```
c ******************************************************************
c
c     ORTEP-III: Oak Ridge Thermal Ellipsoid Plot Program
c     Carroll K. Johnson and Michael N. Burnett
c     Oak Ridge National Laboratory
c     Version 1.0 April 1, 1996
c
c     Send comments, questions, problems, suggestions, etc. to
c     ortep@ornl.gov
c
c ******************************************************************

      PROGRAM ORTEP
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      common /ns/ npf,ndraw,norient,nvar
      logical tmpopn

c *** Drawing Output Options
c *** ndraw=0: no drawing output
c *** ndraw=1: screen output
c *** ndraw=2: Postscript file output
c *** ndraw=3: HPGL file output

c *** Logical Unit Numbers ***
c *** 15,16 are used in subroutine EDITR
c *** 18 (variable iu) is used in subroutine PRELIM
c *** NOUT is set in subroutine UINPUT
      IN=3
      NED=7
      NSR=8
      NPF=10

      call uinput(in,nout)

    2 CALL PRIME

c *** open ORTEP scratch file ***
c *** if already open, close it first ***
      inquire(NSR,opened=tmpopn)
      if (tmpopn) close(NSR)
      open(NSR,status='scratch',form='unformatted')

c *** open a temporary file - needed by the editor ***
c *** if already open, close it first ***
      inquire(NED,opened=tmpopn)
      if (tmpopn) close(NED)
      open(NED,status='scratch')

C     ***** READ JOB TITLE CARD *****
      READ (IN,4)(TITLE(I),I=1,18)
    4 FORMAT(18A4)
      write (NED,4)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(I),I=1,18)
    5 FORMAT(1H0,10X,18A4)
    6 FORMAT(1H1,10X,18A4)
      CALL PRELIM
      IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(I),I=1,18)
C     ***** LOAD INSTRUCTION QUE *****
      NQUE=0
 2005 NQUE=NQUE+1
 2010 READ (IN,2012,END=2015,ERR=3000) QUE(NQUE)
      if (que(nque)(1:1).eq.'#') go to 2010
      hque(nque)=que(nque)
      if (que(nque)(4:9).eq.'    -2') go to 2020
 2012 FORMAT(A72)
      IF(NQUE.LT.96) GO TO 2005
      GO TO 2020
 2015 NQUE=NQUE-1
C     ***** REPOSITION TO POINT BEFORE EOF *****
      BACKSPACE IN
 2020 NBACK=NQUE
      NEXT=1
      ISAVE=0
      GO TO 507
    7 ISAVE=0
C     ***** ZERO AIN ARRAY *****
    8 DO 10 J=1,140
   10 AIN(J)=0.
   11 FORMAT(1H0,4X,17H((((( INSTRUCTION,I5,6H ))))))
   12 FORMAT(I3,I6,7F9.0)
   13 FORMAT(I3,I6,7D15.8)
   14 FORMAT(1H ,9X,7D15.7)
C     ***** READ NEW INSTRUCTION CARD *****
      NCD=0
      N1=-6
   16 N1=N1+7
      N2=N1+6
      IF(ISAVE)22,18,18
   18 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,12)IIC,NF,(AIN(I),I=N1,N2)
      IF(ISAVE)24,24,20
   20 WRITE (NSR)IIC,NF,(AIN(I),I=N1,N2)
      GO TO 24
   22 READ (NSR)IIC,NF,(AIN(I),I=N1,N2)
      IF(IIC)7,24,24
   24 IF(N1-1)26,26,30
   26 IF (NOUT.GE.0)
     &WRITE (NOUT,11)NF
      NF1=NF
      IF(NF1)28,8,30
c *** run editor?
   28 call go2edtr
      if (next.lt.nque) go to 8
```

```
      IF(NF1+2)2,2,3000
   30 CONTINUE
CCC   IF (NOUT.GE.0)
CCC  &WRITE (NOUT,14)(AIN(I),I=N1,N2)
   32 IIC=IIC+1
      GO TO (90,16,38,50),IIC
   33 FORMAT(I3,6X,5I3,8F6.0)
   34 FORMAT(6I3,8E12.5)
   35 FORMAT(1H ,11X,5I3,8F11.5)
C     ***** READ FORMAT 2 TRAILER CARDS *****
   38 NCD=NCD+1
      IF(ISAVE)44,40,40
   40 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,33)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      IF(ISAVE)46,46,42
   42 WRITE (NSR)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      GO TO 46
   44 READ (NSR)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
   46 IF (NOUT.GE.0)
     &WRITE (NOUT,35)(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      GO TO 32
C     ***** READ FORMAT 3 TRAILER CARD *****
   50 IF(ISAVE)52,54,54
   52 READ (NSR)(TITLE2(I),I=1,18)
      GO TO 55
   54 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,4)(TITLE2(I),I=1,18)
   55 IF (NOUT.GE.0)
     &WRITE (NOUT,5)(TITLE2(I),I=1,18)
      IF(ISAVE)90,90,56
   56 WRITE (NSR)(TITLE2(I),I=1,18)
C     ***** EXECUTE INSTRUCTION *****
   90 NJ=NF1/100
      NJ2=NF1-NJ*100
      NJ3=MOD(NJ2,10)
      IF(NJ-12)98,92,92
   92 CALL SPARE(NF1)
      IF(NG)94,8,94
   94 CALL ERPNT(0.D0,NF1)
      GO TO 8
C     ******BRANCH TABLE FOR FUNCTION TYPES******
   98 GO TO(100,200,300,400,500,600,700,800,900,1000,1100),NJ
C     *******100 INSTRUCTIONS-STRUCTURE ANALYSIS FUNCTIONS*******
  100 GO TO (101,101,104,104,101,101,94),NJ2
  101 CALL SEARC
      GO TO 8
C     ***** ANISOTROPIC TEMP FACTOR OUTPUT *****
  104 DO 164 I=1,NATOM
      IF(MOD(I,14)-1)134,114,134
  114 IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,129)
  129 FORMAT(1H010X,4HATOM3X,16HRMS DISPLACEMENT3X,31HROW VECTORS, BASED
     1 ON REFERENCE17X,29HPROBABILITY COVARIANCE MATRIX)
  134 TD=55501.+FLOAT(I)*100000.
      CALL PAXES(TD,-3)
      IF(NG)144,154,144
  144 CALL ERPNT(TD,104)
```

```
  149 FORMAT(1H0,10X,A6,F10.6,6X,3F12.7,10X,3F12.7)
  154 IF (NOUT.GE.0)
     &WRITE (NOUT,149)CHEM(I),RMS(1),(PAC(J,1),J=1,3),(Q(J,1)
     1,J=1,3)
  164 IF (NOUT.GE.0)
     &WRITE (NOUT,159)(RMS(K),(PAC(J,K),J=1,3),(Q(J,K),J=1,3)
     1,K=2,3)
  159 FORMAT(1H ,16X,F10.6,6X,3F12.7,10X,3F12.7)
      GO TO 8
C     *******200 INSTRUCTIONS-PLOTTER CONTROL FUNCTIONS*******
  200 CALL F200
      GO TO 8
C     *******300 INSTRUCTIONS-DRAWING CONTROL FUNCTIONS*******
  300 GO TO (301,302,303,304,94),NJ2
C     *******PLOT DIMENSIONS*******
  301 IF(AIN(1))321,321,311
  311 XLNG(1)=AIN(1)
  321 IF(AIN(2))341,341,331
  331 XLNG(2)=AIN(2)
  341 IF(AIN(3))361,351,351
  351 VIEW=AIN(3)
  361 IF(AIN(4))381,381,371
  371 BRDR=AIN(4)
  381 IF (NOUT.GE.0)
     &WRITE (NOUT,389)XLNG(1),XLNG(2),BRDR
  389 FORMAT(1H010X,11HPLOT LIMITSF6.2,3H BYF6.2,15H IN.  INCLUDINGF6.2,
     112H  IN. MARGIN)
  391 IF (NOUT.GE.0)
     &WRITE (NOUT,399)VIEW
  399 FORMAT(1H ,10X,13HVIEW DISTANCE,F7.3,7H INCHES)
      GO TO 8
C     *******LEGEND ROTATION*******
  302 THETA=AIN(1)
      T1=THETA*.01745329252
      COSTH=COS(T1)
      SINTH=SIN(T1)
      DO 312 J=1,9
  312 SYMB(J,1)=0.
      SYMB(1,1)=COSTH
      SYMB(2,2)=COSTH
      SYMB(3,3)=1.
      SYMB(2,1)=SINTH
      SYMB(1,2)=-SINTH
      IF (NOUT.GE.0)
     &WRITE (NOUT,319)THETA
  319 FORMAT(1H010X,44HREGULAR TITLE AND SYMBOL ROTATION IN DEGREESF8.2)
      GO TO 8
C     ***** RETRACE DISPLACEMENT *****
  303 DISP=AIN(1)
      IF (NOUT.GE.0)
     &WRITE (NOUT,313)DISP
  313 FORMAT(1H0,10X,22HRETRACE DISPLACEMENT =,F7.4,5H INCH)
      GO TO 8
C     ***** change resolution (smoothness) of ellipses *****
  304 res(1)=AIN(1)*.75
      res(2)=.5*res(1)
      res(3)=.25*res(2)
      GO TO 8
C     *******400 INSTRUCTIONS-ATOM LIST FUNCTIONS*******
  400 GO TO (401,401,401,401,401,401,401,490, 94,410,
     1       401,401,401,401,401,401,401, 94),NJ2
```

```
  401 CALL F400
      GO TO 490
  410 LATM=0
      DO 420 I=1,500
      ATOMID(I)=0.
      DO 420 J=1,3
  420 ATOMS(J,I)=0.
  490 IF(LATM)8,8,491
  491 IF (NOUT.GE.0)
     &WRITE (NOUT,499)(ATOMID(I),I=1,LATM)
  499 FORMAT(1H0,10X,23HCONTENTS OF ATOMS ARRAY/(15X,10F10.0))
      GO TO 8
C     *******500 INSTRUCTIONS-CARTESIAN COORDINATE SYSTEM FUNCTIONS*******
  500 CALL F500
      IF (NOUT.GE.0)
     &WRITE  (NOUT,503)(ORGN(J),J=1,3)
  503 FORMAT(1H0,10X,    44HORIGIN FOR PROJECTION AXIS IN CRYSTAL COORD.
     1,3F15.6)
      IF(NJ3-3)507,539,504
  504 IF(NJ3-6)601,507,601
  507 IF (NOUT.GE.0)
     &WRITE (NOUT,529)
      IF (NOUT.GE.0)
     &WRITE (NOUT,519)((REFV(J,I),I=1,3),(AAREV(J,I),I=1,3),J
     1=1,3)
      GO TO 8
  509 FORMAT(1H010X, 49HORTHONORMAL WORKING VECTORS BASED ON CRYSTAL AXE
     1S18X,33HPOST-FACTOR TRANSFORMATION MATRIX/16X,8HX VECTOR8X,8HY VEC
     2TOR8X,8HZ VECTOR)
  519 FORMAT(1H ,10X,3E16.7,8X,3E16.7)
  529 FORMAT(1H010X,51HORTHONORMAL REFERENCE VECTORS BASED ON CRYSTAL AX
     1ES16X,33HPOST-FACTOR TRANSFORMATION MATRIX/16X,8HX VECTOR8X,8HY VE
     2CTOR8X,8HZ VECTOR)
  539 IF (NOUT.GE.0)
     &WRITE (NOUT,509)
      IF (NOUT.GE.0)
     &WRITE (NOUT,519)((WRKV(J,I),I=1,3),(AAWRK(J,I),I=1,3),J
     1=1,3)
      GO TO 8
C     *******600 INSTRUCTIONS-PLOT CENTERING FUNCTIONS*******
  600 CALL F600
  601 IF (NOUT.GE.0)
     &WRITE (NOUT,609)XO(1),XO(2),SCAL1,SCAL2
  609 FORMAT(1H010X,31HORIGIN POINT IN PLOTTER COORD.(F6.2,2H ,F6.2,8H )
     1 IN.  / 11X,15HOVERALL SCALE =F6.3,32H INCH/ANGSTROM ELLIPSOID SCA
     2LE =F6.3)
      GO TO 391
C     *******700 INSTRUCTIONS-ELLIPSOID AND SYMBOL PLOT FUNCTIONS*******
C     ********FILL OUT DETAILS FOR SPECIAL MODELS********
  700 GO TO (701,702,704,705,709,7006,94),NJ3
 7006 AIN(3)=1.
      GO TO 703
  701 AIN(3)=8.
      GO TO 703
  702 AIN(3)=0.
  703 AIN(1)=4.
      AIN(2)=0.
      AIN(4)=0.
      GO TO 709
  704 AIN(1)=3.
      AIN(2)=-5.
      GO TO 706
  705 AIN(1)=1.
      AIN(2)=0.
  706 AIN(3)=1.
      AIN(4)=5.
  709 CALL F700
      GO TO 8
C     *******800 INSTRUCTIONS-BOND FUNCTIONS*******
  800 CALL F800
      GO TO 8
C     *******900 INSTRUCTIONS-TITLE FUNCTIONS*******
  900 CALL F900
      GO TO 8
C     *******1000 INSTRUCTIONS-OVERLAP FUNCTIONS*******
 1000 CALL F1000
      GO TO 8
C     *******1100 INSTRUCTIONS-SAVE SEQUENCE FUNCTIONS*******
 1100 IF(NJ2-2)1101,1102,1103
 1101 ISAVE=1
      GO TO 1104
 1102 ISAVE=0
      J=-1
CCC   END FILE NSR
      GO TO 1104
 1103 ISAVE=-1
 1104 REWIND NSR
      GO TO 8
 3000 CALL EXITNG(NG)
      END
      FUNCTION ARCCOS(X)
C     ARCCOS(X) IN DEGREES
      IF(1.0-ABS(X))1,2,2
    1 X=SIGN(1.0,X)
    2 IF(X)3,4,5
    3 ARCCOS=180.0+ATAN(SQRT(1.0-X*X)/X)*57.29577951
      GO TO 6
    4 ARCCOS=90.0
      GO TO 6
    5 ARCCOS=ATAN (SQRT(1.0-X*X)/X)*57.29577951
    6 RETURN
      END
      SUBROUTINE ATOM(QA,Z)
C     ATOM COORDINATE SUBROUTINE
      REAL*8 QA,TA,D100K
      DIMENSION X(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100K=100000.0
      K=QA/D100K
```

```
      IF(K)109,109,117
  109 X(1)=0.0
      X(2)=0.0
      X(3)=0.0
      GO TO 125
  117 IF(K-NATOM)119,119,503
  503 NG=5
      GO TO 325
  119 DO 123 J=1,3
  123 X(J)=P(J,K)
  125 TA=DABS(QA)
      KSYM=DMOD(TA,D100K)
      KT=KSYM/100
      KS=KSYM-100*KT
      IF(KS-NSYM)203,203,403
  403 NG=4
      GO TO 325
  203 IF(KS)403,205,213
  205 Z(1)=X(1)
      Z(2)=X(2)
      Z(3)=X(3)
      GO TO 311
  213 DO 223 K=1,3
      Z(K)=TS(K,KS)
      DO 223 J=1,3
  223 Z(K)=Z(K)+FS(J,K,KS)*X(J)
  311 IF(KT)403,325,313
  313 IF(KT-555)317,315,317
  315 KSYM=KS
      GO TO 325
  317 K1=KT/100
      K=KT-100*K1
      K2=K/10
      K3=K-10*K2
      Z(1)=Z(1)+FLOAT(K1-5)
      Z(2)=Z(2)+FLOAT(K2-5)
      Z(3)=Z(3)+FLOAT(K3-5)
  325 RETURN
      END
      SUBROUTINE AXEQB(A1,X,B1,JJJ)
C     ***** SOLUTION OF MATRIX EQUATION AX=B FOR X *****
C     ***** USES METHOD OF TRIANGULAR ELIMINATION *****
C     ***** B AND X HAVE DIMENSIONS (3,JJJ),A IS ALWAYS (3,3)
C     ***** TO INVERT A MAKE B 3 BY 3 IDENITY MATRIX *****
      DIMENSION A1(3,3),A(3,3),B(3,3),B1(3,3),X(3,3)
      NV=JJJ
C     ***** TRANSFER DATA *****
      DO 2 I=1,3
      DO 2 J=1,3
      IF(NV-J)2,1,1
    1 B(I,J)=B1(I,J)
    2 A(I,J)=A1(I,J)
C     ***** TRIANGULARIZE MATRIX A *****
      DO 17 I=1,2
      S=0.0
      DO 4 J=I,3
      R=ABS(A(J,I))
      IF(R-S)4,3,3
    3 S=R
      L=J
    4 CONTINUE
      IF(L-I)5,10,5
    5 DO 6 J=I,3
      S=A(I,J)
      A(I,J)=A(L,J)
    6 A(L,J)=S
      DO 8 J=1,NV
      S=B(I,J)
      B(I,J)=B(L,J)
    8 B(L,J)=S
   10 TEM=A(I,I)
      IF(TEM)11,17,11
   11 IPO=I+1
      DO 16 J=IPO,3
      IF(A(J,I))12,16,12
   12 S=A(J,I)/TEM
      A(J,I)=0.0
      DO 13 K=IPO,3
   13 A(J,K)=A(J,K)-A(I,K)*S
      DO 15 K=1,NV
   15 B(J,K)=B(J,K)-B(I,K)*S
   16 CONTINUE
   17 CONTINUE
C     ***** MODIFY SINGULAR MATRIX *****
      DO 20 I=1,3
      IF(A(I,I))20,19,20
   19 A(I,I)=AMAX1(1.E-25,AMAX1(A(1,1),A(2,2),A(3,3))*1.E-15)
   20 CONTINUE
      DO 24 K=1,NV
      DO 24 I=1,3
      N=4-I
      M=N+1
      TEM=B(N,K)
      IF(3-M)23,21,21
   21 DO 22 J=M,3
   22 TEM=TEM-A(N,J)*B(J,K)
   23 B(N,K)=TEM/A(N,N)
   24 X(N,K)=B(N,K)
      RETURN
      END
      SUBROUTINE AXES(U,V,X,ITYPE)
C     ***** STORE THREE ORTHOGONAL VECTORS EACH 1 ANGSTROM LONG *****
C     ***** ITYPE .GT.0 FOR CARTESIAN,.LE.0 FOR TRICLINIC *****
C     *****IABS(ITYPE)=1 W(1)=U,W(2)=(UXV),W(3)=UX(UXV) *****
C     *****IABS(ITYPE)=2 W(1)=U,W(2)=(UXV)XU,W(3)=(UXV) *****
C     ***** ITYPE=0 W(1)=A,W(2)=(AXB)XA,W(3)=(AXB), ABC=CELL VECTORS ***
      DIMENSION U(3),V(3),W(3,3),X(3,3)
      IT=ITYPE
      IF(IT)115,105,115
  105 U(1)=1.
      U(2)=0.
      U(3)=0.
      V(1)=0.
      V(2)=1.
      V(3)=0.
