER
      CALL PQADIF ( NI-1 )
      TMEVNN(NI) = DT - CARIER
      DARIER = EXP( GAMA3*( ( SZI(NI-1) + SZI(NI) )/2. - PQ(NI) ) )
      AN(NI) = ( PQ(NI-1) + A(NI)/( DARIER*GAMAB ) - U(NI) ) * GAMAB
     1        / ( DARIER + 1. )
      UN(NI) = PQ(NI-1) - AN(NI)*DARIER/GAMAB
      SLPEN(NI) = 1./( UN(NI) + AN(NI) )
      PQN(NI) = AN(NI)/GAMAB + UN(NI)
      SZIN(NI) = PQ(NI)
      NTYPEN(NI) = 3
      ZIN(NI) = ZI(NI-1) + CARIER/SLPE(NI-1) + ( DT - CARIER )/SLPEN(NI)
      GO TO 17
C     CASE 16 IS FOR A LEFTWARD Q WAVE NI MEETING A RIGHTWARD S WAVE
C      NI-1.
C     CARIER BELOW IS THE TIME OF TRAVEL BEFORE WAVES MEET.
   16 CARIER = ( ZI(NI) - ZI(NI-1) ) / ( 1./SLPE(NI-1) - 1./SLPE(NI) )
```

16-1

```
      WRITE (76,353) ZIN(NI-1),ZIN(NI),SLPEN(NI-1),SLPEN(NI),UN(NI-1)
     1,UN(NI)
C     CASE BELOW IS TO FOUL PROGRAM
      WRITE (76,712) M,NI
      CALL REMOV1 ( NI,.FALSE. )
C     WE WILL NOW REWORK THE SAME FINE TIME LINE, SO THAT THE MERGED
C      WAVE MAY BE ADVANCED.
      NI = 1
      GO TO 10
C     PART 13 CATERS FOR A LEFTWARD GOING WAVE, NI, CATCHING ANOTHER
C      LEFTWARD GOING WAVE, NI-1, WHICH IS AN ENTROPY WAVE. IF WAVE NI
C      IS AN ENTROPY WAVE ALSO, WE HAVE AN ERROR AND WE MUST STOP
C      PROGRAM.
   13 IF ( NTYPE(NI).LT.2 ) CALL PRSTOP ( 4003 )
C     CASE BELOW IS FOR Q WAVE NI CROSSING LEFTWARD S WAVE NI-1.
C     WE MUST FIX THE VALUES FOR NEW Q WAVE NI-1 AND FOR NEW S WAVE NI,
C      AND ALSO WE MUST SEE WHETHER NEW Q WAVE NI-1 MEETS ANY MORE WAVES
C      OR THE BOUNDARY IN DT.
C     CARIER, HERE, IS THE TIME OF TRAVEL BEFORE WAVES MEET.
      CARIER = ( ZI(NI) - ZI(NI-1) )/( 1./SLPE(NI-1) - 1./SLPE(NI) )
C     WE MUST UPDATE PQ VALUE OF Q WAVE AND RESET CLOCKS. THE ACCURACY
C      OF THE CLOCK ON ENTROPY WAVE WILL ONLY BE ROUGHLY MAINTAINED,
C      MORE ACCURATELY FOR PRESSURE WAVE.
      TMEVNT(NI) = TMEVNT(NI) + CARIER
      CALL PQADIF ( NI )
      TMEVNN(NI-1) = DT - CARIER
C     DARIER IS A REUSEABLE VARIABLE, HERE USED TO SIMPLIFY.
      DARIER = EXP( GAMAB*( (PQ(NI-1) + SZI(NI))/2. - SZI(NI-1) ) )
      AN(NI-1) = ( PQ(NI) + A(NI-1)/GAMAB + U(NI-1) )*GAMAB
     1          / ( DARIER + 1. )
      AN(NI) = AN(NI-1)
      UN(NI-1) = AN(NI-1)*DARIER/GAMAB - PQ(NI)
      UN(NI) = UN(NI-1)
      SZIN(NI-1) = SZI(NI-1)
      SZIN(NI) = SZI(NI-1)
      SLPEN(NI-1) = 1./( UN(NI-1) - AN(NI-1) )
      SLPEN(NI) = 1./UN(NI)
      PQN(NI-1) = AN(NI-1)/GAMAB - UN(NI-1)
      PQN(NI) = ( PQ(NI-1) + SZI(NI) )/2.
      NTYPEN(NI) = 1
      NTYPEN(NI-1) = 2
      ZIN(NI-1) = CARIER/SLPE(NI) + ( DT - CARIER )/SLPEN(NI-1) + ZI(NI)
      ZIN(NI) = CARIER/SLPE(NI-1) + ( DT - CARIER )/SLPEN(NI) + ZI(NI-1)
      IF ( NI.LT.3 ) GO TO 23
C     CASE BELOW IS FOR WAVE NI-2 TO EXIST.
      IF ( ZIN(NI-1).LE.ZIN(NI-2) ) ZIN(NI-1) = ZIN(NI-2) + 1.E-6
      GO TO 111
C     CASE 23 IS FOR NI-1 = 1
   23 IF ( ZIN(NI-1).LE.0. ) ZIN(NI-1) = 1.E-6
      GO TO 111
    4 CONTINUE
      IF ( NI.LT.3 ) GO TO 292
      WRITE (76,291) M,NI
      CALL PRSTOP ( 4011 )
  292 NI = NI - 1
      GO TO 606
C     CASE 33 IS FOR LEFTWARD WAVE NI AND RIGHTWARD WAVE NI-1
```

J-1

```
      NTYPE(NI) = 3
      PRATIO = PCT/PDT
      UDLAST = U(NI)
      TINT = 0.
      U(II-1) = U(NI)
      CALL ADDILH
      ZI(NI) = 0.
C     VALUE IN U(NI) IS VELOCITY OF INTERFACE.
      U(NI) = U(NI+1)
C     VALUE IN A(NI) IS SPEED OF SOUND TO LEFT OF INTERFACE.
      A(NI) = A(II-1)
      SLPE(NI) = 1./U(NI)
C     VALUE IN SZI(NI) IS ENTROPY TO LEFT OF INTERFACE, VALUE IN PQ(NI)
C      IS ENTROPY TO RIGHT.
      SZI(NI) = SDT
      PQ(NI) = SZI(NI+1)
      NTYPE(NI) = 1
      TMEVNT(NI) = 0.
      GO TO 119
C     CASE 545 IS FOR INFLOW
  545 CALL FLOWIN ( A(NI),PQ(NI),U(NI) )
      SZI(NI) = SDT
      SLPE(NI) = 1./( U(NI) + A(NI) )
      ZI(NI) = 0.
      NTYPE(NI) = 3
      PRATIO = PDT/PCI * ( 1. + GAMAB * (U(NI)/A(NI))**2 )**(GAMA/GAMAM)
      UDLAST = U(NI)
      TINT = 0.
      A(II-1) = A(NI)
      U(II-1) = U(NI)
C     II-1 IS THE LOCATION IN THE MATRIX A(NI) WHICH REFERS TO THE (L.H)
C      EXHAUST PORT BOUNDARY.
  119 SZI(II-1) = SDT
      GO TO 10
C     MUST CONSIDER GENERAL PROGRAM FOR A PT.
    1 IF ( SLPE(NI).GE.0.0 ) GO TO 3
C     CASE BELOW CATERS FOR Q WAVE OR ENTROPY WAVE.
      IF ( SLPE(NI-1).GE.0.0 ) GO TO 33
C     CASE BELOW IS FOR LEFTWARD WAVE WHICH DOES NOT MEET A RIGHTWARD
C      WAVE. DOES THE WAVE STRIKE BOUNDARY, OR ANOTHER LEFTWARD WAVE.
      IF ( ZI(NI) + DT/SLPE(NI) .LE. ZIN(NI-1) ) GO TO 11
      IF ( -DT/SLPE(NI).GE.ZI(NI) ) GO TO 4
C     CASE BELOW IS FOR UNIMPEDED LEFTWARD WAVE, STRIKING NEITHER
C      BOUNDARY NOR WAVE WITH POSITIVE SLPE.
      ZIN(NI) = ZI(NI) + DT/SLPE(NI)
      PQN(NI) = PQ(NI)
      SZIN(NI) = SZI(NI)
      UN(NI) = U(NI)
      AN(NI) = A(NI)
      SLPEN(NI) = SLPE(NI)
      NTYPEN(NI) = NTYPE(NI)
      TMEVNN(NI) = TMEVNT(NI) + DT
      GO TO 111
C     CASE 11 IS FOR LEFTWARD WAVE NI TO MEET LEFTWARD WAVE NI - 1. IT
C      WILL BE ASSUMED THAT WAVE NI IS NOT AN S WAVE MEETING A Q WAVE.
   11 IF ( NTYPE(NI-1).LT.2 ) GO TO 13
      WRITE (76,345)
```

I-1

```
      UN(NI) = U(NI)
      AN(NI) = A(NI)
      SLPEN(NI) = SLPE(NI)
      SZIN(NI) = SZI(NI)
      NTYPEN(NI) = NTYPE(NI)
      TMEVNN(NI) = TMEVNT(NI) + DT
      PQN(NI) = PQ(NI)
      GO TO 111
C     PT. ON NEXT TIME LINE FOUND SO JUMP OUT AND GO TO NEXT PT.