  115 DO 125 J=1,3
  125 W(J,1)=U(J)
      IF(IABS(IT)-1)145,135,145
  135 CALL NORM(U,V,W(1,2),IT)
      CALL NORM(U,W(1,2),W(1,3),IT)
      GO TO 155
  145 CALL NORM(U,V,W(1,3),IT)
```

```
      CALL NORM(W(1,3),U,W(1,2),IT)
  155 DO 195 I=1,3
      IF(IT)165,165,175
  165 IC=-1
      GO TO 195
  175 IC=1
  195 CALL UNITY(W(1,I),X(1,I),IC)
      RETURN
      END
      SUBROUTINE BOND(Z1,Z2,NB,NA1,NA2)
      REAL*8 Z1,Z2,WD(2),TD,D100,D1000,D100K
      DIMENSION B1(3,3),E(3,3),S(3,3),U(3,3),VUE(3)
      DIMENSION V7(3),W(13,2),Z(3),RESB(2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
C     ***** OBTAIN POSITIONAL PARAMETERS *****
      DATA RESB/.2,.08/
      D100=100.
      D1000=1000.
      D100K=100000.
      NG1=0
      DO 105 J=1,26
  105 W(J,1)=0.
      WD(1)=Z1
      WD(2)=Z2
      DO 135 I=1,2
      CALL XYZ(WD(I),W(4,I),2)
      IF(NG)125,110,125
  110 DO 115 J=1,3
  115 W(J+6,I)=XT(J)
      K=WD(I)/D100K
      L=DMOD(WD(I)/D100,D1000)
      L1=DMOD(WD(I),D100)
      CALL PLTXY(W(4,I),W(2,I))
      IF(EDGE-BRDR*.25)120,128,128
  120 NG=10
  125 NG1=1
      IF (NOUT.GE.0)
     &WRITE(NOUT,136)CHEM(K),K,L,L1,(W(J,I),J=2,9)
      CALL ERPNT(WD(I),800)
      GO TO 134
  128 IF(NJ2-10)130,134,134
  130 IF (NOUT.GE.0)
     &WRITE(NOUT,136)CHEM(K),K,L,L1,(W(J,I),J=2,9)
  134 continue
  135 CONTINUE
  136 FORMAT(1H 10X,A6,3H  (I3,1H,I3,I2,4H)   2F8.2,5X,3F8.3,13X,3F8.4)
      IF(NG1)999,137,999
  137 CALL DIFV(W(7,1),W(7,2),V7)
      DIST=SQRT(VMV(V7,AA,V7))
      IF(MOD(NJ2,2).EQ.0) GO TO 143
      IF(MOD(NJ2,10).EQ.1) GO TO 143
C     ***** LINE BONDS AND CENTERED SYMBOLS (803,813) *
      HGT=SCL*.12
c *** ORTEP-II calls
c     CALL SIMBOL(W(2,1),W(3,1),HGT,MOD(IDINT(WD(1)/D100K),10),0.,-
c    11)
c     CALL SIMBOL(W(2,2),W(3,2),HGT,MOD(IDINT(WD(2)/D100K),10),0.,-
c    12)
c *** Only one centered symbol (*) is available in ORTEP-III.
c *** It is triggered by the negative value for argument 6.
c *** Argument 4 is ignored by SIMBOL.
      CALL SIMBOL(W(2,1),W(3,1),HGT,' ',0.,-1)
      CALL SIMBOL(W(2,2),W(3,2),HGT,' ',0.,-2)
      GO TO 570
C     ***** STICK BONDS FOR 801,802,811,812 *****
  143 KODE=KD(5,NB)
      IF(KODE)145,144,146
  144 NBND=0
      GO TO 148
  145 KODE=-KODE
  146 NBND=128/2**KODE
C     ***** FIND UPPERMOST ATOM PUT IN POSITION ONE *****
  148 IF(VIEW)152,150,152
  150 W(12,1)=1.
      W(12,2)=1.
      IF(W(6,1)-W(6,2))165,175,175
C     *****VECTOR FROM ATOM TO VIEWPOINT *****
  152 DO 160 I=1,2
      DO 155 J=10,12
  155 W(J,I)=-W(J-6,I)
      W(12,I)=W(12,I)+VIEW
C     ***** DISTANCE SQUARED TO VIEWPOINT *****
  160 W(13,I)=VV(W(10,I),W(10,I))
      IF(W(13,2)-W(13,1))165,175,175
C     ***** SWITCH ATOMS *****
  165 DO 170 J=1,13
      T1=W(J,1)
      W(J,1)=W(J,2)
  170 W(J,2)=T1
      TD=WD(1)
      WD(1)=WD(2)
      WD(2)=TD
C     ***** FORM IDEMFACTOR MATRIX *****
  175 DO 180 J=1,3
      E(J,J)=1.
      E(J+1,1)=0.
  180 E(J+5,1)=0.
C     ***** FORM VECTOR SET RADIAL TO BOND *****
      CALL DIFV(W(4,2),W(4,1),DA(1,3))
      CALL UNITY(DA(1,3),V3,1)
C     ***** UNIT VECTOR FROM BOND MIDPOINT TO REFERENCE VIEWPOINT *****
      DO 183 I=1,3
      V2(I)=0.0
      DO 181 J=1,3
  181 V2(I)=V2(I)+AAREV(J,3)*WRKV(J,I)
      IF(VIEW)183,183,182
  182 V2(I)=V2(I)*VIEW-0.5*(W(I+3,1)+W(I+3,2))
  183 CONTINUE
      CALL UNITY(V2,V2,1)
```

```
      T6=ABS(VV(V3,V2))
      IF(.9994-T6)185,185,187
C     ***** ALTERNATE CALC IF BOND IS ALONG REFERENCE VIEW DIRECTION ***
  185 DO 186 J=1,3
  186 V2(J)=W(J+9,1)+W(J+9,2)
      CALL UNITY(V2,V2,1)
      T6=ABS(VV(V3,V2))
      IF(.9994-T6)390,390,187
  187 CALL AXES(V3,V2,B1,1)
  188 T1=CD(3,NB)/SCAL2
      DO 190 J=1,3
      DA(J,1)=-B1(J,2)*T1
  190 DA(J,2)=-B1(J,3)*T1
      IF(NBND)500,500,195
C     ***** SET PLOTTING RESOLUTION FOR BOND *****
  195 T1=CD(3,NB)*SCL
      NRESOL=4
      NBIS=3
      DO 200 J=1,2
      IF(T1.GE.RESB(J)) GO TO 202
      IF(NBND.LE.NRESOL) GO TO 202
      NBIS=NBIS-1
  200 NRESOL=NRESOL*2
  202 NRES1=NRESOL+1
      CALL RADIAL(NBIS)
C     ***** DERIVE QUADRICS FOR EACH ATOM *****
      DO 380 II=1,2
      CALL PAXES(WD(II),2)
      IF(NG)205,210,205
  205 CALL ERPNT(WD(II),800)
      GO TO 999
C     ***** DOES BOND GO TO ELLIPSOID OR TO ENVELOPE *****
  210 T1=3-II*2
      DO 212 J=1,3
      V3(J)=V3(J)*T1
  212 VUE(J)=0.
      IF(KD(5,NB))260,260,215
  215 IF(VMV(V3,Q,W(10,II)))220,260,260
  220 IBND=0
      IF(VIEW)240,240,225
C     ***** DERIVE TANGENT CONE DIRECTLY WITHOUT ROTATING COORDINATES **
  225 T2=-(SCAL2*RMS(1)*RMS(2)*RMS(3))**2
      DO 230 J=1,3
      V1(J)=-W(J+9,II)/SCAL1
      VUE(J)=V1(J)/SCAL2
C     ***** INVERT ELLIPSOID MATRIX *****
      DO 230 K=J,3
      T1=0.0
      DO 228 I=1,3
  228 T1=T1+PAC(J,I)*PAC(K,I)*RMS(I)**2
      U(J,K)=T1
  230 U(K,J)=T1
C     *****  ADD POLARIZED COFACTOR MATRIX TO ELLIPSOID MATRIX *****
      DO 235 J=1,3
      J1=MOD(J,3)+1
      VJ1=V1(J1)
      J2=MOD(J+1,3)+1
      VJ2=V1(J2)
      DO 235 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
```

```
      S(J,K)=T2*Q(J,K)+(VJ2*(U(J1,K1)*V1(K2)-U(J1,K2)*V1(K1))
     1                + VJ1*(U(J2,K2)*V1(K1)-U(J2,K1)*V1(K2)))
  235 S(K,J)=S(J,K)
      T5=0.0
      GO TO 300
C     ***** DERIVE TANGENT CYLINDER WITH AXIS ALONG Z *****
  240 T1=-1.0/Q(3,3)
      DO 250 J=1,2
      DO 245 K=1,2
  245 S(K,J)=Q(K,J)+Q(K,3)*Q(J,3)*T1
      S(3,J)=0.0
  250 S(J,3)=0.0
      S(3,3)=0.0
      GO TO 270
C     ***** TRANSFER ELLIPSOID *****
  260 DO 265 J=1,9
  265 S(J,1)=Q(J,1)
      IBND=II
  270 T5=1.
C     ***** CHECK FOR BOND TAPER *****
  300 IF(II-2)305,310,310
  305 RADIUS=1.+T6*TAPER
      GO TO 320
  310 RADIUS=1.-T6*TAPER
  320 CALL MV(S,V3,V4)
      T2=VV(V3,V4)
C     ***** COMPUTE BOND INTERSECTION *****
      KL=5-II-II
      KSTP=NRESOL
      IF(NJ2-21)324,322,322
  322 KSTP=32
  324 DO 335 K=1,65,KSTP
      DO 325 J=1,3
      V6(J)=D(J,K)*RADIUS
  325 V5(J)=V6(J)+VUE(J)
      T3=VV(V5,V4)
      T4=T3*T3-T2*(VMV(V5,S,V5)-T5)
      IF(T4)345,330,330
  330 T4=SQRT(T4)
      T1=(T4-T3)/T2
      T3=(-T4-T3)/T2
      L=K+KL-1
      DO 335 J=1,3
      D(J,L)=(V6(J)+T1*V3(J))*SCL
  335 D(J,L+1)=(-V6(J)-T3*V3(J))*SCL
      IF(IBND+21-NJ2)360,338,360
  338 IF(KD(5,NB))360,360,340
C     ***** FOR LOCAL OVERLAP, MAKE BOND QUADRANGLE TANGENT TO ENVELOPING CONE
  340 T3=VV(VUE,V4)
      T4=T3**2-T2*(VMV(VUE,S,VUE)-T5)
      IF(T4)345,350,350
  345 NG=13
      CALL ERPNT(WD(II),800)
      GO TO 999
  350 T1=(SQRT(T4)-T3)/T2
      DO 355 J=1,3
      T4=(T1*V3(J)*SCL-0.5*(D(J,KL)+D(J,KL+64)))*1.001
      D(J,KL)=D(J,KL)+T4
  355 D(J,KL+64)=D(J,KL+64)+T4
  360 CALL PROJ(D(1,KL),DP(1,II),W(4,II),XO,VIEW,1,65,KSTP)
      IF(IBND-1)370,365,370
```

```
  365 CALL PROJ(D(1,KL+KSTP+1),DP(1,II+64+KSTP),W(4,II),XO,VIEW,1,
     & 65-KSTP,KSTP)
      GO TO 380
C     ***** RETRACE TOP HALF *****
  370 KK=64-(II-1)*KSTP
      DO 375 K=KSTP,KK,KSTP
      L=K+II
      M=L+64
      N=66-L
      DP(1,M)=DP(1,N)
  375 DP(2,M)=DP(2,N)
  380 CONTINUE
C     ***** CHECK FOR LOCAL OVERLAP OR HIDDEN BOND *****
      DO 395 K=1,65,32
      T1=0.
      T2=0.
      DO 385 J=1,2
      T1=T1+(DP(J,K)-W(J+1,1))**2
  385 T2=T2+(DP(J,K+1)-W(J+1,1))**2
      IF(T2-T1)390,390,395
  395 CONTINUE
C     ***** CALL GLOBAL OVERLAP ROUTINE *****
      ICQ=0
      CALL LAP800(NA1,NA2,ICQ)
      IF(NJ2-21)400,999,999
  400 IF(ICQ)390,405,405
C     ***** DRAW BOND OUTLINE *****
  405 CALL DRAW(DP(1,1),0.,0.,3)
      DO 415 K=NRES1,129,NRESOL
  415 CALL DRAW(DP(1,K),0.,0.,2)
      DO 420 K=2,66,NRESOL
  420 CALL DRAW(DP(1,K),0.,0.,2)
      CALL DRAW(DP(1,65),0.,0.,2)
C     ***** DRAW BOND DETAIL *****
  425 K=65
  430 K=K-NBND
      IF(K-1)500,500,435
  435 CALL DRAW(DP(1,K),0.,0.,3)
      CALL DRAW(DP(1,K+1),0.,0.,2)
      K=K-NBND
      IF(K-1)500,500,440
  440 CALL DRAW(DP(1,K+1),0.,0.,3)
      CALL DRAW(DP(1,K),0.,0.,2)
      GO TO 430
  500 HGT=CD(4,NB)
      OFF=CD(5,NB)
      IF(HGT)570,570,510
C     ***** PERSPECTIVE BOND LABEL ROUTINE *****
C     ***** BASE DECISIONS ON REFERENCE SYSTEM *****
  510 K=0
      CALL DIFV(W(7,2),W(7,1),V7)
      CALL VM(V7,AAREV,V1)
      CALL AXES(V1,E(1,3),U,1)
      DO 535 I=1,3
      T1=1.
      IF(I-2)515,515,520
  515 IF(VV(U(1,I),SYMB(1,I)))525,530,530
  520 IF(MOD(K,2))530,525,530
  525 T1=-1.
      K=K+1
  530 DO 535 J=1,3
      U(J,I)=U(J,I)*T1
  535 VT(J,I)=B1(J,I)*T1
      DO 540 J=1,3
  540 VT(J,4)=.5*(W(J+3,1)+W(J+3,2))
C     ***** CHECK FOR EXCESS FORESHORTENING *****
      IF(FORE-ABS(U(3,1)))545,550,550
  545 CALL NORM(U(1,2),SYMB(1,3),VT(1,1),1)
      VT(1,3)=SYMB(1,3)
      VT(2,3)=SYMB(2,3)
      VT(3,3)=SYMB(3,3)
      HGT=CD(6,NB)
      OFF=CD(7,NB)
      IF(HGT)550,999,550
  550 T1=CD(8,NB)
      Z(1)=VT(1,4)-HGT*(11.+3.*T1)/7.
      Z(2)=VT(2,4)+OFF-HGT*.5
      Z(3)=VT(3,4)
      XO(3)=Z(3)
      ITILT=1
      I9=T1+2.
      T9=10.**I9
      DISTR=AINT((DIST*T9)+0.5)/T9 +.0001
      CALL NUMBUR(Z(1),Z(2),HGT,DISTR,0.,I9)
  570 ITILT=0
      IF(NJ2-10)580,999,999
  580 IF (NOUT.GE.0)
     &WRITE (NOUT,571)DIST
  571 FORMAT(1H ,59X,10HDISTANCE =,F8.3/1H )
      GO TO 999
  390 NG=14
      CALL ERPNT(WD(2),800)
  999 RETURN
      END
      SUBROUTINE DIFV(X,Y,Z)
C     VECTOR - VECTOR
C     Z(3)=X(3)-Y(3)
      DIMENSION X(3),Y(3),Z(3)
      Z(1)=X(1)-Y(1)
      Z(2)=X(2)-Y(2)
      Z(3)=X(3)-Y(3)
      RETURN
      END
      SUBROUTINE DRAW(W,DX,DY,NPEN)
      DIMENSION W(3),X(3),Y(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      Y(1)=W(1)+DX
      Y(2)=W(2)+DY
      IF(ITILT)115,140,115
C     ***** ROTATE FOR PERSPECTIVE TITLE *****
  115 Y(3)=XO(3)
      DO 120 I=1,3
  120 Z(I)=Y(I)-VT(I,4)
```

```
      DO 130 I=1,3
  130 X(I)=VT(I,1)*Z(1)+VT(I,2)*Z(2)+VT(I,3)*Z(3)+VT(I,4)
      CALL PLTXY(X,Y)
C     ***** CHECK BOUNDRY *****
  140 DO 160 J=1,2
      IF(Y(J)-XLNG(J)+.1)150,150,145
  145 Y(J)=XLNG(J)-.1
  150 IF(Y(J)-.1)155,160,160
  155 Y(J)=.1
  160 CONTINUE
C     ***** CHECK FOR OVERLAP *****
      NCQ=0
      CALL LAPDRW(Y,NPEN,NCQ)
      IF(NCQ)165,165,170
C     ***** CALL PLOTTING ROUTINE IF NO OVERLAPPING ELEMENTS ARE STORED
  165 CALL SCRIBE(Y,NPEN)
  170 RETURN
      END
      SUBROUTINE EDITR
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      common /ns/ npf,ndraw,norient,nvar
      CHARACTER*73 INQ,QUE,hque,tline
      character*80 answer,card
      CHARACTER*1 CH
   10 FORMAT(6X,'         1111111111222222222233333333334444444444555',
     *'5555555666666666677 7',/,1x,'LINE',1x,'1234567890123456789012345',
     *'67890123456789012345678901234567890123456789012')
      NUM=1
C *** PRINT PART OR ALL OF COMMAND QUE ***
      NUM1=MAX0(1,NUM-2)
  100 WRITE(*,103)
  103 FORMAT(1X)
      WRITE(*,111)(J,QUE(J),J=NUM1,NQUE)
  111 FORMAT(1X,I3,2X,A72)
C *** DISPLAY PROMPT ***
  115 WRITE(*,121)
  121 FORMAT(/,' C=Change line #                         D=Delete line(s
     &) # [#]',/,' I=Insert line before #                    T=Type line(s
     &) [#] [#]'
     &,/,' S=Save modified instruction set         O=Restore original in
     &struction set'
     &,/,' P=Save drawing as Postscript            H=Save drawing as HPG
     &L'
     &,/,' R=Redraw structure on screen            Q=Quit',/,' >-> ',$)
C *** READ COMMAND CHARACTER AND LINE NUMBER(S) ***
      read(*,131)answer
  131 format(a)
      if (answer(1:1).eq.' ') go to 133
      if (answer(1:1).ge.'1' .and. answer(1:1).le.'9') go to 133
      last=iend(answer)
      do 132 i=2,last
         ch=answer(i:i)
         if ((ch.ge.'a'.and.ch.le.'z') .or.