C     CASE 55 IS FOR ADVANCEMENT OF LEFTWARD WAVE NI.GT.1. DOES NOT MEET
C      RIGHT-WARD WAVE NI-1, SO PROCEEDS UNIMPEDED.
   55 ZIN(NI) = ZI(NI) + DT/SLPE(NI)
      SZIN(NI) = SZI(NI)
      UN(NI) = U(NI)
      AN(NI) = A(NI)
      SLPEN(NI) = SLPE(NI)
      NTYPEN(NI) = NTYPE(NI)
      TMEVNN(NI) = TMEVNT(NI) + DT
      PQN(NI) = PQ(NI)
      GO TO 111
C     CASE 606 IS FOR STARTING PT. RELOCATED TO BOUNDARY.
  606 IF ( T.LT.X/(3.*RPMN) ) GO TO 541
C     CASE BELOW IS FOR PORT TO BE CLOSED
      IF ( NTYPE(NI).LT.2 ) CALL PRSTOP ( 4001 )
      CALL PQADIF ( NI )
      ZI(NI) = 0.0
      U(NI) = 0.0
      A(NI) = GAMAM*PQ(NI)/2.
      SLPE(NI) = 1./(U(NI) + A(NI))
      SZI(NI) = SDT
      NTYPE(NI) = 3
      PDT = PO * A(NI)**ZB * EXP( -GAMA*(SZI(NI) - SD))
      GO TO 10
C     CASE 541 IS FOR PORT OPEN. WE MUST FIRST SEE IF WAVE IS ENTROPY
C      WAVE.
  541 IF (NTYPE(NI).GT.1) GO TO 9
      IF ( U(NI).GT.0. ) CALL PRSTOP ( 4002 )
C     CASE BELOW IS FOR WAVE NI, AN S WAVE, REACHING L.H. BOUNDARY AND
C      GOING INTO CYLINDER.
      A(II-1) = A(II-1)*EXP(GAMAB*(PQ(NI) - SDT))
      SDT = PQ(NI)
      CALL REMOVI ( 2,.FALSE. )
      SZI(II-1) = SDT
      GO TO 10
C     WE MUST TAKE CARE OF INITIAL CASE BY SENDING PROGRAM TO AREAVL
    9 CALL PQADIF ( NI )
      IF ( NC.LT.1 ) GO TO 319
      PCJ = PCO*SXN
      CALL MASSEX
  319 AL = ( PCT/PO )**(1./ZB) * EXP( GAMAB*( SDT - SD ) )
      CALL AREAVL
      NC = 2
      IF ( PQ(NI).GT.AL/GAMAB ) GO TO 545
      SZI(NI) = SDT
      CALL FLWOUT ( A(NI),PQ(NI),U(NI),A(II-1) )
      SLPE(NI) = 1./( U(NI) + A(NI) )
      ZI(NI) = 0.
```

H-1

```
      SUBROUTINE STRIPE
C
C
C        SUBROUTINE WILL CALCULATE WAVE DIAGRAM FOR A TUNED EXHAUST PIPE
C         SYSTEM.
C        THE PRESSURE AT A POINT IN THE EXHAUST SYSTEM, THE VELOCITY AT
C         THE TAILPIPE OUTLET AND THE RADIATED SOUND PRESSURE AT A POINT
C         IN THE FAR FIELD ARE ALL PLOTTED.
C        THE RMS SPL IS FOUND FOR BOTH PRESSURE TRACES.
C        PROGRAM ASSUMES NO FRICTION WITH PIPE WALLS, NO HEAT LOSS
C         THROUGH WALLS, BUT GAS IN PIPE HAS OTHERWISE CORRECT ENTROPY
C         VALUES.
C        SUBROUTINE ALLOWS FOR A TUNED SYSTEM WITH 5 SEGMENTS.
C         FIRST SEGMENT IS STRAIGHT, 2ND IS EXPANDING CONE SECTION,
C         3RD IS STRAIGHT, 4TH IS CONTRACTING CONE AND 5TH IS STRAIGHT
C         TAILPIPE.
C         THERE ARE THREE DIAMETERS AND FIVE LENGTHS.
C         THE DIAMETERS ARE D1 TO D3 AND LENGTHS ARE XL1 TO XL5.
C
C-----------------------------------------------------------------------
C
      COMMON /A/ A(100),ACT,ACTC,ACO,ACOC,AD,AE,AL,AN(100),AT,ATC,AO
      COMMON /B/ B,BBQ,BC,BORE,BOREA
      COMMON /C/ CONLEN,CP,CTX
      COMMON /D/ DIST,DT,DTAU,DTT,D1,D2,D3
      COMMON /G/ GAMA,GAMAB,GAMAM,GAMAP
      COMMON /I/ I,II,IJ
      COMMON /J/ J,JA,JJ,JREV,JREX
      COMMON /K/ KJ,KK
      COMMON /M/ M,MM
      COMMON /N/ NC,ND,NI,NNN,NNZ,NREV,NTIM,NTYPE(100),NTYPEN(100)
     1          ,NWAVES,NWVDIS
      COMMON /O/ ONETHD
      COMMON /P/ PCJ,PCT,PCTC,PCO,PCOC,PDT,PE(400),PI,PQ(100),PQN(100)
     1          ,PRATIC,PRATIO,PX(1200),PO
      COMMON /Q/ Q
      COMMON /R/ R,RC,RGAS,RPMN
      COMMON /S/ SCCO,SCT,SCTC,SCO,SD,SDT,SLPE(100),SLPEN(100),STROKE
     1          ,SXN,SZI(100),SZIN(100)
      COMMON /T/ T,TCT,TCTC,TCO,TCOC,TIME(400),TIMEX(800),TINT,TLAMIC
     1          ,TLAMIN,TMEVNN(100),TMEVNT(100),TRDUC,TRDUCE,TSTART
     2          ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /V/ VCT,VCTC,VCTCO,VCO,VCOC,VS,VSC
      COMMON /X/ X,XC,XCI,XE,XEC,XL1,XL3,XZ2,XZ4,X1,X2,X3,X4
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
  120 M = M + 1
   27 NI = NI + 1
   10 IF ( NI.GT.1 ) GO TO 1
      IF ( SLPE(NI).GE.0.0 ) GO TO 2
      IF ( -DT/SLPE(NI).GE.ZI(NI) ) GO TO 606
C     CASE BELOW IS FOR Q WAVE, OR LEFTWARD ENTROPY WAVE WHICH DOES NOT
C      MEET A RIGHTWARD WAVE.
      ZIN(NI) = ZI(NI) + DT/SLPE(NI)
```

G-1

```
      C-----------------------------------------------------------------------
      C
345        CALL STPIPE
      C
      C-----------------------------------------------------------------------
      C
           WRITE (1) (PE(L),L=1,NREV),(TIME(L),L=1,NREV),(UX(L),L=1,NTIM)
350       1,(TIMEX(L),L=1,NTIM)
           WRITE (76,25) NREV, NTIM, JA
      C
      C-----------------------------------------------------------------------
      C
355     130 FORMAT (100H  ADJACENT ENTROPY WAVES WILL BE MERGED BY WVCLOS IF T
          1HEY ARE CLOSER THAN NON-DIMENSIONAL DISTANCE =,F7.3)
         25 FORMAT (7H NREV =,I4,4X,6HNTIM =,I5/26H  NO. OF WAVES IN SYSTEM =,
          1I4)
           END
```

```
      TCT = TCO
      NREV = 60./(RPMN*DTAU)
      X[illegible] = (D2 - D1)/XL2
      XZ4 [illegible] (D3 - D2)/XL4
      X2 = [illegible]1 + XL2
      X3 = X[illegible] XL3
      X4 = X3 + XL4
      TRDUCE = TRDUC/ZLNM
      KK = 0
      JJ = 0
      ZLN = XL1 + XL2 + XL3 + XL4 + XL5
      DT = AO*DTAU/(FLOAT(MM)*ZLNM)
      DTT = DTAU/FLOAT( MM )
      T = DTT
      VSC = VS
      XEC = XC*PI/180.
C     INITIAL VALUES (AT I.P.O.)OF CRANKCASE VARIABLES WILL BE SET
      VCOC = VOLEX ( -1.,PI*( X-XCI )/180.-XE,VSC,RC )
      ATC = 1.E-10
      TCOC = (ACOC**2)/(GAMA*RGAS)
      VCTCO = VOLEX ( -1.,PI*( X-XC )/180.-XE,VSC,RC )
      VCTC = VCTCO
C     PCTC PRESS.IN CRANKCASE WILL INITIALLY BE SET TO VALUE AT T.P.O.
      PCTC = PCOC * (VCOC/VCTCO)**GAMA
      SCT = SCO
C     CP, SPECIFIC HEAT AT CONST. PRESS. DEFINED.
C     INITIAL VALUE OF ENTROPY IN CRANKCASE SET RELATIVE TO ENTROPY
C      IN CYLINDER.
      CP = GAMA*RGAS/GAMAM
      SCCO = SCO + CP*ALOG( ACOC**2 / (GAMA*RGAS*TCO) )
     1      - RGAS*ALOG( PCOC / (PCO*SXN) )
C     SCTC, ENTROPY IN CRANKCASE SET TO SCCO.
      SCTC = SCCO
C     INITIAL VALUE OF TCTC, TEMP. IN CRANKCASE SET TO VALUE AT T.P.O.