     *       (ch.ge.'A'.and.ch.le.'Z')) go to 133
         ich=ichar(ch)-48
         if ((ch.ne.' ') .and. ((ich.lt.0) .or. (ich.gt.9))) go to 133
  132 continue
      go to 135
  133 write(*,134)
  134 format (/,'***INVALID INPUT! Enter 1 letter and 0, 1, or 2 integer
     *s separated by spaces.***')
      go to 115
  135 answer=answer(1:last)//' 0 0'
c     read(answer,*)ch,num,num2
      open(15,status='scratch')
      write(15,136) answer(1:1),answer(2:75)
  136 format('''',a1,'''',1x,a)
      rewind(15)
      read(15,*)ch,num,num2
      close(15)
      numz=num
      NUM=MAX0(1,NUM)
      if (num2.gt.nque) then
         write(*,137)
  137    format (/,'*** value out of range ***')
         go to 115
      end if
      IF(NUM.GT.NUM2) NUM2=NUM
      write(6,*) ' '
      IF(CH.EQ.'T'.or.ch.eq.'t') GO TO 210
      IF(CH.EQ.'D'.or.ch.eq.'d') GO TO 240
      IF(CH.EQ.'C'.or.ch.eq.'c') GO TO 270
      IF(CH.EQ.'I'.or.ch.eq.'i') GO TO 310
      IF(CH.EQ.'O'.or.ch.eq.'o') GO TO 410
      IF(CH.EQ.'S'.or.ch.eq.'s') GO TO 420
      IF(CH.EQ.'R'.or.ch.eq.'r') GO TO 540
      IF(CH.EQ.'Q'.or.ch.eq.'q') GO TO 590
      IF(CH.EQ.'P'.or.ch.eq.'p') GO TO 510
      IF(CH.EQ.'H'.or.ch.eq.'h') GO TO 520
      GO TO 115
C *** TYPE LINES ***
  210 if (numz.eq.0) num2=nque
      WRITE(*,10)
      WRITE(*,111)(J,QUE(J),J=NUM,NUM2)
      GO TO 115
C *** DELETE LINES ***
  240 if (numz.eq.0) then
         write(6,*) ' *** Supply line number(s) with command ***'
         go to 115
      end if
      DO 260 I=NUM,NUM2
      NQUE=NQUE-1
      NEXT=NEXT-1
      DO 250 J=NUM,NQUE
  250 QUE(J)=QUE(J+1)
  260 QUE(NQUE+1)= '                                       '
      GO TO 100
C *** CHANGE AN OLD LINE ***
```

```
  270 if (numz.eq.0) then
         write(6,*) ' *** Supply line number with command ***'
         go to 115
      end if
c     NUM3=MAX0(1,NUM-3)
c     WRITE(*,111)(J,QUE(J),J=NUM3,NUM)
      write(*,271)
  271 format(' *** NOTE: Type @ to substitute a space in the original li
     &ne. ***',/)
      WRITE(*,10)
      WRITE(*,111)NUM,QUE(NUM)
      WRITE(*,276)NUM
  276 FORMAT(1X,I3,'  ',$)
      read(*,281) tline
  281 FORMAT(A72)
      do 282 i=1,72
         if (tline(i:i).ne.' ') then
            if (tline(i:i).eq.'@') then
               que(num)(i:i)=' '
            else
               que(num)(i:i)=tline(i:i)
            end if
         end if
  282 continue
      write(*,283)
  283 format(/,'      Line now reads:')
      WRITE(*,284)QUE(NUM)
  284 format(6x,a)
      write(*,285)
  285 format(/,' Hit ENTER or RETURN key ',$)
      read(*,131) CH
      GO TO 100
C *** INSERT A NEW LINE ***
  310 if (numz.eq.0) then
         write(6,*) ' *** Supply line number with command ***'
         go to 115
      end if
      IF(NQUE+1 .GT. 96) GO TO 115
      NQUE=NQUE+1
      NN=NQUE
      N=NQUE-NUM
      DO 320 J=1,N
      QUE(NN)=QUE(NN-1)
  320 NN=NN-1
      NUM4=MAX0(1,NUM-4)
      NUM1=MAX0(1,NUM-1)
      write(*,10)
      WRITE(*,111)(J,QUE(J),J=NUM4,NUM1)
      WRITE(*,276)NUM
      READ(*,281) QUE(NUM)
      NEXT=NEXT+1
      GO TO 100
C *** RETRIEVE OLD SET OF INSTRUCTIONS ***
  410 DO 415 J=1,NBACK
  415 que(j)=hque(j)
      NUM=1
      NUM1=1
      NQUE=NBACK
      NUM2=NQUE
      GO TO 100
C *** SAVE CURRENT SET OF INSTRUCTIONS ***
  420 CONTINUE
  421 write (*,422)
  422 format(' Enter file name:  ',$)
      read (*,131) answer
      open (16,file=answer,status='new',err=460)
      rewind(NED)
  430 read (NED,131,end=440) card
      write (16,131) card(1:iend(card))
      go to 430
  440 WRITE(16,450)(QUE(I),I=1,NQUE)
  450 FORMAT(A73)
      close(16)
      GO TO 115
  460 write(6,*) 'File already exists.  Choose a different name.'
      go to 421
C *** SAVE PICTURE AS POSTSCRIPT***
  510 ndraw=2
      go to 541
C *** SAVE PICTURE AS HPGL***
  520 ndraw=3
      go to 541
C *** REDRAW PICTURE ***
  540 ndraw=1
c *** default instruction save
  541 open (16,file='TEP.NEW',status='unknown')
      rewind(NED)
  542 read (NED,131,end=543) card
      write (16,131) card(1:iend(card))
      go to 542
  543 WRITE(16,450)(QUE(I),I=1,NQUE)
      close(16)
      call recycle
  590 RETURN
      END
      SUBROUTINE EIGEN (W,VALU,VECT)
C     ***** EIGENVALUES AND EIGENVECTORS OF 3X3 MATRIX *****
      DIMENSION W(3,3),VALU(3),VECT(3,3),A(3,3),B(3,3),V(3),U(3)
      COMMON NG
C     ***** STATEMENT FUNCTION *****
      PHIF(Z)=((B2-Z)*Z+B1)*Z+B0
C     ***** START OF PROGRAM *****
      ERRND=5.E-7
      SIGMA=0.
      DO 115 J=1,3
      DO 115 I=1,3
      TEM=W(I,J)
      A(I,J)=TEM
  115 SIGMA=SIGMA+TEM*TEM
C     ***** CHECK FOR NULL MATRIX *****
      IF(SIGMA)230,230,120
  120 SIGMA=SQRT(SIGMA)
C     ***** FORM CHARACTERISTIC EQUATION *****
      B2=A(1,1)+A(2,2)+A(3,3)
      B1=-A(1,1)*A(2,2)-A(1,1)*A(3,3)-A(2,2)*A(3,3)+A(1,3)*A(3,1)
     1+A(2,3)*A(3,2)+A(1,2)*A(2,1)
      B0=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)+A(1,3)*A(3,2)*A(2,1)-
     1A(1,3)*A(3,1)*A(2,2)-A(1,1)*A(2,3)*A(3,2)-A(1,2)*A(2,1)*A(3,3)
C     ***** FIRST ROOT BY BISECTION *****
      X=0.
      Y=SIGMA
      TEM=PHIF(SIGMA)
```

```
      VNEW=0.0
      IF(B0)135,250,145
  135 IF(TEM)140,140,165
  140 Y=-Y
      GO TO 165
  145 Y=0.
      X=SIGMA
      IF(TEM)165,165,150
  150 X=-X
C     ***** NOW PHIF(X).LT.0.AND.PHIF(Y).GT.0. *****
  165 VNEW=(X+Y)*.5
      DO 225 I=1,40
  175 IF(PHIF(VNEW))180,250,185
  180 X=VNEW
      GO TO 200
  185 Y=VNEW
  200 VOLD=VNEW
      VNEW=(X+Y)*.5
      TEM=ABS(VOLD-VNEW)
      IF(TEM-ERRND)250,250,205
  205 IF(VOLD)210,225,210
  210 IF (ABS(TEM/VOLD)-ERRND)250,250,225
  225 CONTINUE
C     ***** DID NOT CONVERGE, SET ERROR INDICATOR *****
  230 NG=6
      GO TO 400
C     ***** STORE FIRST ROOT *****
  250 U(3)=VNEW
C     ***** DEFLATE *****
      C1=B2-VNEW
      C0=B1+C1*VNEW
C     ***** SOLVE QUADRATIC *****
      TEM=C1*C1+4.*C0
      IF(TEM)255,265,260
C     ***** IGNORE IMAGINARY COMPONENT OF COMPLEX ROOT *****
  255 TEM=0.
      GO TO 265
  260 TEM=SQRT(TEM)
  265 U(1)=.5*(C1-TEM)
      U(2)=.5*(C1+TEM)
C     ***** SORT ROOTS *****
      DO 275 J=1,2
      IF(U(J)-U(3))275,275,270
  270 TEM=U(J)
      U(J)=U(3)
      U(3)=TEM
  275 CONTINUE
      LLL=-2
      DO 375 III=1,2
C     ***** CHECK FOR MULTIPLE ROOTS *****
      TEM=ERRND*100.
      NG=0
      L=1
      DO 305 I=1,2
      IF(U(I+1)-U(I)-TEM)300,300,290
  290 IF(U(I))295,305,295
  295 IF(ABS((U(I+1)-U(I))/U(I))-TEM)300,300,305
  300 L=L-1
      NG=NG-2*I
  305 CONTINUE
      IF(LLL-L)308,400,400
  308 LLL=L
C     ***** EIGENVECTOR ROUTINE *****
      DO 375 II=1,3
      T1=U(II)
      IF(L)315,310,322
C     ***** TWO VECTORS NULL FOR DOUBLE ROOT *****
  310 IF(NG+5-II)315,322,315
C     ***** ALL VECTORS NULL FOR TRIPLE ROOT *****
  315 DO 320 J=1,3
  320 VECT(J,II)=0.0
      GO TO 375
  322 DO 325 J=1,3
  325 A(J,J)=W(J,J)-T1
      SMAX=0.0
      DO 355 I=1,3
      I1=1
      IF(I-2)335,335,340
  335 I1=I+1
  340 B(I,1)=A(I,2)*A(I1,3)-A(I,3)*A(I1,2)
      B(I,2)=A(I,3)*A(I1,1)-A(I,1)*A(I1,3)
      B(I,3)=A(I,1)*A(I1,2)-A(I,2)*A(I1,1)
      TEM=B(I,1)**2+B(I,2)**2+B(I,3)**2
      IF(TEM-SMAX)355,355,350
  350 SMAX=TEM
      IMAX=I
  355 CONTINUE
      IF(SMAX)353,353,360
  353 NG=7
      GO TO 375
  360 SMAX=SQRT(SMAX)
      DO 365 J=1,3
  365 V(J)=B(IMAX,J)/SMAX
C     ***** REFINE EIGENVECTOR *****
      CALL AXEQB(A,V,V,1)
      TEM=AMAX1(ABS(V(1)),ABS(V(2)),ABS(V(3)))
      DO 370 J=1,3
  370 V(J)=V(J)/TEM
      CALL UNITY(V,VECT(1,II),1)
C     ***** REFINE EIGENVALUE *****
      T1=VMV(VECT(1,II),W,VECT(1,II))
      U(II)=T1
  375 VALU(II)=T1
  400 RETURN
      END
      SUBROUTINE ERPNT(TD,N)
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      character*63 msg(18)
      data msg /
     1 'No sentinel found after reading 96 symmetry cards',
     2 'No sentinel found after reading parameter cards for 100 atoms',
     3 'Aniso temp factor coefs form non-positive definite matrix',
```

```
   4 'Symmetry operation no. is higher than no. of input operations',
   5 'Atom number is higher than the number of input atoms',
   6 'Null temp factor matrix or failure in bisection routine',
   7 'Eigenvector routine failure due to null vector',
   8 'Error initializing screen driver',
   9 'Unidentified instruction number',
   a 'Atom out of bounds',
   b 'No vector search codes',
   c 'Insufficient number of atoms in ATOMS list',
   d 'Imaginary bond intersection (i.e., bond larger than atom)',
   e 'Hidden (end-on) bond',
   f 'Null vector as base line',
   g 'ATOMS array is full',
   h 'Maximum number of overlapping atoms (20) exceeded',
   i 'Maximum number of overlapping bonds (30) exceeded' /
    IF (NOUT.GE.0) then
       WRITE (NOUT,115)NG,TD,N
115    FORMAT(1H ,10X,10HFAULT NG =,I3,F10.0,I6)
       write (nout,116) msg(ng)
116    format(1H ,10x,a,/)
    end if
    NG=0
    RETURN
    END
    SUBROUTINE EXITNG(ING)
    REAL*8 AIN,ATOMID
    CHARACTER*4 TITLE,TITLE2
    COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
   1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
   2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
   3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
   4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
   5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
   6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
   7 ,XLNG(3),XO(3),XT(3)
    character*12 routin(18)
    character*63 msg(18)
    data routin /
   1 'PRELIM', 'PRELIM', 'PRELIM', 'ATOM, PAXES', 'ATOM, PAXES',
   2 'EIGEN', 'EIGEN', 'INITSC ', 'MAIN, SPARE', 'BOND, F700',
   3 'F800', 'F600, SEARCH', 'BOND', 'BOND', 'F900', 'STORE',
   4 'LAP700, LAPAB', 'LAP800, LAPAB' /
    data msg /
   1 'No sentinel found after reading 96 symmetry cards',
   2 'No sentinel found after reading parameter cards for 100 atoms',
   3 'Aniso temp factor coefs form non-positive definite matrix',
   4 'Symmetry operation no. is higher than no. of input operations',
   5 'Atom number is higher than the number of input atoms',
   6 'Null temp factor matrix or failure in bisection routine',
   7 'Eigenvector routine failure due to null vector',
   8 'Error initializing screen driver',
   9 'Unidentified instruction number',
   a 'Atom out of bounds',
   b 'No vector search codes',
   c 'Insufficient number of atoms in ATOMS list',
   d 'Imaginary bond intersection (i.e., bond larger than atom)',
   e 'Hidden (end-on) bond',
   f 'Null vector as base line',
   g 'ATOMS array is full',
   h 'Maximum number of overlapping atoms (20) exceeded',
   i 'Maximum number of overlapping bonds (30) exceeded' /
      if (ng.gt.0) then
         if (nout.gt.0) then
            write (nout,101) ing
            write (nout,102) routin(ing)
            write (nout,103) msg(ing)
         end if
         write (*,101) ing
         write (*,102) routin(ing)
         write (*,103) msg(ing)
      end if
      IF(NOUT.GT.0) CLOSE(NOUT,STATUS='KEEP')
 101  format('Fault Indicator:  ',i2)
 102  format('Subroutine(s) Involved:  ',a)
 103  format('Fault:  ',a)
      STOP
      END
      SUBROUTINE F200
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans

c *** NO drawing
      if (ndraw.eq.0) return

      go to (201,202,201,204,205), nj2
c *** initialize plotting (201 or 203 inst) ***
  201 xtrans=0.
      ytrans=0.
      if (ndraw .eq. 1) call initsc
      if (ndraw .eq. 2) call initps
      if (ndraw .eq. 3) call inithp
      if (ndraw .eq. 9) then
         open(unit=npf,file='TEP.EDT',status='unknown')
         nvar=1
      end if
      return
c *** change origin of plotting area or terminate (202 inst) ***
  202 if (ain(1) .eq. 0. .and. ain(2) .eq. 0.) then
         if (ndraw .eq. 2) call endps
         if (ndraw .eq. 3) call endhp
         if (ndraw .eq. 1) call endsc
         if (ndraw .eq. 9) close(npf)
      else
         xtrans=ain(1)
         ytrans=ain(2)
         if (ndraw .eq. 9) write (npf,203) xtrans,ytrans
  203    format('TRN',2(1x,f10.6))
      end if
      return
c *** change plot color (204 inst) ***
  204 icolor=ain(1)
      if (ndraw .eq. 1) call colrsc(icolor)
```

```
      if (ndraw .eq. 2) call colrps(icolor)
      if (ndraw .eq. 3) call colrhp(icolor)
      if (ndraw .eq. 9) call colrsc(icolor)
      return
c *** change pen width (205 inst) ***
c *** parameter units are thousandths of an inch (default=5)
  205 penw=ain(1)
      if (ndraw .eq. 1) call penwsc(penw)
      if (ndraw .eq. 2) call penwps(penw)
      if (ndraw .eq. 3) call penwhp(penw)
      if (ndraw .eq. 9) call penwsc(penw)
      return
      end
      SUBROUTINE F400
C     ***** ATOM LIST FUNCTIONS *****
      REAL*8 D100,D1000,D100K,TD,TD1,TD2
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100=100.
      D1000=1000.
      D100K=100000.
      NG=0
      IF(LATM)402,402,400
  400 DO 401 I=1,LATM
  401 CALL ATOM(ATOMID(I),ATOMS(1,I))
  402 IF(MOD(NJ2,10)-1)499,404,403
  403 IF(MOD(NJ2,10)-7)406,404,499
  406 CALL SEARC
      GO TO 499
C     ***** STORES (401) OR REMOVES (411) RUNS OF ATOMS *****
C     ***** RUN HIERARCHY = ATOM NO./SYM/ A/B/C TRANS. *****
  404 II=1
C     ***** FIND RUNS IN AIN ARRAY *****
  405 TD1=AIN(II)
      IF(TD1)410,410,420
  410 II=II+1
      IF(140-II)499,405,405
  420 JJ=II
C     ***** SET INITIAL RUN VALUES *****
      M1=TD1/D100K
      M2=DMOD(TD1,D100)
      M5=DMOD(TD1/D100,D1000)
      IF(M5)422,422,423
  422 M5=555
  423 M3=M5/100
      M4=MOD(M5/10,10)
      M5=MOD(M5,10)
  425 JJ=JJ+1
      IF(140-JJ)435,430,430
  430 TD2=-AIN(JJ)
      IF(TD2)435,425,440
  435 II=JJ-1
C     ***** SET TERMINAL VALUES FOR DEGENERATE RUN *****
      N1=M1
      N2=M2
      N3=M3
      N4=M4
      N5=M5
      GO TO 450
  440 II=JJ
C     ***** SET TERMINAL RUN VALUES *****
      N1=TD2/D100K
      N2=DMOD(TD2,D100)
      N5=DMOD(TD2/D100,D1000)
      IF(N5)445,445,446
  445 N5=555
  446 N3=N5/100
      N4=MOD(N5/10,10)
      N5=MOD(N5,10)
C     ***** LOOP THROUGH ALL RUNS *****
  450 DO 490 L5=M5,N5
      DO 490 L4=M4,N4
      DO 490 L3=M3,N3
      DO 490 L2=M2,N2
      DO 490 L1=M1,N1
      TD=DBLE(L1)*D100K+DBLE(L3*10000+L4*1000+L5*100+L2)
      CALL ATOM(TD,V1(1))
      IF(NG)455,458,455
  455 CALL ERPNT(TD,401)
      GO TO 490
C     ***** CHECK IDENT CODE IF 407/417 INSTRUCTION *****
  458 IF(MOD(NJ2,10)-7)475,460,490
  460 ID1=IDENT(1,L1)
      ID2=IDENT(2,L1)
      IF(NCD)490,490,465
  465 DO 470 J=1,NCD
         if (kd(1,j).gt.0. .and. kd(3,j).gt.0.) then
            if ((id1.ge.kd(1,j) .and. id1.le.kd(2,j)) .and.
     &          (id2.ge.kd(3,j) .and. id2.le.kd(4,j))) go to 475
         else if (kd(1,j).gt.0.) then
            if (id1.ge.kd(1,j) .and. id1.le.kd(2,j)) go to 475
         else if (kd(3,j).gt.0.) then
            if (id2.ge.kd(3,j) .and. id2.le.kd(4,j)) go to 475
         end if
c     IF(ID1-KD(1,J))470,467,467
c 467 IF(KD(2,J)-ID1)470,468,468
c 468 IF(ID2-KD(3,J))470,469,469
c 469 IF(KD(4,J)-ID2)470,475,475
  470 CONTINUE
      GO TO 490
  475 CALL STOR(TD)
  490 CONTINUE
      GO TO 410
  499 RETURN
      END
      SUBROUTINE F500
      DIMENSION RM(3,3),V(3,4)
      REAL*8 TD,D100,D1000,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
```

```
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      CHARACTER*8 CHEM
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      NG=0
      D100=100.
      D1000=1000.
      D100K=100000.
      IF(NJ2-11)500,700,710
  500 IF(NJ2-1)710,501,510
  501 TD=AIN(1)
      CALL ATOM(TD,ORGN)
      IF(NG)502,504,502
  502 CALL ERPNT(TD,501)
      CALL EXITNG(NG)
  504 DO 506 K=1,4
      TD=AIN(K+1)
      CALL ATOM(TD,V(1,K))
      IF(NG)502,506,502
  506 CONTINUE
      DO 507 J=1,3
      V1(J)=V(J,2)-V(J,1)
  507 V2(J)=V(J,4)-V(J,3)
      IND=-1
      IF(AIN(7))509,509,508
  508 IND=-2
  509 CALL AXES(V1,V2,REFV,IND)
      GO TO 670
  510 IF(NJ2-4)515,511,599
C      ***** SHIFT ORIGIN FOR PROJECTION AXIS (IN INCHES) *****
  511 DO 513 J=1,3
      DO 512 K=1,3
      T1=AIN(K)
  512 ORGN(J)=ORGN(J)+REFV(J,K)*T1/SCAL1
      T2=AIN(J)
  513 XO(J)=XO(J)+T2
      GO TO 675
C      ***** FORM ROTATION MATRIX *****
  515 DO 514 J=1,3
      DO 514 K=1,3
  514 V(J,K)=REFV(J,K)
      DO 517 L=1,139,2
      I=AIN(L)
      IF(I) 532,519,516
  516 X=AIN(L+1)*1.7453293D-2
      T1=COS(X)
      T2=SIN(X)
      I3=MOD(I+2,3)+1
      I1=MOD(I3,3)+1
      I2=MOD(I1,3)+1
      RM(I1,I1)=T1
      RM(I1,I2)=T2
      RM(I1,I3)=0.0
      RM(I2,I1)=-T2
      RM(I2,I2)=T1
      RM(I2,I3)=0.0
      RM(I3,I1)=0.0
      RM(I3,I2)=0.0
      RM(I3,I3)=1.0
  517 CALL MM(V,RM,V)
  519 IF(NJ2-3)518,525,599
  518 DO 522 J=1,3
      DO 522 I=1,3
  522 REFV(I,J)=V(I,J)
      GO TO 552
  525 DO 528 J=1,3
      DO 528 I=1,3
  528 WRKV(I,J)=V(I,J)
      GO TO 552
  532 IF(NJ2-3)535,552,599
  535 I=MOD(-I,3)
      DO 542 J=1,I
      DO 542 K=1,3
      T1=REFV(K,3)
      REFV(K,3)=REFV(K,2)
      REFV(K,2)=REFV(K,1)
  542 REFV(K,1)=T1
  552 CONTINUE
      IF(NJ2-3)670,582,599
  582 CALL MM(AA,WRKV,AAWRK)
      GO TO 699
  599 IF(NJ2-5)699,600,607
  600 IF(LATM-1)605,610,610
  605 NG=12
  606 CALL ERPNT(0.D0,506)
      CALL EXITNG(NG)
  607 IF(NJ2-6)699,608,710
  608 IF(LATM-3)605,610,610
  610 DO 612 J=1,3
      V2(J)=0.0
      DO 612 I=1,3
  612 RM(I,J)=0.0
      AWT=0.0
      DO 620 K=1,LATM
      CALL ATOM(ATOMID(K),ATOMS(1,K))
      T2=1.0
      IF(NCD)618,618,613
  613 I1=ATOMID(K)/D100K
      DO 616 J=1,NCD
      if (kd(5,j).eq.1) i1=ident(1,k)
      if (kd(5,j).eq.2) i1=ident(2,k)
      T2=CD(1,J)
      IF(I1-KD(1,J))616,614,614
  614 IF(KD(2,J)-I1)616,618,618
  616 CONTINUE
      GO TO 620
  618 AWT=AWT+T2
      DO 619 J=1,3
  619 V2(J)=V2(J)+ATOMS(J,K)*T2
  620 CONTINUE
      IF(AWT)605,605,621
C     ***** PUT ORIGIN AT CENTER OF GRAVITY *****
  621 DO 622 J=1,3
  622 ORGN(J)=V2(J)/AWT
```

```
      IF(NJ2-6)699,624,710
C     ***** FORM PRODUCT-MOMENT MATRIX FOR ATOMS IN ATOM LIST *****
  624 DO 630 K=1,LATM
      T2=1.0
      IF(NCD)628,628,625
  625 I1=ATOMID(K)/D100K
      DO 627 J=1,NCD
      if (kd(5,j).eq.1) i1=ident(1,k)
      if (kd(5,j).eq.2) i1=ident(2,k)
      T2=CD(1,J)
      IF(I1-KD(1,J))627,626,626
  626 IF(KD(2,J)-I1)627,628,628
  627 CONTINUE
      GO TO 630
  628 DO 629 J=1,3
      T1=(ATOMS(J,K)-ORGN(J))*T2
      DO 629 I=1,3
  629 RM(I,J)=T1*(ATOMS(I,K)-ORGN(I))+RM(I,J)
      T1=0.03/(RM(1,1)+RM(2,2)+RM(3,3))
  630 CONTINUE
      DO 632 J=1,3
      DO 632 I=1,3
  632 RM(I,J)=RM(I,J)*T1
C     ***** TRANSFORM TO INERTIAL AXIS SYSTEM *****
      CALL MM(RM,AA,DA)
      CALL EIGEN(DA,RMS,PAT)
      IF(RMS(2))605,605,635
  635 IF(NG)640,633,606
C     ***** MAKE SURE VECTORS ARE ORTHOGONAL --> NEW REFERENCE VECTORS *
  633 CALL AXES(PAT(1,3),PAT(1,1),REFV,-1)
      GO TO 665
C     ***** TWO EQUAL EIGENVECTORS SPECIAL CASE *****
  640 IF(NG+6)665,665,645
  645 N=NG+5
      CALL UNITY(PAT(1,N),V1,-1)
      DO 650 K=1,3
      IF(ABS(VMV(V1,AA,REFV(1,K)))-.58)655,650,650
  650 CONTINUE
  655 CALL AXES(V1,REFV(1,K),DA,-1)
      DO 660 K=1,3
      L=MOD(N-K+2,3)+1
      DO 660 J=1,3
  660 REFV(J,L)=DA(J,K)
  665 NG=0
  670 CALL MM(AA,REFV,AAREV)
  675 DO 680 J=1,3
      DO 680 I=1,3
      WRKV(I,J)=REFV(I,J)
  680 AAWRK(I,J)=AAREV(I,J)
C     ***** ELIMINATE ALL PREVIOUSLY STORED OVERLAP INFORMATION ****
C     ***** (ALL INSTRUCTIONS FROM 501 THROUGH 510 DO THIS *****
  699 CALL LAP500(0)
      GO TO 710
C     ***** STORE NEW OVERLAP INFORMATION (INSTRUCTION 511) *****
  700 CALL LAP500(1)
  710 RETURN
      END
      SUBROUTINE F600
C     ***** SCALING AND CENTERING FUNCTIONS *****
      DIMENSION MAX(3),SCAL(4),X(3),XMAX(3),XMIN(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      dimension crtval(99)
      data crtval /
     1 0.3389, 0.4299, 0.4951, 0.5479, 0.5932, 0.6334, 0.6699, 0.7035,
     2 0.7349, 0.7644, 0.7924, 0.8192, 0.8447, 0.8694, 0.8932, 0.9162,
     3 0.9386, 0.9605, 0.9818, 1.0026, 1.0230, 1.0430, 1.0627, 1.0821,
     4 1.1012, 1.1200, 1.1386, 1.1570, 1.1751, 1.1932, 1.2110, 1.2288,
     5 1.2464, 1.2638, 1.2812, 1.2985, 1.3158, 1.3330, 1.3501, 1.3672,
     6 1.3842, 1.4013, 1.4183, 1.4354, 1.4524, 1.4695, 1.4866, 1.5037,
     7 1.5209, 1.5382, 1.5555, 1.5729, 1.5904, 1.6080, 1.6257, 1.6436,
     8 1.6616, 1.6797, 1.6980, 1.7164, 1.7351, 1.7540, 1.7730, 1.7924,
     9 1.8119, 1.8318, 1.8519, 1.8724, 1.8932, 1.9144, 1.9360, 1.9580,
     a 1.9804, 2.0034, 2.0269, 2.0510, 2.0757, 2.1012, 2.1274, 2.1544,
     b 2.1824, 2.2114, 2.2416, 2.2730, 2.3059, 2.3404, 2.3767, 2.4153,
     c 2.4563, 2.5003, 2.5478, 2.5997, 2.6571, 2.7216, 2.7955, 2.8829,
     d 2.9912, 3.1365, 3.3682 /
C     ***** DEL = 1. FOR INCRUMENTING FUNCTIONS *****
C     ***** DEL = 0. FOR REGULAR FUNCTIONS *****
      DEL=FLOAT(MOD(NJ2/10,2))
      NJ2=MOD(NJ2,10)
C     ***** EXPLICIT ORIGIN AND SCALE *****
      T1=AIN(1)
      IF(T1)602,604,602
  602 XO(1)=T1+XO(1)*DEL
  604 T2=AIN(2)
      IF(T2)606,608,606
  606 XO(2)=T2+XO(2)*DEL
  608 T3=AIN(3)
      IF(T3)612,612,609
  609 IF(DEL)611,611,610
  610 SCAL1=SCAL1*T3
      GO TO 612
  611 SCAL1=T3
  612 T4=AIN(4)
      IF(T4)615,616,614
C     ***** SET ELLIPSOID SCALE FACTOR *****
  614 SCAL2=T4
      go to 616
  615 t4=-t4
      if (t4.gt.0. .and. t4.lt.1.) t4=100.*t4
      it4=t4
      scal2=crtval(it4)
C     ***** AUTOMATIC ORIGIN AND/OR SCALE *****
  616 IF(NJ2-2)790,622,620
  620 XO(1)=XLNG(1)*.5
      XO(2)=XLNG(2)*.5
  622 IF(NJ2-3)625,640,625
  625 SCAL1=1.
  630 IF(LATM-1)635,635,640
  635 NG=12
      CALL ERPNT(0.D0,602)
      CALL EXITNG(NG)
  640 DO 650 J=1,3
```

```
      XMAX(J)=-1.E5
  650 XMIN(J)=1.E5
C     ***** FIT BOX AROUND SET OF ATOMS *****
      DO 670 I=1,LATM
      CALL XYZ(ATOMID(I),ATOMS(1,I),3)
      IF(NG)652,653,652
  652 CALL ERPNT(ATOMID(I),600)
      GO TO 670
  653 DO 668 J=1,3
      T1=ATOMS(J,I)
      IF(XMAX(J)-T1)655,660,660
  655 XMAX(J)=T1
      MAX(J)=I
  660 IF(T1-XMIN(J))665,668,668
  665 XMIN(J)=T1
  668 CONTINUE
  670 CONTINUE
C     ***** KM=TOP ATOM *****
      KM=MAX(3)
      SMULT=1.
      DO 780 M=1,5
      IF(M-2)740,675,678
C     ***** CHECK VIEW DISTANCE *****
  675 IF(VIEW)785,785,680
  678 IF(NJ2-3)680,785,680
  680 T1=ATOMS(3,KM)*SMULT
      IF(VIEW*.5-T1)685,690,690
C     ***** INCREASE VIEW DISTANCE *****
  685 VIEW=2.*T1
C     ***** FIND PERSPECTIVE PROJECTION LIMITS *****
  690 DO 700 J=1,2
      XMAX(J)=-1.E5
  700 XMIN(J)=1.E5
      DO 725 I=1,LATM
      DO 705 J=1,3
  705 X(J)=ATOMS(J,I)*SMULT
      T2=VIEW/(VIEW-X(3))
      DO 725 J=1,2
      T1=X(J)*T2
      IF(XMAX(J)-T1)710,715,715
  710 XMAX(J)=T1
  715 IF(T1-XMIN(J))720,725,725
  720 XMIN(J)=T1
  725 CONTINUE
C     ***** REFINE PARAMETERS *****
  740 IF(NJ2 -3)745,742,755
  742 SMUL2=1.
      GO TO 765
C     ***** AUTOMATIC SCALE ONLY *****
  745 DO 750 J=1,2
      T2=XO(J)
      SCAL(J)=(BRDR-T2)/XMIN(J)
  750 SCAL(J+2)=(XLNG(J)-BRDR-T2)/XMAX(J)
      SMUL2=AMIN1(SCAL(1),SCAL(2),SCAL(3),SCAL(4))
      GO TO 780
C     ***** AUTOMATIC SCALE AND POSITION *****
  755 DO 760 J=1,2
  760 SCAL(J)=(XLNG(J)-BRDR*2.)/(XMAX(J)-XMIN(J))
      SMUL2=AMIN1(SCAL(1),SCAL(2))
C     ***** AUTOMATIC POSITION *****
  765 DO 770 J=1,2
  770 XO(J)=.5*(XLNG(J)-SMUL2*(XMAX(J)+XMIN(J)))
  780 SMULT=SMULT*SMUL2
      VIEW=VIEW*SMUL2
  785 SCAL1=SCAL1*SMULT
  790 SCL=SCAL1*SCAL2
C     ***** ELIMINATE ALL PREVIOUSLY STORED OVERLAP INFORMATION *****
      CALL LAP500(0)
      RETURN
      END
      SUBROUTINE F700
C     ***** SUBROUTINE TO DRAW ELLIPSOIDS *****
      DIMENSION EYE(3),VIEWV(3),X(3),Z(3)
      REAL*8 TD,TD2,D100,D1000,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans
C     ***** SET ELLIPSOID GRAPHIC DETAILS *****
      D100=100.
      D1000=1000.
      D100K=100000.
      ITILT=0
      NG=0
      NFIRST=1
      NPLANE=AIN(1)
      IF(NPLANE-1)720,715,720
  715 NFIRST=4
      NPLANE=4
  720 NSOLID=AIN(2)
      NDOT=64/2**(IABS(NSOLID))
      LINES=AIN(3)
      NDASH=AIN(4)
      CHSYM=AIN(5)
      T6=AIN(6)
      DH=T6-CHSYM*17./7.
      T7=AIN(7)
      DV=T7-CHSYM*.5
C     ***** ESTABLISH REFERENCE POINT OF VIEW *****
      T1=1.E6
      IF(VIEW)740,740,735
  735 T1=VIEW/SCAL1
  740 DO 741 J=1,3
  741 EYE(J)=REFV(J,3)*T1+ORGN(J)
      LNS=-1
C     ***** LOOP THROUGH ATOM LIST *****
      DO 1105 ITOM=1,LATM
      TD=ATOMID(ITOM)
      K=TD/D100K
      IF(AIN(10))744,744,7412
```

```
 7412 IF(AIN(12)-1.0D0)742,7414,7416
 7414 TD2=IDENT(1,K)
      GO TO 7422
 7416 TD2=IDENT(2,K)
      GO TO 7422
  742 TD2=DINT(TD/D100K)
 7422 IF(TD2-AIN(10))1105,743,743
  743 IF(AIN(11)-TD2)1105,744,744
  744 CALL XYZ(TD,X,2)
      IF(NG)758,746,758
  746 CALL PLTXY(X,Z)
      L=DMOD(TD/D100,D1000)
      L1=DMOD(TD,D100)
      if (ndraw.eq.1) WRITE (npf,750) CHEM(K),K,L,L1,
     &                                 Z(1)+xtrans,Z(2)+ytrans
      IF(NJ2-10)747,754,754
  747 LNS=MOD(LNS+1,18)
      IF(LNS)749,748,749
  748 IF (NOUT.GE.0)
     &WRITE (NOUT,751)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,752)
  749 IF (NOUT.GE.0)
     &WRITE (NOUT,750) CHEM(K),K,L,L1,Z(1),Z(2),
     1(X(I),I=1,3),(XT(I),I=1,3)
  750 FORMAT(1H 10X,A6,3H  (I3,1H,I3,I2,4H)   2F8.2,3X,3F8.3,11X,3F8.4)
  751 FORMAT(1H1,10X,18A4)
  752 FORMAT(1H010X,18HSYMBOL   ATOM CODE7X,16HPLOTTER X,Y(IN.) 3X,21HCA
     1RTESIAN X,Y,Z (IN.)15X,20HCRYSTAL SYSTEM X,Y,Z/1H 19X,45H(DIRECTIO
     2N COSINES(I,J),I=1,3),RMSD(J)),J=1,312X,42HFOR PRINCIPAL AXES BASE
     3D ON WORKING SYSTEM/1H )
  754 IF(EDGE-BRDR*.75)755,760,760
  755 NG=10
  758 CALL ERPNT(TD,700)
      GO TO 1105
C     ***** CALL OVERLAP ROUTINE *****
  760 ICQ=0
      CALL LAP700(ITOM,ICQ)
      IF(ICQ)762,764,764
C     ***** OMIT HIDDEN ATOM *****
  762 NG=14
      GO TO 758
  764 IF(CHSYM)775,775,765
C     ***** PLOT CHEMICAL SYMBOLS *****
  765 T4=1.
      IF(VIEW)767,767,766
  766 T4=VIEW/(VIEW-X(3))
  767 T3=CHSYM*T4
      T4=DISP*T4*.5
      V1(1)=X(1)+DH*SYMB(1,1)+DV*SYMB(1,2)
      V1(2)=X(2)+DH*SYMB(2,1)+DV*SYMB(2,2)
      V1(3)=X(3)
      CALL PLTXY(V1,V3)