      TCTC = TCOC * (VCOC/VCTCO)**GAMAM
C     SET INITIAL VALUE OF ACTC FROM TCTC
      ACTC = SQRT( GAMA*RGAS*TCTC )
      SD =ALOG(PCO*SXN/PO)/GAMA +2.*ALOG(AO/ACO)/GAMAM + SCO/(GAMA*RGAS)
      IF (NNN.GT.0) GO TO 560
      JA = 2 * IFIX( X/( 24.*RPMN*DTAU ) )
      IF ( NWAVES.GT.1 ) JA = NWAVES
      CALL WVSTAT
  560 CONTINUE
      I = JA
      SDT = SZI(1)
C     MUST GIVE INITIAL VALUES TO U(II),U(II - 1),SZI(II),SZI(II - 1)
      U(II) = U(I)
      U(II - 1) = U(1)
      SZI(II) = SZI(I)
      SZI(II - 1) = SZI(1)
      A(II) = A(I)
      A(II - 1) = A(1)
      NI = 0
      M = 0
      CARIER = FLOAT(NWVDIS)*AO*DTAU/(FLOAT(MM)*ZLNM)
      WRITE (76,130) CARIER
C
```

```
      C
      C        SIZES OF ARRAYS PE, TIME SHOULD NOT BE LESS THAN JREV.
      C        SIZES OF ARRAYS UX, TIMEX SHOULD NOT BE LESS THAN JREX.
      C
      C-----------------------------------------------------------------------
      C
            CALL PLOT25
            CALL XLIMIT (400.0)
      C
      C-----------------------------------------------------------------------
      C
            PI = 3.1415927
            READ(75,31)IJ,DIST,STROKE,B,R,VS,BORE,CONLEN
         31 FORMAT(I3,2F10.5,F12.10,F10.5,F12.10,2F10.5)
            READ(75,32)X,ZLNM,D1,D2,D3,XL1,XL2,XL3
         32 FORMAT(8F10.5)
            READ(75,33)XL4,XL5
         33 FORMAT(2F10.5)
      C     WE MUST NON-DIMENSIONALISE ALL THE ABOVE VALUES BY ZLNM=.025 METRE
            D1=D1/ZLNM
            D2=D2/ZLNM
            D3=D3/ZLNM
            XL1=XL1/ZLNM
            XL2=XL2/ZLNM
            XL3=XL3/ZLNM
            XL4=XL4/ZLNM
            XL5=XL5/ZLNM
            READ(75,34)II,GAMA,RGAS,PCO,TRDUC,MM,BBQ,XCI,XC
         34 FORMAT(I3,F12.10,3F10.5,I3,3F10.5)
            READ(75,35)RC,BC,KJ,NWVDIS,ND,TSTART,TSTOP,NNN,J,NNZ
         35 FORMAT(F10.5,F12.10,3I3,2F10.5,3I3)
            NTIM = 0
            NC = 0
            TINT = 0.
            BOREA=PI*BORE*BORE/4.
            Q = 0.5*STROKE/CONLEN
            XE = X*PI/180.
            GAMAM = GAMA - 1.0
            GAMAP = GAMA + 1.0
            ZB = 2.*GAMA/GAMAM
            CTX = ( 2./GAMAP )**( ( GAMAP/2. )/GAMAM )
            PRATIC = ( GAMAP/2. )**( GAMA/GAMAM )
            GAMAB = GAMAM/2.
      C     SXN IS CONSTANT TO CHANGE METRIC PRESSURE TO PSI.
            SXN = 6900.
            AD=PI*(D1*ZLNM)**2/4.
            AE=PI*(D3*ZLNM)**2/4.
            ACT = ACO
            PCT = PCO*SXN
      C     MUST SET INITIAL VALUES OF TLAMIN, TLAMIC TO ZERO
            TLAMIN = 0.
            TLAMIC = 0.
      C     INITIAL VALUE OF PDT IS ATMOS. PRESS M.K.S. UNITS
            READ(75,36)PO,PDT,AO,ACOC,PCOC,SCO
         36 FORMAT(2F10.1,2F10.5,F10.1,F10.5)
            VCO = VOLEX ( 1.,-XE,VS,R )
            TCO = ( ACO**2 )/( GAMA*RGAS )
```

```
C      TSTOP IS TIME IN SECS. AFTER E.P.O. WHEN SUBROUTINE PRSTOP CEASES
C       TO PRINT OUT WAVE DIAGRAM INFORMATION.
C       NOTE- IN FORM OF PROGRAM AS PRESENTED IN THIS THESIS, BOTH
C       TSTART AND TSTOP DO NOT HAVE THEIR PROPER FUNCTION, AND SHOULD BE
C       SPECIFIED AS NUMBERS GREATER THAN 1. FOR PROGRAM TO WORK.