      IF(EDGE-CHSYM)775,768,768
  768 V2(3)=0.
      DO 770 I=1,3,2
      V2(1)=V3(1)+FLOAT(I-2)*T4
      DO 770 J=1,3,2
      V2(2)=V3(2)+FLOAT(J-2)*T4
      CALL SIMBOL(V2(1),V2(2),T3,CHEM(K),THETA,6)
      IF(T4)775,775,770
```

```
  770 CONTINUE
  775 IF(NPLANE)1105,1105,780
C     ***** ELLIPSOID PRINC VECTORS TOWARD VIEWER *****
  780 CALL PAXES(TD,2)
      IF(NG)758,783,758
  783 CALL DIFV(EYE,XT,VIEWV)
      CALL UNITY(VIEWV,VIEWV,-1)
      CALL VM(VIEWV,AA,V2)
      DO 795 I=1,3
      IF(VV(V2,PAT(1,I)))785,795,795
  785 DO 790 J=1,3
      PAC(J,I)=-PAC(J,I)
  790 PAT(J,I)=-PAT(J,I)
  795 CONTINUE
      DO 800 J=1,3
      PAC(J,4)=PAC(J,1)
  800 PAC(J,5)=PAC(J,2)
      IF(NJ2-10)802,803,803
  801 FORMAT(1H ,13X,3(3X,3F8.4,F8.5)/1H )
  802 IF (NOUT.GE.0)
     &WRITE (NOUT,801) ((PAC(J,K),J=1,3),RMS(K) ,K=1,3)
C     ***** V4 = VECTOR NORMAL TO POLAR PLANE *****
  803 continue
      CALL VM(VIEWV,AAWRK,V6)
      CALL UNITY(V6,V6,1)
      CALL MV(Q,V6,V4)
      CALL UNITY(V4,V4,1)
C     ***** SET PLOTTING RESOLUTION FOR ELLIPSOID *****
      t3a=sqrt(rms(3)*rms(2))
      t3b=sqrt(rms(2)*rms(1))
      t3c=sqrt(rms(3)*rms(1))
      T3=((t3a+t3b+t3c)/3.0)*SCL
      NRESOL=1
      NBIS=5
      DO 805 J=1,3
      IF(T3-RES(J))804,810,810
  804 NBIS=NBIS-1
  805 NRESOL=NRESOL*2
  810 NRES1=NRESOL+1
C     ***** LOOP THROUGH PRINC AND POLAR PLANES *****
      DO 1100 II=NFIRST,NPLANE
      II0=MOD(II+2,3)+1
      II1=MOD(II,3)+1
      II2=MOD(II+1,3)+1
C     ***** GENERATE CONJUGATE DIAMETERS *****
      IF(.99938- ABS(VV(V4,PAC(1,II2))))820,820,830
  820 T1=RMS(II0)*SCL
      T2=RMS(II1)*SCL
      DO 825 J=1,3
      DA(J,1)=PAC(J,II0)*T1
  825 DA(J,2)=PAC(J,II1)*T2
      GO TO 850
  830 CALL NORM(PAC(1,II0),PAC(1,II1),V1,1)
      CALL NORM(V1,V4,V2,1)
      CALL UNITY(V2,V2,1)
      CALL MV(Q,V2,V3)
      IF(II-4)835,840,840
  835 CALL NORM(V3,V1,V5,1)
      GO TO 843
  840 CALL NORM(V3,V4,V5,1)
  843 CALL UNITY(V5,V5,1)
```

```
      T1=SCL/SQRT(VMV(V2,Q,V2))
      T2=SCL/SQRT(VMV(V5,Q,V5))
      DO 845 J=1,3
      DA(J,1)=V2(J)*T1
  845 DA(J,2)=V5(J)*T2
C     ***** GENERATE ELLIPSE *****
  850 CALL RADIAL(NBIS)
      IF(II-4)900,851,851
  851 IF(NSOLID)859,859,852
C     ***** PLOT DOTTED BOUNDARY ELLIPSE *****
  852 IF(NDOT-NRESOL)853,855,855
  853 CALL RADIAL(NSOLID-1)
  855 CALL PROJ(D,DP,X,XO,VIEW,1,129,NDOT)
      DO 857 J=1,129,NDOT
      CALL DRAW(DP(1,J),DISP,DISP,3)
      DO 856 I=1,3,2
      T1=FLOAT(I-2)*DISP
      DO 856 K=1,3,2
      T2=FLOAT(K-2)*DISP
      CALL DRAW(DP(1,J),T1,T2,2)
      IF(DISP)857,857,856
  856 CONTINUE
  857 CONTINUE
      GO TO 1100
C     ***** PLOT SOLID BOUNDARY ELLIPSE *****
  859 CALL PROJ(D,DP,X,XO,VIEW,1,129,NRESOL)
      CALL DRAW(DP,0.,0.,3)
      DO 860 J=NRES1,129,NRESOL
  860 CALL DRAW(DP(1,J),0.,0.,2)
      IF(DISP)1100,1100,865
C     ***** BOUNDARY ANNULUS AS A LINEAR FUNCTION OF HEIGHT *****
  865 CALL DIFV(XT,ORGN,V1)
      T5=VV(V1,AAREV(1,3))*SCAL1
      T8=AIN(8)
      T9=AIN(9)
      NCYCLE=.5+(T8+T5*T9)/DISP
      IF(NCYCLE)1100,1100,870
  870 T3=(2.*DISP)/(T1+T2)
C     ***** INCREASE ANNULAR THICKNESS *****
      DO 875 I=1,NCYCLE
      T4=T3*FLOAT(I)
      DO 875 J=1,129,NRESOL
  875 CALL DRAW(DP(1,J),D(1,J)*T4,D(2,J)*T4,2)
      GO TO 1100
  900 CALL PROJ(D,DP,X,XO,VIEW,1,65,NRESOL)
C     ***** PLOT HALF AN ELLIPSE *****
      CALL DRAW(DP,0.,0.,3)
      DO 905 J=NRES1,65,NRESOL
  905 CALL DRAW(DP(1,J),0.,0.,2)
      IF(DISP)930,930,910
C     ***** ACCENTUATE FRONT HALF *****
  910 DO 925 I=1,3,2
      T2=FLOAT(I-2)*DISP
      DO 915 J=1,65,NRESOL
      K=66-J
  915 CALL DRAW(DP(1,K),DISP,T2,2)
      DO 925 K=1,65,NRESOL
  925 CALL DRAW(DP(1,K),-DISP,-T2,2)
  930 IF(NSOLID)940,967,935
  935 L=NDOT
      IF(NDOT-NRESOL)938,945,940
  938 CALL RADIAL(NSOLID-1)
      GO TO 945
  940 L=NRESOL
  945 CALL PROJ(D(1,65),DP(1,65),X,XO,VIEW,1,65,L)
      IF(NSOLID)960,967,950
C     ***** DOTTED LINE ON REVERSE SIDE *****
  950 DO 958 J=65,129,NDOT
      CALL DRAW(DP(1,J),DISP,DISP,3)
      DO 955 I=1,3,2
      T1=FLOAT(I-2)*DISP
      DO 955 K=1,3,2
      T2=FLOAT(K-2)*DISP
      CALL DRAW(DP(1,J),T1,T2,2)
      IF(DISP)958,958,955
  955 CONTINUE
  958 CONTINUE
      GO TO 967
C     ***** SINGLE LINE ON REVERSE SIDE *****
  960 DO 965 J=65,129,NRESOL
  965 CALL DRAW(DP(1,J),0.,0.,2)
C     ***** DETAIL INTERIOR FEATURES *****
  967 T2=NDASH*2
      DO 975 J=1,3
      T1=PAC(J,II0)*RMS(II0)*SCL
      DA(J,1)=T1
      DA(J,2)=PAC(J,II1)*RMS(II1)*SCL
      DA(J,3)=0.
      IF(NDASH)975,975,970
  970 V1(J)=-T1
      V2(J)=T1/T2
  975 CONTINUE
      IF(NDASH)987,987,980
C     ***** DASHED LINE FOR REVERSE AXIS *****
  980 DO 985 J=1,NDASH
      DO 985 K=1,2
      L=4-K
      CALL PROJ(V1,DP,X,XO,VIEW,1,1,1)
      CALL DRAW(DP,0.,0.,L)
      DO 985 I=1,3
  985 V1(I)=V1(I)+V2(I)
C     ***** SOLID LINE FOR FORWARD AXIS *****
  987 IF(LINES)1100,1100,988
  988 CALL PROJ(DA,DP,X,XO,VIEW,1,3,1)
      T1=DISP*.5
      DO 990 I=1,3,2
      T2=FLOAT(2-I)*T1
      CALL DRAW(DP,T1,T2,3)
      CALL DRAW(DP(1,3),T1,T2,2)
      IF(DISP)1000,1000,989
  989 CALL DRAW(DP(1,3),-T1,T2,2)
  990 CALL DRAW(DP,-T1,T2,2)
C     ***** SHADE QUADRANT BETWEEN TWO PRINCIPAL AXES *****
 1000 L=LINES-1
      IF(L)1100,1100,1005
 1005 T2=LINES
      DO 1025 I=1,L
      T1=FLOAT(I)/T2
      T3=SQRT(1.-T1*T1)
      IF(MOD(I,2))1010,1015,1010
 1010 M=I*2
      N=M-1
```

```
      GO TO 1020
 1015 N=I*2
      M=N-1
 1020 DO 1025 J=1,3
      T4=DA(J,1)*T1
      D(J,M)=T4
 1025 D(J,N)=DA(J,2)*T3+T4
      L=L*2
      CALL PROJ(D,DP,X,XO,VIEW,1,L,1)
      DO 1030 I=2,L,2
      CALL DRAW(DP(1,I-1),0.,0.,3)
 1030 CALL DRAW(DP(1,I),0.,0.,2)
 1100 CONTINUE
 1105 CONTINUE
C     ***** ELIMINATE LOCAL OVERLAP INFORMATION BEFORE RETURNING *****
      CALL LAP500(-1)
      RETURN
      END
      SUBROUTINE F800
C     ***** SUBROUTINE FINDS ATOM PAIRS FOR BONDS *****
      DIMENSION IA(3),W1(6)
      REAL*8 TD1,TD2,TD3,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      D100K=100000.
c *** old
c     NJ4=MOD(NJ2,10)-4
c *** new
      IAN=AIN(2)
      NJ4=(MOD(NJ2,10)-4)+(IAN*2)
      if (nj.eq.10) nj4=(ian*2)-2
      LNS=-4
      IF(MOD(NJ2,10)-2)805,848,848
C     ***** EXPLICIT DESCRIPTION *****
  805 II=0
      IF(NCD)810,810,815
  810 NG=11
      CALL ERPNT(0.D0,NJ*100+NJ2)
      GO TO 980
  815 II=II+1
      IF(140-II)980,980,820
  820 TD1=AIN(II)
      IF(TD1)815,815,825
  825 II=II+1
      TD2=AIN(II)
      IF(TD2)815,815,830
  830 IF(NJ2-10)832,838,838
```

```
  832 LNS=MOD(LNS+4,56)
      IF(LNS)838,834,838
  834 IF (NOUT.GE.0)
     &WRITE (NOUT,835)(TITLE(I),I=1,18)
  835 FORMAT(1H1,10X,18A4)
      IF (NOUT.GE.0)
     &WRITE (NOUT,837)
  837 FORMAT(1H010X,18HSYMBOL   ATOM CODE6X,16HPLOTTER X,Y(IN.) 6X,21HCA
     1RTESIAN X,Y,Z (IN.)17X,20HCRYSTAL SYSTEM X,Y,Z/1H )
C     ***** CHECK IF BOND ATOMS ARE IN ATOMS LIST (FOR OVERLAP CALC) ***
  838 NA1=0
      NA2=0
      IF(LATM-2)845,839,839
  839 N2=2
      DO 844 K=1,LATM
      TD3=ATOMID(K)
      IF(TD3-TD1)841,840,841
  840 NA1=K
      GO TO 843
  841 IF(TD3-TD2)844,842,844
  842 NA2=K
  843 N2=N2-1
      IF(N2)845,845,844
  844 CONTINUE
  845 IF(NA2-NA1)846,847,847
  846 NA3=NA1
      NA1=NA2
      NA2=NA3
      TD3=TD1
      TD1=TD2
      TD2=TD3
  847 CALL BOND(TD1,TD2,1,NA1,NA2)
      GO TO 815
C     ***** IMPLICIT DESCRIPTION *****
  848 IF(LATM-2)810,850,850
  850 SCAL3=SCAL1
      SCAL1=1.
      DO 855 I=1,LATM
  855 CALL XYZ(ATOMID(I),ATOMS(1,I),2)
      SCAL1=SCAL3
      IF(NCD)810,810,860
  860 IF (NOUT.GE.0)
     &WRITE (NOUT,861)
  861 FORMAT(1H010X,20HBOND SELECTION CODES//11X,94H(SEQUENCE(A))(SEQUEN
     1CE(B)) (BOND) (DISTANCES)( BOND )(PERSP.--LABELS)(NORMAL--LABELS)(
     2DIGITS) /11X,93H(  MIN  MAX )(  MIN  MAX ) (TYPE) (MIN   MAX)(RADI
     3US)(HEIGHT  OFFSET)(HEIGHT  OFFSET)(NUMBER))
      DMAX=0.
      DO 870 I=1,NCD
      IF(DMAX-CD(2,I))865,866,866
  865 DMAX=CD(2,I)
  866 IF (NOUT.GE.0)
     &WRITE (NOUT,871)(KD(J,I),J=1,5),(CD(J,I),J=1,8)
  870 CONTINUE
  871 FORMAT(1H ,10X,I6,I5,I8,I5,I8,2F6.2,5F8.3,F7.0)
      DMAX=DMAX*DMAX
C     ***** LOOP THROUGH ATOMS ARRAY *****
      DO 977 M=1,LATM
      NA1=M
      TD1=ATOMID(M)
      MI=TD1/D100K
```

```
      IF(NJ4)8722,8724,8718
 8718 IF(NJ4-2)8724,8726,8720
 8720 IF(NJ4-4)8726,8722,8722
 8722 IA(1)=TD1/D100K
      GO TO 8728
 8724 IA(1)=IDENT(1,MI)
      GO TO 8728
 8726 IA(1)=IDENT(2,MI)
 8728 IA(3)=IA(1)
      W1(1)=ATOMS(1,M)
      W1(2)=ATOMS(2,M)
      W1(3)=ATOMS(3,M)
      L=M+1
      IF(LATM-L)977,872,872
  872 DO 975 N=L,LATM
      NA2=N
      DIST=(ATOMS(1,N)-W1(1))**2
      IF(DMAX-DIST)975,873,873
  873 DIST=DIST+(ATOMS(2,N)-W1(2))**2
      IF(DMAX-DIST)975,874,874
  874 DIST=DIST+(ATOMS(3,N)-W1(3))**2
      IF(DMAX-DIST)975,875,875
  875 DIST=SQRT(DIST)
      TD2=ATOMID(N)
      NI=TD2/D100K
c     IF(NJ4)876,877,878
c 876 IA(2)=TD2/D100K
c     GO TO 879
c 877 IA(2)=IDENT(1,NI)
c     GO TO 879
c 878 IA(2)=IDENT(2,NI)
      IF(NJ4.LT.0) IA(2)=TD2/D100K
      IF(NJ4.EQ.0.OR.NJ4.EQ.1) IA(2)=IDENT(1,NI)
      IF(NJ4.GT.1) IA(2)=IDENT(2,NI)
C     ***** SELECT BONDS ACCORDING TO CODES *****
  879 DO 950 J=1,NCD
      JB=J
      IF(DIST-CD(1,J))   950,880,880
  880 IF(CD(2,J)-DIST)   950,881,881
  881 DO 885 K=1,2
      IF(IA(K)-KD(1,J))  885,882,882
  882 IF(KD(2,J)-IA(K))  885,883,883
  883 IF(IA(K+1)-KD(3,J))885,884,884
  884 IF(KD(4,J)-IA(K+1))885,890,890
  885 CONTINUE
      GO TO 950
C     ***** CHECK FOR POLYHEDRA CODE *****
  890 IF(CD(4,J))900,955,955
  900 W1(4)=ATOMS(1,N)
      W1(5)=ATOMS(2,N)
      W1(6)=ATOMS(3,N)
      KM1=ABS(CD(4,J))
      KM2=ABS(CD(5,J))
      DSQ1=(CD(6,J))**2
      DSQ2=(CD(7,J))**2
C     ***** SEARCH FOR POLYHEDRA CENTER *****
      DO 935 IM=1,LATM
      K3=ATOMID(IM)/D100K
      if (ian.eq.1) k3=ident(1,im)
      if (ian.eq.2) k3=ident(2,im)
      IF(K3-KM1)935,905,905
```

```
  905 IF(KM2-K3)935,910,910
C     ***** CHECK POLYHEDRA DISTANCE RANGE *****
  910 DO 930 J1=1,4,3
      DSQ=(ATOMS(1,IM)-W1(J1))**2
      IF(DSQ2-DSQ)935,915,915
  915 DSQ=DSQ+(ATOMS(2,IM)-W1(J1+1))**2
      IF(DSQ2-DSQ)935,920,920
  920 DSQ=DSQ+(ATOMS(3,IM)-W1(J1+2))**2
      IF(DSQ2-DSQ)935,925,925
  925 IF(DSQ-DSQ1)935,930,930
  930 CONTINUE
      GO TO 955
  935 CONTINUE
C     ***** END OF POLYHEDRA CHECK *****
  950 CONTINUE
      GO TO 975
C     ***** PREPARE TO DRAW BOND *****
  955 IF(NJ2-10)960,970,970
  960 LNS=MOD(LNS+4,56)
      IF(LNS)965,965,970
  965 IF (NOUT.GE.0)
     &WRITE (NOUT,835)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,837)
  970 CALL BOND(TD1,TD2,JB,NA1,NA2)
  975 CONTINUE
  977 CONTINUE
C     ***** ELIMINATE LOCAL OVERLAP INFORMATION BEFORE RETURNING *****
  980 IF(NJ2-21)985,990,990
  985 CALL LAP500(-1)
  990 if (nj2.eq.22) then
      IF(NQUAD)993,993,991
C     ***** PRINT OUT NUMBER OF BOND QUADRANGLES STORED *****
C     ***** PRINT OUT QUADRANGLE IDENTIFICATION ARRAY *****
  991 IF (NOUT.GE.0)
     &WRITE (NOUT,992)NQUAD,(QUAD(9,J),J=1,NQUAD)
  992 FORMAT(1H0,10X,27HBOND OVERLAP ARRAY CONTAINS,I4,23H BONDS (MAXIMU
     1M IS 599)/  11X,  66HATOM-PAIR NUMBERS IN ARRAY REFER TO SEQUENCE
     2IN SORTED ATOMS ARRAY/(15X,10F10.0))
      end if
  993 RETURN
      END
      SUBROUTINE F900
      DIMENSION X(3),XW(3,5),Y(3),Z(3)
      REAL*8 D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      character*72 tmpti, tmpti2
C     ***** LABELING FUNCTION SUBROUTINE *****
```

```
      D100K=100000.
      ITILT=0
      NJ3=MOD(NJ2,10)
      TH=THETA
      SINTH=SYMB(2,1)
      COSTH=SYMB(1,1)
      ILAST=1
      T2=AIN(2)
      IF(T2-11100.)910,910,905
  905 ILAST=2
  910 DO 925 II=1,ILAST
C     ***** OBTAIN WORKING CARTESIAN COORDINATES *****
      CALL XYZ(AIN(II),XW(1,II),2)
      IF(NG)915,925,915
  925 CALL XYZ(AIN(II),XW(1,II+3),3)
      II=1
C     ***** FIND MEAN REFERENCE POINT *****
      DO 930 J=1,3
      T2=XW(J,ILAST)
      T1=XW(J,1)
      XW(J,3)=T2-T1
  930 X(J)=(T2+T1)*.5
C     ***** PERSPECTIVE SCALING FACTOR *****
      SCAL=1.
      IF(VIEW)940,940,935
  935 SCAL=VIEW/(VIEW-X(3))
  940 T1=AIN(5)
      HGT=SCAL*T1
      IF(NJ2-3)960,950,945
  945 IF(NJ2-6)950,950,960
C     ***** PROJECTED VECTOR BASELINE *****
  950 CALL PLTXY(XW(1,4),V1)
      CALL PLTXY(XW(1,5),V2)
      T1=V2(1)-V1(1)
      T2=V2(2)-V1(2)
      T3=SQRT(T1*T1+T2*T2)
      IF(T3)912,912,955
  955 COSTH=T1/T3
      SINTH=T2/T3
      TH=ARCCOS(COSTH)
      IF(SINTH)958,960,960
  958 TH=-TH
  960 IF(NJ2-13)965,985,985
C     ***** FIND CENTER OF PROJECTED LABEL *****
  965 T6=AIN(6)
      T7=AIN(7)
      Y(1)=SCAL*(X(1)+T6*COSTH-T7*SINTH)+XO(1)
      Y(2)=SCAL*(X(2)+T6*SINTH+T7*COSTH)+XO(2)
      Y(3)=0.