C      J IS NUMBER OF ENGINE REVOLUTIONS REQUIRED. USUALLY 3
C       REVOLUTIONS ARE ENOUGH. IT IS LIKELY THAT AFTER MANY (EG. 10)
C       REVOLUTIONS THE ANSWERS WILL CEASE TO BE AS ACCURATE AS THE
C       ERRORS IN THE CALCULATION WOULD BE EXPECTED TO BE COMPOUNDED.
C      NNN IS NUMBER OF ENGINE REVOLUTIONS CURRENTLY PERFORMED.
C       SET TO 0 AT START.
C      NNZ IS NUMBER OF ENGINE REVOLUTIONS PERFORMED FROM THE START
C       BEFORE THE PROGRAM PLOTS OUT INFORMATION ON THE PLOTTER USING
C       AUTPLT.
C      TINT IS TIME INTERVAL BETWEEN WAVE REFLECTIONS AT PORT.
C      SXN IS CONSTANT TO CHANGE METRIC PRESSURE TO PSI.
C      AD IS AREA OF PIPE AT START OF EXHAUST DUCT IN MS*MS.
C      AE IS AREA OF TAILPIPE OUTLET. MS*MS.
C      AO IS REFERENCE SPEED OF SOUND. M/SEC.
C      VCO IS VOL. IN CYLINDER AT E.P.O. MS**3.
C      TCO IS TEMP. OF GAS IN CYLINDER AT E.P.O.
C      ZLNM IS NON-DIMENSIONALISING LENGTH IN METRES.
C      JJ IS OBSOLETE TERM. SET TO 0
C      DTT IS DIMENSIONAL FORM OF DT.
C      T IS TIME FROM E.P.O. SECS.
C      VSC IS CRANKCASE SWEPT VOL., M**3.
C      SCO IS INITIAL VALUE OF ENTROPY IN CYLINDER, IN DIMENSIONAL FORM.
C      SCT IS ENTROPY IN CYLINDER (INTENSIVE FORM).
C      SD IS NON DIMENSIONALISED REFERENCE ENTROPY VALUE, FOR PO, AO.
C      II IS THE LOCATION IN MATRIX A(NI) WHICH REFERS TO THE (R.H.) OPEN
C       TAILPIPE BOUNDARY.
C      II-1 REFERS TO THE PORT BOUNDARY OF THE EXHAUST PIPE.
C      PDT IS INITIAL PRESSURE IN EXHAUST SYSTEM WHERE GAS IS AT REST.
C       USUALLY ATMOSPHERIC PRESSURE, WHICH IS USUALLY THE SAME AS THE
C       REFERENCE PRESSURE. PASCALS.
C      PO IS REFERENCE PRESSURE IN PASCALS. USUALLY ATMOSPHERIC
C       PRESSURE.
C      ACOC IS VALUE OF CRANKCASE SPEED OF SOUND AT I.P.C. (INLET PORT
C       CLOSING) M/SEC.
C      PCOC IS VALUE OF CRANKCASE PRESSURE AT I.P.C. IN PASCALS.
C       USUALLY USE ATMOSPHERIC PRESSURE.
C      ZLNI IS EXHAUST SYSTEM LENGTH MS.
C
C-----------------------------------------------------------------------
C
       ONETHD = 0.3333333333333333
       READ(75,30)DTAU,RPMN,ACO,ZLNI,NWAVES
    30 FORMAT(4F10.5,I3)
       JREV = 60./( DTAU*RPMN ) + 3.
       JREX = 60.*ACO*FLOAT(NWAVES)/(RPMN*2.*ZLNI)
       JREX = JREX + 20
C
C-----------------------------------------------------------------------
C
C
C            PROGRAM INITIALIZES MANY VALUES FOR STPIPE.
C         MAY ALSO BE USED FOR WRITING VALUES ON PERMENANT FILE.
```

```
      C      VS IS SWEPT VOL. IN CUBIC METRES.
      C      BORE IS ENGINE CYLINDER BORE (DIAMETER). METRES.
      C      CONLEN IS CONROD LENGTH IN MS.
      C      X IS ANGLE FROM E.P.O. TO B.D.C. IN DEGREES.
      C      D1 IS DIAM. OF PIPE LEADING FROM ENGINE TO EXPANSION CONE,
      C       IN METRES.
      C      D2 IS DIAM. OF LARGE ENDS OF CONES, FOR STRAIGHT SECTION JOINING
      C       CONES, METRES.
      C      D3 IS TAILPIPE DIAM., METRES.
      C      XL1 IS LENGTH OF PIPE LEADING FROM ENGINE. METRES.
      C      XL2 IS LENGTH OF FIRST CONICAL SECTION OF TUNED EXPANSION CHAMBER.
      C      XL3 IS LENGTH OF STRAIGHT SECTION JOINING TWO LARGE ENDS OF CONES,
      C       METRES.
      C      XL4 IS LENGTH OF SECOND CONICAL SECTION.
      C      XL5 IS LENGTH OF TAILPIPE.
      C      II IS EQUAL TO 2 MORE THAN THE MAXIMUM NUMBER OF C+, C- AND P
      C       CHARACTERISTICS ALLOWED. II SETS LIMITS FOR ARRAY SPACES. THE
      C       ARRAY SPACES REQUIRED IN SUBROUTINE SPLMUF MUST CURRENTLY BE
      C       SPECIFIED SEPARATELY.
      C      GAMA IS SPECIFIC HEAT RATIO. (ONLY ONE VALUE USED).
      C      RGAS IS GAS CONSTANT, R, IN MKS UNITS.
      C      PCO IS PRESSURE IN CYLINDER AT E.P.O. IN PSI.
      C      TRDUC IS DISTANCE FROM EXHAUST PORT TO POSITION IN THE EXHAUST
      C       SYSTEM AT WHICH THE PRESSURE CYCLE IS TO BE KNOWN, I.E., LOCATION
      C       OF TRANSDUCER. METRES.
      C      MM IS NUMBER OF PARTS DTAU IS DIVIDED INTO. DTAU/MM IS THE BASIC
      C       TIME INCREMENT FOR THE CONSTRUCTION OF THE X-T DIAGRAM. IF THE
      C       CHARACTERISTICS CROWD TOO CLOSE TOGETHER AND THE PROGRAM DOES
      C       NOT WORK PROPERLY, THE PROGRAM MAY BE FIXED BY EITHER INCREASING
      C       MM OR BY REDUCING THE NUMBER OF CHARACTERISTICS. TYPICALLY MM=100
      C      BBQ IS THE FLOW LOSS FACTOR FOR FLOW THROUGH EXHAUST AND TRANSFER
      C       PORTS. TYPICALLY 0.8.
      C      XCI IS ANGLE FROM I.P.O.TO B.D.C., DEGREES.
      C      XC IS ANGLE FROM T.P.O. TO B.D.C. IN DEGREES.
      C      RC IS CRANKCASE COMPRESSION RATIO.
      C      BC IS TOTAL AREA OF FULLY OPEN TRANSFER PORTS, MS**2. SEE B.
      C      KJ IS NUMBER OF TIMES ITERATIONS WILL BE PERFORMED IN SUBROUTINES
      C       FLWOUT, FLOWIN. IF ANSWER ACHIEVES DESIRED ACCURACY BEFORE KJ
      C       ITERATIONS, THE CALCULATION IS CUT SHORT.
      C       TYPICALLY 28.
      C      NWVDIS IS NUMBER WHICH CONTROLS HOW CLOSE TOGETHER ADJACENT
      C       ENTROPY WAVES WILL BE BEFORE SUBROUTINE WVCLOS MERGES THEM
      C       TOGETHER. ADJACENT ENTROPY WAVES WILL BE MERGED IF THEY ARE
      C       CLOSER THAN THE NON-DIMENSIONAL DISTANCE NWVDIS*AO*DTAU/(MM*ZLNM)
      C      ND    WHEN PROGRAM IS SET TO PRINT OUT THE CURRENT WAVE DIAGRAM
      C       INFORMATION, WHICH IS DESIRABLE WHEN DIAGNOSING AN ERROR, THE
      C       PROGRAM BYPASSES THE NUMBER ND OF THE DTAU/MM PERIODS OF TIME
      C       BEFORE THE INFORMATION IS PRINTED OUT. THIS ELIMINATES WASTEFUL
      C       OUTPUT. SEE DTAU.
      C      TSTART IS TIME FROM E.P.O. AFTER WHICH SUBROUTINE PRSTOP WILL
      C       START TO PRINT OUT THE WAVE DIAGRAM (X-T DIAGRAM) INFORMATION.
      C       UNLESS INVESTIGATING AN ERROR, THIS VALUE IS SET VERY LARGE, SO
      C       THAT NOTHING IS PRINTED. AS PROGRAM IS AT PRESENT, PROGRAM
      C       PRINTS OUT AT TIME TSTART ON EVERY REVOLUTION, THAT IS, IT DOES
      C       NOT SPECIFICALLY PRINT OUT INFORMATION ONLY FOR THE THIRD
      C       REVOLUTION OR SUCH.
      C       SECS.
```

B1

```
     2          ,TSTOP
      COMMON /U/ U(100),UDLAST,UN(100),UX(800)
      COMMON /V/ VCT,VCTC,VCTCO,VCO,VCOC,VS,VSC
      COMMON /X/ X,XC,XCI,XE,XEC,XL1,XL3,XZ2,XZ4,X1,X2,X3,X4
      COMMON /Z/ ZB,ZI(100),ZIN(100),ZLN,ZLNM
C
C-----------------------------------------------------------------------
C
C        SET UP DATA.