C     ***** CHECK FOR LEGEND RESET *****
      DO 980 J=1,2
      T1=AIN(J+2)
      IF(T1)975,980,970
  970 Y(J)=T1
      GO TO 980
  975 Y(J)=XLNG(J)+T1
  980 CONTINUE
C     ***** SET PARAMETERS FOR INDIVIDUAL FUNCTIONS *****
  985 GO TO(990,995,995,1000,1000,1000,915,1105,1105,915,915,915,1005,10
     105,1005,1005,915),NJ2
  990 T6=17.
      L=AIN(1)/D100K
      ILAST=1
      DXW=0.
      DYW=0.
      GO TO 1030
  995 T6=215.
      ILAST=18
      T1=HGT*24./7.
      DXW=COSTH*T1
      DYW=SINTH*T1
      GO TO 1030
 1000 T6=10+3*(NJ3-4)
      DIST=SQRT(VV(XW(1,3),XW(1,3)))/SCAL1
      GO TO 1030
C     ***** TRUE PERSPECTIVE LABELS *****
 1005 CALL UNITY(XW(1,3),VT(1,1),1)
      IF(ABS(VT(3,1))-.9994)1010,912,912
C     ***** FORM PERSPECTIVE ROTATION MATRIX *****
 1010 CALL NORM(AID(1,3),VT(1,1),VT(1,2),1)
      CALL UNITY(VT(1,2),VT(1,2),1)
      CALL NORM(VT(1,1),VT(1,2),VT(1,3),1)
      DO 1015 J=1,3
 1015 VT(J,4)=X(J)
      ITILT=1
      HGT=AIN(5)
      TH=0.
      Y(3)=X(3)
      TT7=AIN(7)
      Y(2)=X(2)+TT7-HGT*.5
      TT6=AIN(6)
      IF(NJ2-13)1030,1025,1020
C     ***** PERSPECTIVE BOND LABELS *****
 1020 Y(1)=X(1)+TT6-HGT*FLOAT(22+3*(6-NJ3))/7.
      DIST=SQRT(VV(XW(1,3),XW(1,3)))/SCAL1
      GO TO 1050
C     ***** PERSPECTIVE TITLES *****
 1025 Y(1)=X(1)+TT6-HGT*215./7.
      ILAST=18
      DXW=HGT*24./7.
      DYW=0.
      GO TO 1050
 1030 DH=HGT*T6/7.
      DV=HGT*.5
      Y(1)=Y(1)-DH*COSTH+DV*SINTH
      Y(2)=Y(2)-DH*SINTH-DV*COSTH
      Y(3)=0.
C     ***** PLOT VARIOUS LABELS *****
 1050 Z(3)=Y(3)
      XO(3)=Y(3)
      GO TO(1060,1060,1060,1090,1090,1090,915,1105,1105),NJ3
 1060 if (nj3 .eq. 1) go to 1061
c *** if title begins in column 1, center it
      if (title2(1)(1:1) .ne. ' ') then
         do 101 i=1,72
            tmpti(i:i) = ' '
            tmpti2(i:i) = ' '
  101    continue
         do 102 i=1,18
            tmpti(i*4-3:i*4)=title2(i)
  102    continue
         do 103 i=72,1,-1
```

```
            if (tmpti(i:i) .ne. ' ') then
               klast = i
               go to 104
            end if
  103    continue
  104    ioffset = (72 - klast) / 2
         do 105 i=1,klast
            tmpti2(i+ioffset:i+ioffset) = tmpti(i:i)
  105    continue
         do 106 i=1,18
            title2(i) = tmpti2(i*4-3:i*4)
  106    continue
      end if
 1061 DO 1085 I=1,ILAST
      DO 1075 J=1,3,2
      Z(1)=Y(1)+FLOAT(J-2)*DISP*.5
      DO 1075 K=1,3,2
      Z(2)=Y(2)+FLOAT(K-2)*DISP*.5
      IF(NJ3-2)1065,1068,1068
C     ***** PLOT CHEMICAL SYMBOL *****
 1065 CALL SIMBOL(Z(1),Z(2),HGT,CHEM(L),TH,6)
      GO TO 1070
C     ***** PLOT TITLES *****
 1068 CALL SIMBOL(Z(1),Z(2),HGT,TITLE2(I),TH,4)
 1070 IF(DISP)1080,1080,1075
 1075 CONTINUE
 1080 Y(1)=Y(1)+DXW
 1085 Y(2)=Y(2)+DYW
      GO TO 1199
C     ***** PLOT BOND DISTANCE LABELS *****
 1090 I9=NJ3-3
      T9=10.**I9
      DISTR=AINT((DIST*T9)+0.5)/T9 +.0001
      CALL NUMBUR(Y(1),Y(2),HGT,DISTR,TH,I9)
      GO TO 1199
C     ***** PLOT CENTERED SYMBOLS *****
 1105 TT8=AIN(8)
c *** ORTEP-II call
c     CALL SIMBOL(Y(1),Y(2),HGT,IFIX(TT8),TH,7-NJ3)
c *** Only one centered symbol (*) is available in ORTEP-III.
c *** It is triggered by the negative value for argument 6.
c *** Argument 4 is ignored by SIMBOL.
      CALL SIMBOL(Y(1),Y(2),HGT,' ',TH,7-NJ3)
      GO TO 1199
  912 NG=15
  915 CALL ERPNT(AIN(II),NJ*100+NJ2)
 1199 ITILT=0
      RETURN
      END
      SUBROUTINE F1000
c *** 1001 identical to 511
      CALL LAP500(1)
      RETURN
      END
      function iend(string)
c *** returns position of last non-space character in string
      character string*(*)
      do 800 i=len(string),1,-1
         if (string(i:i) .ne. ' ') then
            iend = i
            return
```

```
            end if
  800    continue
         iend = 1
         return
         end
      SUBROUTINE LAP500(NTYPE)
C     ***** STORE PROJECTED ATOM CONICS AND BOND QUADRANGLES *****
      DIMENSION QC(3,3),QD(3,3),VD1(3),VD2(3)
      REAL*8 QC,QD,VD1,VD2,TD1,TD2,TD3,TD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
C     ***** ELIMINATE ALL PREVIOUSLY STORED LOCAL OVERLAP INFORMATION **
      NCOVER=0
      NQOVER=0
      IF(NTYPE)420,195,195
C     ***** ELIMINATE ALL PREVIOUSLY STORED GLOBAL OVERLAP INFORMATION *
  195 NCONIC=0
      NQUAD=0
      IF(NTYPE)420,420,200
C     ***** CONSTANT FOR OVERLAP MARGIN (WHITE MARGIN AT OVERLAP) *****
  200 IF(AIN(1))205,215,210
C     ***** NEGATIVE NUMBER OR POSITIVE INTEGER GIVES OVMRGN=0.0 *****
  205 OVMRGN=0.0
      GO TO 220
C     ***** SET OVERLAP MARGIN WIDTH DIRECTLY IN INCHES *****
  210 OVMRGN=AIN(1)-DINT(AIN(1))
      GO TO 220
C      ***** DEFAULT OPTION, OVERLAP MARGIN WIDTH AS A FUNCTION OF SCAL1
  215 if (scal1.lt..25) then
         OVMRGN=AMAX1(SQRT(SCAL1)*0.050,0.010)
      else
         OVMRGN=AMAX1(SQRT(SCAL1)*0.030,0.025)
      end if
  220 IF (NOUT.GE.0)
     &WRITE (NOUT,2) OVMRGN
    2 FORMAT(1H0,10X,17HOVERLAP MARGIN IS, F6.3,5H INCH)
  225 IF(LATM)230,230,235
  230 NG=12
      CALL ERPNT(0.D0,510+NJ2)
      GO TO 420
C     ***** SORT ATOMS LIST BY -VIEWDISTANCE OR BY Z PARAMETER
  235 IF(VIEW)250,250,240
C     ***** CALCULATE VIEWDISTANCES**2 *(-1) IF VIEW.GT.ZERO *****
  240 DO 245 I=1,LATM
      CALL XYZ(ATOMID(I),V3,2)
      V3(3)=V3(3)-VIEW
  245 ATOMS(3,I)=-VV(V3,V3)
      GO TO 260
C     ***** STORE CARTESIAN COORDINATES IF VIEW.EQ.ZERO *****
  250 DO 255 I=1,LATM
  255 CALL XYZ(ATOMID(I),ATOMS(1,I),2)
```

```
C     ***** SORTING PROCEDURE BY SHELL, COMM ACM 2,30 (1959) *****
  260 M=LATM
  265 M=M/2
      IF(M)300,300,270
  270 K=LATM-M
      J=1
  275 I=J
  280 IM=I+M
      IF(ATOMS(3,I)-ATOMS(3,IM))295,295,285
  285 TD=ATOMID(I)
      ATOMID(I)=ATOMID(IM)
      ATOMID(IM)=TD
      T1=ATOMS(3,I)
      ATOMS(3,I)=ATOMS(3,IM)
      ATOMS(3,IM)=T1
      I=I-M
      IF(I)295,295,280
  295 J=J+1
      IF(J-K)275,275,265
C     ***** LOOP THROUGH ALL ATOMS IN SORTED ATOMS LIST *****
  300 DO 405 IA=1,LATM
      CALL XYZ(ATOMID(IA),ATOMS(1,IA),2)
      CALL PAXES(ATOMID(IA),2)
      DO 305 J=1,3
      V1(J)=ATOMS(J,IA)
      VD1(J)=V1(J)
      DO 305 K=1,3
  305 QD(J,K)=Q(J,K)
      IF(VIEW)340,340,310
C     ***** CALCULATE ENVELOPING CONE WITH ORIGIN AT VIEWPOINT *****
  310 V1(3)=V1(3)-VIEW
      VD1(3)=V1(3)
C     ***** FORM COFACTOR MATRIX *****
      DO 315 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 315 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
      QC(J,K)= QD(J1,K1)*QD(J2,K2)-QD(J1,K2)*QD(J2,K1)
  315 QC(K,J)=QC(J,K)
C     ***** FORM POLARIZED COFACTOR MATRIX AND ADD TO ELLIPSOID MATRIX *
      TD2=-SCL**2
C     ***** TD1 IS AN ARBITRARY SCALING FACTOR *****
      TD1=VMV(V1,Q,V1)
      DO 325 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 320 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
      QD(J,K)=((VD1(J2)*(QC(J1,K1)*VD1(K2)-QC(J1,K2)*VD1(K1))
     1 +VD1(J1)*(VD1(K1)*QC(J2,K2)-VD1(K2)*QC(J2,K1)))+TD2*QD(J,K))/TD1
  320 QD(K,J)=QD(J,K)
C     ***** PROJECTED ELLIPSE IN HOMOGENEOUS COORD OF WORKING SYSTEM ***
      QD(J,3)=-QD(J,3)*VIEW
  325 QD(3,J)=-QD(3,J)*VIEW
C     ***** PROJECT CENTER OF ATOM ONTO PROJECTION PLANE *****
      TD1=-VIEW/VD1(3)
      VD2(1)=VD1(1)*TD1
      VD2(2)=VD1(2)*TD1
C     ***** TRANSFORM TO NEW ORIGIN TO IMPROVE CONDITION OF MATRIX Q ***
      DO 330 J=1,3
      DO 330 K=1,2
  330 QD(J,3)=QD(J,3)+QD(J,K)*VD2(K)
      DO 335 J=1,3
      DO 335 K=1,2
  335 QD(3,J)=QD(3,J)+VD2(K)*QD(K,J)
      V6(1)=XO(1)+VD2(1)
      V6(2)=XO(2)+VD2(2)
      GO TO 355
C     ***** CALCULATE ENVELOPING CYLINDER ALONG Z OF WORKING SYSTEM ****
  340 DO 345 J=1,2
      DO 345 K=1,2
  345 QD(J,K)=QD(J,K)-QD(J,3)*QD(K,3)/QD(3,3)
      DO 350 J=1,2
      QD(J,3)=0.0
      QD(3,J)=0.0
  350 V6(J)=XO(J)+ATOMS(J,IA)
C     ***** PROJECTED ELLIPSE IN HOMOGENEOUS COORD ABOUT CENTER OF ATOM
      QD(3,3)=-SCL**2
C     ***** FIT RECTANGLE AROUND ELLIPSE ALLOWING OVERLAP MARGIN *****
C     ***** FORM MATRIX OF COFACTORS *****
  355 DO 360 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 360 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
  360 QC(J,K)= QD(J1,K1)*QD(J2,K2)-QD(J1,K2)*QD(J2,K1)
C     ***** RESCALE MATRIX OF COFACTORS SO THAT QC(3,3)=1.0 *****
      DO 365 J=1,3
      DO 365 K=J,3
      QC(J,K)= QC(J,K)/QC(3,3)
  365 QC(K,J)=QC(J,K)
      TD2=QD(3,3)
      NDG=0
      DO 385 J=1,2
C     ***** SOLVE QUADRATIC EQUATION *****
      T1 =QC(3,J)**2-QC(J,J)
      IF(T1)370,370,375
C     ***** ROUNDOFF PROBLEMS, RESET LIMITS IN X OR Y *****
  370 NDG=1
      V5(J)=0.001+OVMRGN
      GO TO 380
  375 V5(J)=  SQRT(T1)+OVMRGN
      V6(J)=V6(J)+QC(3,J)
      TD2=TD2+QD(3,J)*QC(3,J)
  380 CONIC(2*J-1,IA)=V6(J)-V5(J)
  385 CONIC(2*J,IA)=V6(J)+V5(J)
      IF(NDG)390,390,395
  390 IF(TD2)400,395,395
C     ***** ELLIPSE IMAGINARY DUE TO ROUNDOFF, RESET TO REAL VALUE *****
  395 CONIC(5,IA)=1.0/((CONIC(2,IA)-CONIC(1,IA))*0.5)**2
      CONIC(6,IA)=0.0
      CONIC(7,IA)=1.0/((CONIC(4,IA)-CONIC(3,IA))*0.5)**2
      GO TO 405
C     ***** STORE NORMALIZED QUADRATIC COEFFICIENTS FOR ELLIPSE *****
C     ***** SCALED BY OVERLAP MARGIN PARAMETER *****
  400 TD3= -(1.0-2.0*OVMRGN/(V5(1)+V5(2)))**2 /TD2
      CONIC(5,IA)=QD(1,1)*TD3
      CONIC(6,IA)=QD(1,2)*TD3
```

```
      CONIC(7,IA)=QD(2,2)*TD3
  405 CONTINUE
      NCONIC=LATM
C     ***** PRINT OUT SORTED ATOMS ARRAY *****
      IF (NOUT.GE.0)
     &WRITE (NOUT,4)(ATOMID(J),J=1,LATM)
    4 FORMAT(1H0,10X,30HCONTENTS OF SORTED ATOMS ARRAY/(15X,10F10.0))
C     ***** STORE BOND QUADRANGLES IF SEARCH CODES ARE GIVEN *****
      IF(NCD)420,420,410
C     ***** GENERATE PSEUDO-INSTRUCTION 822 TO CALCULATE BONDS *****
  410 NJ2=22
      CALL F800
c *** the lines below have been moved to the end of F800
c       IF(NQUAD)420,420,415
c C     ***** PRINT OUT NUMBER OF BOND QUADRANGLES STORED *****
c C     ***** PRINT OUT QUADRANGLE IDENTIFICATION ARRAY *****
c   415 IF (NOUT.GE.0)
c      &WRITE (NOUT,6)NQUAD,(QUAD(9,J),J=1,NQUAD)
c     6 FORMAT(1H0,10X,27HBOND OVERLAP ARRAY CONTAINS,I4,23H BONDS (MAXIMU
c      1M IS 599)/  11X,  66HATOM-PAIR NUMBERS IN ARRAY REFER TO SEQUENCE
c      2IN SORTED ATOMS ARRAY/(15X,10F10.0))
c *** the lines above have been moved to the end of F800
  420 RETURN
      END
      SUBROUTINE LAP700(NA,ICQ)
      DIMENSION DETER(2),QA(3,3,2),QC(3,3,2),V12(3,2),YMIN(2),YMAX(2)
      DIMENSION OVMR(2)
      REAL*8 AOV3,AOV3SQ,BOV3,DETER,PI,PHI,POV3,POV3CU,QA,QC,QOV2,QOV2SQ
      REAL*8 ROOT,TD,TIDD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      PI=3.1415926535897932
      ICQ=0
      NCOVER=0
      NQOVER=0
      OVMR(1)=OVMRGN
      OVMR(2)=0.0
      IF(NCONIC-NA)200,200,205
  200 RETURN
C     ***** ROUGH CHECK FOR OVERLAPPING ATOMS *****
  205 DO  210 J=1,2
      YMIN(J)=CONIC(2*J-1,NA)
  210 YMAX(J)=CONIC(2*J,NA)
      L=0
      DO 420 IA=NA,NCONIC
      IF(IA-NA)230,230,215
  215 DO 225 J=1,2
      IF(YMAX(J)-CONIC(2*J-1,IA))420,420,220
  220 IF(YMIN(J)-CONIC(2*J,IA))225,420,420
  225 CONTINUE
C     ***** EXACT CHECK FOR OVERLAPPING ATOMS *****
  230 IF(L-1)235,235,240
  235 L=L+1
  240 CALL LAPCON(CONIC(1,IA),DA,V12(1,L),OVMR(L))
      DO 245 J=1,3
      DO 245 K=1,3
  245 QA(J,K,L)=DA(J,K)
C     ***** CALCULATE COFACTORS AND DETERMINANTS *****
      DETER(L)=0.0
      DO 250 J=1,3
      J1=MOD(J+3,3)+1
      J2=MOD(J+1,3)+1
      DO 250 K=1,3
      K1=MOD(K+3,3)+1
      K2=MOD(K+1,3)+1
      TD=QA(J1,K1,L)*QA(J2,K2,L)-QA(J1,K2,L)*QA(J2,K1,L)
      DETER(L)=DETER(L)+TD*QA(J,K,L)
  250 QC(J,K,L)=TD
C     ***** DETER(L) IS THE DETERMINANT TIMES 3 *****
      IF(L-1)420,420,255
C     ***** FORM CHARACTERISTIC EQUATION AND EXAMINE ITS ROOTS *****
  255 AOV3=0.0
      BOV3=0.0
      DO 260 J=1,3
      DO 260 K=1,3
      AOV3=AOV3+QC(J,K,2)*QA(J,K,1)
  260 BOV3=BOV3+QC(J,K,1)*QA(J,K,2)
      AOV3=AOV3/ DETER(2)
      AOV3SQ=AOV3**2
      BOV3=BOV3/ DETER(2)
      POV3=BOV3-AOV3SQ
      QOV2=AOV3*(AOV3SQ-BOV3*1.5D0)+DETER(1)/(DETER(2)*2.0D0)
C     ***** CHECK DISCRIMINANT OF CHARACTERISTIC CUBIC EQUATION *****
      ITYPE=0
      POV3CU=POV3**3
      QOV2SQ=QOV2**2
      IF(POV3CU+QOV2SQ)270,310,265
  265 IF(POV3CU*1.00001 +QOV2SQ)310,310,400
  270 IF(POV3CU+1.00001 *QOV2SQ)275,310,310
C     ***** THREE REAL ROOTS, ALL DIFFERENT *****
  275 ITYPE=1
C     ***** NO INTERSECTION IF A/3 AND B/3 INVARIANTS ARE NEGATIVE *****
      IF(AOV3)280,285,285
  280 IF(BOV3)420,285,285
C     ***** CALCULATE ONE ROOT OF CHARACTERISTIC CUBIC EQUATION *****
  285 IF(QOV2)295,290,295
  290 PHI=PI/2.0D0
      GO TO 305
  295 PHI=DATAN(-DSQRT(-POV3CU-QOV2SQ)/QOV2)
      IF(PHI)300,305,305
  300 PHI=PHI+PI
  305 ROOT=2.0D0*DSQRT(-POV3)*DCOS(PHI/3.0D0)-AOV3
      GO TO 325
C     ***** THREE REAL ROOTS, AT LEAST TWO ARE EQUAL *****
  310 ITYPE=2
C     ***** CHECK SIGNS OF INVARIANTS A/3 AND B/3 *****
      IF(AOV3)315,320,320
  315 IF(BOV3)420,320,320
C     ***** CALCULATE REPEATED ROOT OF CUBIC EQUATION *****
  320 ROOT=DSIGN(DSQRT(-POV3),QOV2)-AOV3
C     ***** FORM DEGENERATE CONIC (LINE PAIR WHICH MAY BE COINCIDENT) **
  325 DO 330 J=1,3
```

```
      DO 330 K=1,3
  330 DA(J,K)=QA(J,K,1)+ROOT*QA(J,K,2)
C     ***** EXAMINE INVARIANTS OF THE DEGENERATE CONIC *****
      T6=DA(1,1)*DA(2,2)
      T7=DA(1,2)**2
C     ***** NEGATIVE DENOTES REAL INTERSECTING LINE PAIR *****
C     ***** POSITIVE DENOTES IMAGINARY LINES INTERSECTING AT REAL POINT
      IF(T6-T7)335,345,340
  335 IF(T6*1.0001  -T7)400,345,345
  340 IF(T6-1.0001  *T7)345,345,365
  345 T8=DA(3,3)*(DA(1,1)+DA(2,2))
      T9=DA(1,3)**2+DA(2,3)**2
C     ***** NEGATIVE DENOTES REAL PARALLEL LINE PAIR *****
C     ***** POSITIVE DENOTES IMAGINARY PARALLELS *****
C     ***** ZERO DENOTES ONE REAL LINE (COINCIDENT PARALLELS) *****
      IF(T8-T9)350,360,355
  350 IF(T8*1.0001  -T9)400,360,360
  355 IF(T8-1.0001  *T9)360,360,365
C     ***** COINCIDENT LINE PAIR FOUND FOR THE REPEATED ROOT *****
  360 ITYPE=3
C     ***** COMPARE AREAS OF CONICS *****
  365 KA=1
      KB=2
      IF(QC(3,3,KA)-QC(3,3,KB))370,375,375
  370 KA=2
      KB=1
C     ***** SEE IF ONE CONIC IS INSIDE THE OTHER CONIC *****
  375 T1=0.0
      DO 385 J=1,3
      T2=QA(J,3,KB)
      DO 380 K=1,2
  380 T2=T2+QA(J,K,KB)*V12(K,KA)
  385 T1=T1+V12(J,KA)*T2
C     ***** DISCARD IF KA IS OUTSIDE KB *****
      IF(T1)390,390,420
  390 IF(KA-1)395,395,400
C     ***** THE OVERLAPPING ATOM HIDES THE ORIGINAL ATOM *****
  395 ICQ=-1
      RETURN
C     ***** STORE OVERLAPPING ATOM *****
  400 ICQ=ICQ+1
      IF(NCOVER-20)410,405,405
  405 NG=17
      CALL ERPNT(ATOMID(IA),700)
      NCOVER=NCOVER-1
  410 NCOVER=NCOVER+1
      IJ=1
      DO 415 I=1,3
      DO 415 J=I,3
      COVER(IJ,NCOVER)=QA(I,J,2)
  415 IJ=IJ+1
      KC(NCOVER)=IA
  420 CONTINUE
C     ***** SECOND PART OF SUBROUTINE CHECKS FOR BONDS OVER THE ATOM ***
  425 IF(NQUAD)470,470,430
  430 ITY=0
C     ***** ROUGH CHECK FOR OVERLAPPING BONDS *****
      DO 465 IQ=1,NQUAD
      TID=QUAD(9,IQ)
      TIDD=TID
      NA1=TID/1000.