C
C     DTAU IS BASIC TIME INCREMENT IN SECONDS. THE PROGRAM PLOTS AN X-T
C      DIAGRAM BY SPECIFYING THE POSITION OF THE C+, C- AND P
C      CHARACTERISTICS ALONG THE DUCT AT SUCCESSIVE INSTANTS OF TIME.
C      THIS TIME INTERVAL IS DTAU/MM. THE PROGRAM CALCULATES THE
C      PRESSURE AT A CERTAIN POINT IN THE EXHAUST SYSTEM AT INTERVALS
C      DTAU. ALSO THE PROGRAM PRINTS OUT THE CYLINDER AND CRANKCASE
C      PRESSURE AT THE SAME TIME AS IT CALCULATES THE TRANSDUCER
C      PRESSURE.
C      TYPICALLY 10.**(-4.) SECS. (FOR DTAU).
C      IF DATU IS TOO SMALL, THE ARRAY SPACE FOR SOME VARIABLES MUST BE
C      INCREASED.
C     RPMN IS ENGINE SPEED IN REVOLUTIONS PER MINUTE. THIS VALUE IS THE
C      SAME FOR ALL SUCCESSIVE REVOLUTIONS.
C     ACO IS SPEED OF SOUND IN CYLINDER AT EXHAUST PORT OPENING (E.P.O.)
C      IN M/SEC. THIS VALUE IS THE SAME FOR ALL SUCCESSIVE REVOLUTIONS.
C     ZLNI IS LENGTH OF WHOLE EXHAUST SYSTEM FROM THE EXHAUST PORT TO
C      THE TAILPIPE OUTLET. METRES.
C     NWAVES IS NUMBER OF C+ AND C- CHARACTERISTICS, COMBINED, AT START
C      OF CALCULATION IN THE EXHAUST SYSTEM. THESE CHARACTERISTICS ARE
C      EVENLY SPACED INITIALLY WITH AN EQUAL NUMBER OF C+ AND C-.
C      TYPICALLY 60 MAY BE USED FOR GOOD ACCURACY. IF MANY MORE ARE
C      REQUIRED, THE ARRAY SPACE FOR THE VALUES ASSOCIATED WITH THE
C      CHARACTERISTICS MUST BE INCREASED. AS C+, C- AND P
C      CHARACTERISTICS ARE ALL CONTAINED IN THE SAME ARRAYS IT MAY
C      NEVER BE PRECISELY KNOWN HOW MANY ARRAY SPACES ARE REQUIRED,
C      BEFORE A TEST CALCULATION IS PERFORMED, ALTHOUGH WITH 60 C+ AND
C      C- CHARACTERISTICS, 100 SPACES SHOULD BE ENOUGH.
C     JREV, WHICH IS 3 LARGER THAN NREV IN STPIPE, IS THE SIZE OF
C      VARIOUS ARRAYS IN STPIPE.
C     JREX, WHICH IS SLIGHTLY GREATER THAN THE NUMBER OF TIMES A P WAVE
C      REACHES THE OPEN PIPE END IN ONE ENGINE REVOLUTION (SO LONG AS
C      THE VALUE OF AVERAGE WAVE SPEED IN THE PIPE IS LESS THAN ACO),
C      WILL SET THE SIZE OF SOME ARRAYS IN STPIPE.
C      JREX WILL BE INCREASED BY 20 FOR SAFETY.
C     IJ IS NUMBER OF HARMONICS REQUIRED IN THE FOURIER ANALYSIS OF
C      THE RADIATED SOUND PRESSURE CYCLES, AND THE CYCLES OF PRESSURE
C      VALUES AT A CERTAIN POINT IN THE EXHAUST DUCT. USUALLY LESS THAN
C      100. MUST BE LESS THAN 150.
C     DIST IS THE DISTANCE FROM THE EXHAUST TAILPIPE OUTLET TO THE
C      MEASURING POSITION, AT WHICH PLACE THE RADIATED SPL IS REQUIRED.
C      DIST IS IN METRES.
C     STROKE IS ENGINE STROKE IN MS.
C     B IS TOTAL AREA OF FULLY OPEN EXHAUST PORT. IT IS ASSUMED THAT
C      THE PORT IS RECTANGULAR AND THAT IT IS FULLY OPEN AT B.D.C.
C      (METRES)
C     R IS COMPRESSION RATIO OF ENGINE, THAT IS, TOTAL CYLINDER VOLUME
C      AT B.D.C. DIVIDED BY CLEARANCE VOLUME AT T.D.C.
```