      NA2=AMOD(TID,1000.)
      IF(NA-NA2)435,435,465
  435 DO 445 J=1,2
      IF(YMAX(J)-AMIN1(QUAD(J,IQ),QUAD(J+2,IQ),QUAD(J+4,IQ),QUAD(J+6,IQ)
     1 ))465,465,440
  440 IF(YMIN(J)-AMAX1(QUAD(J,IQ),QUAD(J+2,IQ),QUAD(J+4,IQ),QUAD(J+6,IQ)
     1 ))445,465,465
  445 CONTINUE
C     ***** EXACT CHECK FOR OVERLAPPING BONDS *****
  450 ITY=ITY-1
      IQQ=0
      IQR=IQ
      CALL LAPAB(IQR,NA,IQQ,ITY)
      IF(IQQ)455,460,460
  455 ICQ=-1
      RETURN
  460 ICQ=ICQ+1
      IF(NQOVER-30)465,470,470
  465 CONTINUE
  470 RETURN
      END
      SUBROUTINE LAP800(NA1,NA2,ICQ)
C     ***** SUBROUTINE CHECKS FOR ATOMS AND BONDS OVERLAPPING A BOND ***
      DIMENSION FL(4,4),Y1(2),Y2(2),YMAX(2),YMIN(2),QUA(3,4)
      DIMENSION VUE(3)
      REAL*8 TIDD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IQ=0
      ICQ=0
      IF(NA1*NA2)245,245,195
  195 TID1=FLOAT(NA1)*1000.+FLOAT(NA2)
      IF(NCONIC)245,245,200
  200 IF(NJ2-21)250,205,205
C     ***** PART 1, CALLED FROM BOND, STORES BOND OUTLINE INFORMATION **
  205 IF(NQUAD-599)215,210,210
  210 NG=16
      CALL ERPNT(ATOMID(NA1),822)
      GO TO 245
  215 NQUAD=NQUAD+1
C     ***** CALCULATE OVERLAP MARGIN FOR BOND QUADRANGLE *****
      T1=0.0
      T2=0.0
      DO 220 J=1,2
      Y1(J)=DP(J,1)-DP(J,65)
      Y2(J)=DP(J,2)-DP(J,66)
      T1=T1+Y1(J)**2
  220 T2=T2+Y2(J)**2
      IF(T1*T2)225,225,230
  225 T1=0.0
      T2=0.0
```

```
      GO TO 235
  230 T1=OVMRGN/SQRT(T1)
      T2=OVMRGN/SQRT(T2)
C     ***** STORE BOND QUADRANGLE *****
  235 DO 240 J=1,2
      Y1(J)=Y1(J)*T1
      Y2(J)=Y2(J)*T2
      QUAD(J,NQUAD)=DP(J,1)+Y1(J)
      QUAD(J+2 ,NQUAD)=DP(J,2)+Y2(J)
      QUAD(J+4,NQUAD)=DP(J,66)-Y2(J)
  240 QUAD(J+6,NQUAD)=DP(J,65)-Y1(J)
      QUAD(9,NQUAD)=TID1
  245 RETURN
C     ***** PART 2, CALLED FROM BOND, OVERLAP CHECK FOR BOND NA1-NA2 ***
  250 NCOVER=0
      NQOVER=0
      TOL=1.E-5
      IF(NCONIC-NA1)245,245,255
C     ***** SAVE QUADRANGLE TEMPORARILY *****
  255 IQ=NQUAD+1
      DO 260 J=1,2
      QUAD(J,IQ)=DP(J,1)
      QUAD(J+2,IQ)=DP(J,2)
      QUAD(J+4,IQ)=DP(J,66)
  260 QUAD(J+6,IQ)=DP(J,65)
      QUAD(9,IQ)=TID1
C     ***** FIT RECTANGLE AROUND QUADRANGLE *****
  265 DO 270 J=1,2
      YMIN(J)=AMIN1(DP(J,1),DP(J,2),DP(J,66),DP(J,65))
  270 YMAX(J)=AMAX1(DP(J,1),DP(J,2),DP(J,66),DP(J,65))
C     ***** ROUGH CHECK FOR ATOM-OVER-BOND OVERLAP *****
      NA1P1=NA1+1
      ITY=0
      DO 305 IA=NA1P1,NCONIC
      DO 285 J=1,2
      IF(IA-NA2)275,305,275
  275 IF(YMAX(J)-CONIC(2*J-1,IA))305,305,280
  280 IF(YMIN(J)-CONIC(2*J,IA))285,305,305
  285 CONTINUE
C     ***** CHECK FOR TRUE ATOM-OVER-BOND OVERLAP *****
      ITY=ITY+1
      IAQ=IA
      CALL LAPAB(IQ,IAQ,IQQ,ITY)
      IF(IQQ)290,305,300
  300 ICQ=ICQ+1
      IF(NCOVER-20)305,310,310
  305 CONTINUE
  310 IF(NQUAD)295,295,315
C     ***** HIDDEN BOND *****
  290 ICQ=-1
  295 RETURN
C     ***** ROUGH CHECK FOR BOND-OVER-BOND OVERLAP *****
  315 CALL DIFV(ATOMS(1,NA2),ATOMS(1,NA1),V1)
      CALL UNITY(V1,V1,1)
      VUE(1)=  ATOMS(1,NA1)
      VUE(2)=  ATOMS(2,NA1)
      VUE(3)=  ATOMS(3,NA1)-VIEW
      DO 495 IB=1,NQUAD
      TID2=QUAD(9,IB)
      IF(TID1-TID2)320,495,320
  320 NB2=AMOD(TID2,1000.)
      NB1=TID2/1000.
      IF(NA1-NB2)325,495,495
  325 DO 335 J=1,2
      IF(YMAX(J)-AMIN1(QUAD(J,IB),QUAD(J+2,IB),QUAD(J+4,IB),QUAD(J+6,IB)
     1 ))495,495,330
  330 IF(YMIN(J)-AMAX1(QUAD(J,IB),QUAD(J+2,IB),QUAD(J+4,IB),QUAD(J+6,IB)
     1 ))335,495,495
  335 CONTINUE
C     ***** SET UP LINEAR FORMS FOR EDGES OF QUADRANGLE *****
      DO 345 L=1,4
      K=2*L
      K1=MOD(K,8)+2
      QUA(1,L)=QUAD(K,IB)-QUAD(K1,IB)
      QUA(2,L)=QUAD(K1-1,IB)-QUAD(K-1,IB)
      QUA(3,L)=QUAD(K-1,IB)*QUAD(K1,IB)-QUAD(K,IB)*QUAD(K1-1,IB)
C     ***** NORMALIZE LINE EQUATION COEFFICIENTS *****
      T1=SQRT(QUA(1,L)**2+QUA(2,L)**2)
      IF(T1)495,495,340
  340 DO 345 J=1,3
  345 QUA(J,L)=QUA(J,L)/T1
C     ***** EVALUATE LINEAR FORMS AND SIGNATURES FOR QUADRANGLE *****
      T3=3.0
      DO 365 K=1,4
      T2=3.0
      J=K*2
      DO 355 L=1,4
      T1=QUAD(J-1,IQ)*QUA(1,L)+QUAD(J,IQ)*QUA(2,L)+QUA(3,L)
      IF(T1)350,355,355
  350 T2=T2-1.0
  355 FL(L,K)=T1
      IF(T2)360,365,365
  360 T3=T3-1.0
  365 CONTINUE
C     ***** CHECK FOR 4 POINTS INSIDE QUADRANGLE *****
      IF(T3)370,375,375
  370 ITYPE=-1
      GO TO 415
C     ***** CHECK FOR 1 TO 3 POINTS INSIDE QUADRANGLE ****
  375 IF(T3-3.0)380,385,385
  380 ITYPE=0
      GO TO 415
C     ***** DETERMINE WHICH EDGES ARE CROSSED BY THE 4 LINE SEGMENTS ***
  385 DO 405 L=1,4
      L1=MOD(L,4)+1
C     ***** LINE SEGMENT L FROM POINT Y1 TO POINT Y2 *****
      Y1(1)=QUAD(L*2-1,IQ)
      Y1(2)=QUAD(L*2,IQ)
      Y2(1)=QUAD(L1*2-1,IQ)
      Y2(2)=QUAD(L1*2,IQ)
      DO 405 K=1,4
      T1=FL(K,L)
      T2=FL(K,L1)
      T3=T1-T2
C     ***** T1 AND T2 MUST HAVE OPPOSITE SIGNS FOR INTERSECTION TO OCCUR
      IF(T1*T2)390,390,405
C     ***** COMPONENT OF SEGMENT L PERPENDICULAR TO EDGE K OF IB IS T3
  390 IF(ABS(T3)-1.E-5)405,405,395
C     ***** CALCULATE COORDINATES OF INTERSECTION *****
  395 T4=(T1*Y2(1)-T2*Y1(1))/T3
      T5=(T1*Y2(2)-T2*Y1(2))/T3
      K0=2*K
```

```
      K1=2*(MOD(K,4)+1)
C     ***** IS INTERSECTION WITHIN QUADRANGLE IQ *****
      T6=(T4-QUAD(K0-1,IB))*(QUAD(K1-1,IB)-T4)+(T5-QUAD(K0,IB))*
     1 (QUAD(K1,IB)-T5)
      IF(ABS(T6)-1.E-4)410,410,400
  400 IF(T6)405,410,410
  405 CONTINUE
      GO TO 495
  410 ITYPE=1
C     ***** CHECK OVER/UNDER AMBIGUITY *****
  415 IF((NA1-NB1)*(NA2-NB2)*(NA2-NB1))425,420,425
C     ***** BONDS SHARE AN ATOM *****
  420 IF(NA1+NA2-NB1-NB2)465,495,495
  425 CALL DIFV(ATOMS(1,NB2),ATOMS(1,NB1),V2)
      CALL DIFV(ATOMS(1,NB1),ATOMS(1,NA1),V4)
      CALL UNITY(V2,V2,1)
      CALL UNITY(V4,V4,1)
      CALL NORM(V1,V2,V3,1)
      IF(VV(V3,V3)-TOL)430,430,435
C     ***** PARALLEL BONDS, RECALCULATE V3 *****
  430 CALL NORM(V1,V4,V5,1)
      CALL NORM(V5,V1,V3,1)
C     ***** CHECK FOR COLLINEAR BONDS *****
      IF(VV(V3,V3)-TOL)465,465,450
  435 IF(VV(V3,V4)+TOL)440,450,450
  440 DO 445 J=1,3
  445 V3(J)=-V3(J)
C     ***** V3 IS NORMAL TO BONDS IQ AND IB GOING FROM IQ TOWARD IB ***
  450 IF(VIEW)455,455,460
  455 IF(V3(3))495,495,465
  460 IF(VV(VUE,V3))465,495,495
C     ***** OVERLAPPING BOND FOUND *****
  465 ICQ=ICQ+1
      IF(ITYPE)470,475,475
C     ***** HIDDEN BOND *****
  470 ICQ=-1
      RETURN
C     ***** STORE INTERFERING QUADRANGLE *****
  475 IF(NQOVER-30)485,480,480
  480 NG=18
      TIDD=TID2
      CALL ERPNT(TIDD,800)
      RETURN
  485 NQOVER=NQOVER+1
      DO 490 K=1,4
      DO 490 J=1,3
  490 QOVER(J,K,NQOVER)=QUA(J,K)
      KQ(NQOVER)=IB
  495 CONTINUE
  500 RETURN
      END
      SUBROUTINE LAPAB(IQ,IA,ICQ,ITY)
C     ***** SUBROUTINE CHECKS FOR OVERLAP BETWEEN ATOMS AND BONDS *****
C     ***** CALLED BY SUBROUTINES LAP700 AND LAP800 *****
      DIMENSION BF(4),CON(3,3),QF(5),QUA(3,4)
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 TIDD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      TID=QUAD(9,IQ)
      NA1=TID/1000.
      NA2=AMOD(TID,1000.)
C     ***** ITY.GT.0, CHECK FOR ATOMS OVER A BOND ****
C     ***** ITY.LT.0, CHECK FOR BONDS OVER AN ATOM *****
      ICQ=0
      IF(ITY)210,200,205
  200 RETURN
  205 CALL LAPCON(CONIC(1,IA),CON,V1,0.0)
      IF(ITY-2)220,240,240
  210 IF(ITY+2)220,220,215
  215 CALL LAPCON(CONIC(1,IA),CON,V1,OVMRGN)
C     ***** SET UP LINEAR FORMS FOR EDGES OF QUADRANGLE *****
  220 DO 235 L=1,4
      K=2*L
      K1=MOD(K,8)+2
      QUA(1,L)=QUAD(K,IQ)-QUAD(K1,IQ)
      QUA(2,L)=QUAD(K1-1,IQ)-QUAD(K-1,IQ)
      QUA(3,L)=QUAD(K-1,IQ)*QUAD(K1,IQ)-QUAD(K,IQ)*QUAD(K1-1,IQ)
      T1=SQRT(QUA(1,L)**2+QUA(2,L)**2)
      IF(T1)225,225,230
  225 ITY=0
      ICQ=0
      GO TO 430
C     ***** TRANSFORM COEFFICIENTS FOR EDGES TO NORMAL FORM *****
  230 DO 235 J=1,3
  235 QUA(J,L)=QUA(J,L)/T1
C     ***** EVALUATE 4 QUADRATIC AND 4 BILINEAR FORMS *****
  240 V2(3)=1.0
      V3(3)=1.0
      T2=3.0
      DO 265 L=1,4
      L1=(MOD(L,4)+1)*2
      V2(1)=QUAD(2*L-1,IQ)
      V2(2)=QUAD(2*L,IQ)
      V3(1)=QUAD(L1-1,IQ)
      V3(2)=QUAD(L1,IQ)
      QF(L)=0.0
      BF(L)=0.0
      DO 250 K=1,3
      T1=CON(3,K)
      DO 245 J=1,2
  245 T1=T1+V2(J)*CON(J,K)
      QF(L)=QF(L)+T1*V2(K)
  250 BF(L)=BF(L)+T1*V3(K)
      IF(QF(L))260,255,265
  255 T2=T2-0.8
      GO TO 265
  260 T2=T2-1.0
  265 CONTINUE
      QF(5)=QF(1)
C     ***** CHECK FOR 4 POINTS OF QUADRANGLE INSIDE OR ON ELLIPSE *****
      IF(T2)270,275,275
  270 ITYPE=-1
```

```
      GO TO 330
C     ***** CHECK FOR 1 TO 3 POINTS OF QUADRANGLE INSIDE THE ELLIPSE ***
  275 IF(T2-2.2)280,285,285
  280 ITYPE=0
      IF(NA2-IA)340,375,335
C     ***** CHECK FOR QUADRANGLE-ELLIPSE INTERSECTION *****
  285 DO 305 K=1,4
C     ***** EVALUATE DISCRIMINANT *****
      T1=BF(K)**2-QF(K)*QF(K+1)
      IF(T1)305,305,290
  290 T1=SQRT(T1)
C     ***** IS INTERSECTION WITHIN BOUNDS OF QUADRANGLE *****
      T3=QF(K)-BF(K)
      T4=T3+QF(K+1)-BF(K)
      IF(ABS(T4)-1.E-5)305,305,295
  295 T5=(T3-T1)/T4
      IF(T5)305,280,300
  300 IF(1.0-T5)305,305,280
  305 CONTINUE
C     ***** NO VALID INTERSECTION FOUND *****
C     ***** CHECK FOR CENTER OF ELLIPSE WITHIN THE QUADRANGLE ****
      T3=3.0
      DO 320 K=1,4
      T1=QUA(3,K)
      DO 310 J=1,2
  310 T1=T1+V1(J)*QUA(J,K)
      IF(T1)315,320,320
  315 T3=T3-1.0
  320 CONTINUE
      IF(T3)325,370,370
  325 ITYPE=1
C     ***** CHECK OVER/UNDER AMBIGUITY *****
  330 IF(NA2-IA)375,375,335
  335 IF(IA-NA1)375,375,340
  340 CALL DIFV(ATOMS(1,NA2),ATOMS(1,NA1),V2)
      CALL DIFV(ATOMS(1,IA),ATOMS(1,NA1),V3)
      CALL UNITY(V2,V2,1)
      CALL UNITY(V3,V3,1)
      CALL NORM(V2,V3,V4,1)
      IF(VV(V4,V4)-1.E-5)345,345,350
C     ***** CENTER OF ATOM IQ IS ON THE BOND LINE *****
  345 IF(ITY)370,370,385
C     ***** CENTER OF ATOM IQ IS NOT ON THE BOND LINE *****
  350 CALL NORM(V4,V2,V5,1)
      T1=-V5(3)
      IF(VIEW)365,365,355
  355 T1=V5(3)*(ATOMS(3,IA)-VIEW)
      DO 360 J=1,2
  360 T1=T1+V5(J)*ATOMS(J,IA)
  365 IF(T1*FLOAT(ITY))375,375,370
C     ***** NO INTERFERENCE FOUND *****
  370 ICQ=0
      GO TO 430
C     ***** ITYPE=1 ENCLOSED ELLIPSE / ITYPE=-1 ENCLOSED QUADRANGLE ****
  375 IF(ITYPE*ITY)380,385,385
C     ***** HIDDEN ATOM OR HIDDEN BOND *****
  380 ICQ=-1
      GO TO 430
  385 ICQ=1
      IF(ITY)410,390,390
C     ***** STORE INTERFERING ELLIPSE *****
  390 IF(NCOVER-20)400,395,395
  395 NG=17
      CALL ERPNT(ATOMID(IA),800)
      NCOVER=NCOVER-1
  400 NCOVER=NCOVER+1
      IJ=1
      DO 405 I=1,3
      DO 405 J=I,3
      COVER(IJ,NCOVER)=CON(I,J)
  405 IJ=IJ+1
      KC(NCOVER)=IA
      GO TO 430
C     ***** STORE INTERFERING QUADRANGLE *****
  410 IF(NQOVER-30)420,415,415
  415 NG=18
      TIDD=TID
      CALL ERPNT(TIDD,700)
      NQOVER=NQOVER-1
  420 NQOVER=NQOVER+1
      DO 425 K=1,4
      DO 425 J=1,3
  425 QOVER(J,K,NQOVER)=QUA(J,K)
      KQ(NQOVER)=IQ
  430 RETURN
      END
      SUBROUTINE LAPCON(CON1,CON,Y,OVMR)
C     ***** TRANSFORM CONIC TO PLOTTER HOMOGENEOUS COORDINATE SYSTEM ***
C     ***** CALLED BY SUBROUTINES LAP700 AND LAPAB *****
      DIMENSION CON1(7),CON(3,3),Y(3)
      Y(1)=(CON1(1)+CON1(2))*0.5
      Y(2)=(CON1(3)+CON1(4))*0.5
      Y(3)=1.0
      CON(1,1)=CON1(5)
      CON(1,2)=CON1(6)
      CON(2,1)=CON1(6)
      CON(2,2)=CON1(7)
      T1=(CON1(2)-CON1(1)+CON1(4)-CON1(3))*0.25
      CON(3,3)=-((T1-OVMR)/T1)**2
      DO 205 K=1,2
      CON(K,3)=0.0
      DO 200 J=1,2
  200 CON(K,3)=CON(K,3)-Y(J)*CON(J,K)
      CON(3,K)=CON(K,3)
  205 CON(3,3)=CON(3,3)-CON(3,K)*Y(K)
      RETURN
      END
      SUBROUTINE LAPDRW(Y,NPEN,NCQ)
C     ***** SUBROUTINE ELIMINATES HIDDEN LINES AND DRAWS VISIBLE LINES *
      DIMENSION CB(20),CQ(50,2),QL(4,30,2),SEG(2),Y(3),YN(3),YO(3),Z(3)
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
```

```
      NCQ=NCOVER+NQOVER
      IF(NCQ)200,200,205
  200 RETURN
C     ***** CHECK ALL OVERLAPPING ATOMS AND BONDS *****
  205 NPM3=NPEN-3
      IF(NPM3)  210,230,230
C     ***** SAVE INFORMATION FROM LAST POINT IF PEN IS DOWN *****
  210 YO(1)=YN(1)
      YO(2)=YN(2)
      YO(3)=1.0
      NPO=NPN
      DO 215 K=1,NCQ
  215 CQ(K,1)=CQ(K,2)
      IF(NQOVER)230,230,220
  220 DO 225 K=1,NQOVER
      DO 225 J=1,4
  225 QL(J,K,1)=QL(J,K,2)
C     ***** EVALUATE CONIC QUADRATIC FORMS AT NEW POINT YN *****
  230 YN(1)=Y(1)
      YN(2)=Y(2)
      YN(3)=1.0
      NPN=NPEN
      IF(NCOVER)250,250,235
  235 DO 245 K=1,NCOVER
      Z(1)=YN(1)*COVER(1,K)+YN(2)*COVER(2,K)+COVER(3,K)
      Z(2)=YN(1)*COVER(2,K)+YN(2)*COVER(4,K)+COVER(5,K)
      Z(3)=YN(1)*COVER(3,K)+YN(2)*COVER(5,K)+COVER(6,K)
      CQ(K,2)=Z(1)*YN(1)+Z(2)*YN(2)+Z(3)
C     ***** EVALUATE CONIC BILINEAR FORM IF PEN IS DOWN *****
      IF(NPM3)240,245,245
  240 CB(K)= Z(1)*YO(1)+Z(2)*YO(2)+Z(3)
  245 CONTINUE
C     ***** EVALUATE LINEAR FORMS AND SIGNATURE FOR QUADRANGLE *****
  250 IF(NQOVER)275,275,255
  255 KCQ=NCOVER
      DO 270 K=1,NQOVER
      T2=3.0
      DO 265 J=1,4
      T1=YN(1)*QOVER(1,J,K)+YN(2)*QOVER(2,J,K)+QOVER(3,J,K)
      IF(T1)260,265,265
  260 T2=T2-1.0
  265 QL(J,K,2)=T1
      KCQ=KCQ+1
C     ***** T2=-1 INSIDE, =0 ACROSS ANY EDGE, =1 ACROSS ANY VERTEX *****
  270 CQ(KCQ,2)=T2
C     ***** IF PEN IS UP, OMIT ALL SUBSEQUENT CHECKING *****
  275 IF(NPM3)285,280,280
  280 NPN=3
      CALL SCRIBE(YN,NPN)
      RETURN
C     ***** CHECK FOR HIDDEN SEGMENT *****
  285 DO 295 K=1,NCQ
      IF(CQ(K,1))290,295,295
  290 IF(CQ(K,2))280,295,295
  295 CONTINUE
C     ***** FIND ENTRY AND EXIT POINTS ON EACH CONIC *****
      NINT=0
      IF(NCOVER)330,330,300
  300 DO 325 K=1,NCOVER
C     ***** EVALUATE DISCRIMINANT *****
      T1=CB(K)**2-CQ(K,1)*CQ(K,2)
      IF(T1)325,325,305
  305 T1=SQRT(T1)
C     ***** SOLVE QUADRATIC EQATION *****
      T2=CQ(K,1)-CB(K)
      T3=T2+CQ(K,2)-CB(K)
      IF(ABS(T3)-1.E-5)325,325,310
  310 T4=(T2-T1)/T3
      T5=(T2+T1)/T3
C     ***** VALID INTERSECTION IF T4.LT.1 AND T5.GT.0 *****
      IF(T4-1.0)315,325,325
  315 IF(T5)325,325,320
C     ***** SAVE VALID CONIC INTERSECTIONS *****
  320 NINT=NINT+1
      SEGM(NINT,1)=T4
      SEGM(NINT,2)=T5
  325 CONTINUE
  330 IF(NQOVER)425,425,335
C     ***** FIND ENTRY AND EXIT POINTS FOR EACH QUADRANGLE *****
  335 DO 420 K=1,NQOVER
      I12=0
      KCQ=NCOVER+K
C     ***** CHECK FOR SINGLE INSIDE POINT *****
      SEG(1)=CQ(KCQ,1)
      IF(SEG(1))345,340,340
  340 SEG(1)=1.0-CQ(KCQ,2)
      IF(SEG(1)-1.0)350,350,345
C     ***** INSIDE POINT FOUND, ONLY ONE INTERSECTION POSSIBLE *****
  345 I12=1
C     ***** FIND WHICH EDGES ARE CROSSED BY THE SEGMENT *****
  350 DO 410 J=1,4
      T1=QL(J,K,1)
      T2=QL(J,K,2)
      T3=T1-T2
      IF(T1*T2)355,355,410
C     ***** CHECK FOR SEGMENT ON AN EDGE *****
  355 IF(ABS(T3)-1.E-5)420,420,360
C     ***** CALCULATE COORDINATES OF INTERSECTION *****
  360 T4=(T1*YN(1)-T2*YO(1))/T3
      T5=(T1*YN(2)-T2*YO(2))/T3
      J1=2*(MOD(J,4)+1)
      IQ=KQ(K)
C     ***** IS INTERSECTION WITHIN LIMITS OF QUADRANGLE *****
      T6=(T4-QUAD(2*J-1,IQ))*(QUAD(J1-1,IQ)-T4)+(T5-QUAD(2*J,IQ))*
     1 (QUAD(J1,IQ)-T5)
      IF(ABS(T6)-1.E-4)370,370,365
  365 IF(T6)410,370,370
C     ***** CALCULATE FRACTION PARAMETER AND STORE IT *****
  370 T1=T1/T3
      IF(I12-1)375,380,395
C     ***** STORE FIRST INTERSECTION *****
  375 I12=1
      GO TO 390
C     ***** STORE SECOND INTERSECTION ****
  380 I12=2
      IF(T1-SEG(1))385,405,405
  385 SEG(2)=SEG(1)
  390 SEG(1)= T1
      GO TO 410
C     ***** MORE THAN TWO INTERSECTIONS (I.E.,QUADRANGLE DIAGONAL) *****
  395 IF(T1-SEG(1))390,410,400
  400 IF(T1-SEG(2))410,410,405
```

```
  405 SEG(2)=T1
  410 CONTINUE
      IF(I12-1)420,420,415
C     ***** STORE FRACTION PARAMETERS *****
  415 NINT=NINT+1
      SEGM(NINT,1)=SEG(1)
      SEGM(NINT,2)=SEG(2)
  420 CONTINUE
C     ***** END OF ENTRY-AND-EXIT-POINT CALCULATIONS *****
  425 IF(NINT-1)430,490,435
C     ***** NO INTERFERENCE FOUND, DRAW ENTIRE SEGMENT *****
  430 CALL SCRIBE(YN,2)
      RETURN
C     **** SORT SEGMENT INTERSECTION LIST *****
C     ***** SORTING PROCEDURE BY SHELL,D.L. COMM. ACM 2,30-32 (1959) ***
  435 M=NINT
  440 M=M/2
      IF(M)490,490,445
  445 K=NINT-M
      J=1
  450 I=J
  455 IM=I+M
      IF(SEGM(I,1))460,470,470
  460 IF(SEGM(IM,1))465,465,485
  465 IF(SEGM(I,2)-SEGM(IM,2))485,485,475
  470 IF(SEGM(I,1)-SEGM(IM,1))485,485,475
  475 DO 480 L=1,2
      T1=SEGM(I,L)
      SEGM(I,L)=SEGM(IM,L)
  480 SEGM(IM,L)=T1
      I=I-M
      IF(I)485,485,455
  485 J=J+1
      IF(J-K)450,450,440
C     ***** FIND STARTING POINT P0 AND END POINT P1 *****
  490 P0=0.0
      K=0
  495 K=K+1
      IF(K-NINT)500,500,515
  500 P1=SEGM(K,1)
      IF(P1)510,505,505
  505 IF(P1-P0)510,510,520
  510 P0=AMAX1(P0,SEGM(K,2))
      IF(P0-1.0)495,530,525
  515 P1=1.0
C     ***** DRAW SEGMENT FROM P0 TO P1 *****
  520 IF(P0)535,535,530
  525 P0=1.0
  530 Z(1)=YO(1)*(1.-P0)+YN(1)*P0
      Z(2)=YO(2)*(1.-P0)+YN(2)*P0
      NPN=3
      CALL SCRIBE(Z,NPN)
      IF(P0-1.0)535,540,540
  535 Z(1)=YO(1)*(1.-P1)+YN(1)*P1
      Z(2)=YO(2)*(1.-P1)+YN(2)*P1
      NPN=2
      CALL SCRIBE(Z,NPN)
      IF(P1-1.0)510,540,540
  540 RETURN
      END
      character*(*) function maksym(k,gp)
```

```
c *** returns character string representation of symmetry operator
      dimension gp(3,4,192)
      character*1 xyz(3)
      character*5 fract(23)
      character*12 part(3)
      data fract/'1/24','1/12','1/8','1/6','5/24','1/4','7/24','1/3',
     *           '3/8','5/12','11/24','1/2','13/24','7/12','5/8','2/3',
     *           '17/24','3/4','19/24','5/6','7/8','11/12','23/24'/
      data xyz/'x','y','z'/
      do 200 i=1,3
         part(i) = ' '
         iff = 0
         do 300 j=1,3
            if (ifix(gp(i,j,k)) .ne. 0) then
               if (ifix(gp(i,j,k)) .eq. -1)
     *            part(i) = part(i)(1:iend(part(i))) // '-' // xyz(j)
               if (ifix(gp(i,j,k)) .eq. 1 .and. iff .eq. 0)
     *            part(i) = part(i)(1:iend(part(i))) // ' ' // xyz(j)
               if (ifix(gp(i,j,k)) .eq. 1 .and. iff .eq. 1)
     *            part(i) = part(i)(1:iend(part(i))) // '+' // xyz(j)
               iff = 1
            end if
  300    continue
         gpval = gp(i,4,k)
         if(gpval.gt..01 .or. gpval.lt.-.01) then
            if (gpval.lt.0.) then
               part(i) = part(i)(1:iend(part(i))) // '-'
            else
               part(i) = part(i)(1:iend(part(i))) // '+'
            end if
            gpval=abs(gpval)
            tfour=1./24.
            do 301 mm=1,23
               tf=float(mm)*tfour
               if (gpval.gt.(tf-.01) .and. gpval.lt.(tf+.01)) iw=mm
  301       continue
            part(i) = part(i)(1:iend(part(i))) // fract(iw)
         end if
  200 continue
      maksym = part(1) // part(2) // part(3)
      return
      end
      SUBROUTINE MM(X,Y,Z)
C     MULTIPLY TWO MATRICES
C     Z(3,3)=X(3,3)*Y(3,3)
      DIMENSION X(3,3),Y(3,3),Z(3,3)
      X11=X(1,1)
      X12=X(1,2)
      X13=X(1,3)
      X21=X(2,1)
      X22=X(2,2)
      X23=X(2,3)
      X31=X(3,1)
      X32=X(3,2)
      X33=X(3,3)
      Y11=Y(1,1)
      Y12=Y(1,2)
      Y13=Y(1,3)
      Y21=Y(2,1)
      Y22=Y(2,2)
      Y23=Y(2,3)
```

```
      Y31=Y(3,1)
      Y32=Y(3,2)
      Y33=Y(3,3)
      Z(1,1)=X11*Y11+X12*Y21+X13*Y31
      Z(2,1)=X21*Y11+X22*Y21+X23*Y31
      Z(3,1)=X31*Y11+X32*Y21+X33*Y31
      Z(1,2)=X11*Y12+X12*Y22+X13*Y32
      Z(2,2)=X21*Y12+X22*Y22+X23*Y32
      Z(3,2)=X31*Y12+X32*Y22+X33*Y32
      Z(1,3)=X11*Y13+X12*Y23+X13*Y33
      Z(2,3)=X21*Y13+X22*Y23+X23*Y33
      Z(3,3)=X31*Y13+X32*Y23+X33*Y33
      RETURN
      END
      SUBROUTINE MV(X,Y,Z)
C     MATRIX * VECTOR
C     Z(3)=X(3,3)*Y(3)
      DIMENSION X(3,3),Y(3),Z(3)
      Y1=Y(1)
      Y2=Y(2)
      Y3=Y(3)
      Z(1)=X(1,1)*Y1+X(1,2)*Y2+X(1,3)*Y3
      Z(2)=X(2,1)*Y1+X(2,2)*Y2+X(2,3)*Y3
      Z(3)=X(3,1)*Y1+X(3,2)*Y2+X(3,3)*Y3
      RETURN
      END
      SUBROUTINE NORM(X,Y,Z,ITYPE)
C     ***** VECTOR PRODUCT  Z=X*Y *****
C     ***** ITYPE .GT.0 FOR CARTESIAN,.LE.0 FOR TRICLINIC *****
      DIMENSION X(3),Y(3),Z(3),Z1(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      DO 125 I=1,3
      I1=MOD(I+3,3)+1
      I2=MOD(I+1,3)+1
      T1=X(I1)*Y(I2)-X(I2)*Y(I1)
      IF(ITYPE)115,115,105
  105 Z(I)=T1
      GO TO 125
  115 Z1(I)=T1
  125 CONTINUE
      IF(ITYPE)135,135,300
  135 CALL MV(BB,Z1,Z)
  300 RETURN
      END
      SUBROUTINE NUMBUR(W,W2,HGT,DIST,THT,ND)
C-----CONVERT BOND DISTANCE FOR PLOTTING IN ORTEP
      DIMENSION W(3)
      CHARACTER*8 IFMT,ITXT
      CHARACTER*1 ITEX(8)
      EQUIVALENCE (ITEX(1),ITXT)
C-----COMPUTE NUMBER OF CHARACTERS FOR OUTPUT
      NC=ND+1
```

```
      XD=DIST
   10 IF(XD.LT.1.0) GO TO 20
      IF(NC.GE.9) GO TO 30
      NC=NC+1
      XD=XD/10.0
      GO TO 10
C-----SET UP FORMAT STATEMENT
   20 WRITE (IFMT,25) NC,ND
   25 FORMAT('(F',I1,'.',I1,')')
C-----ENCODE DISTANCE AND PUT IT OUT
      WRITE (ITXT,IFMT) DIST
      CALL SIMBOL(W,W2,HGT,ITEX,THT,NC)
   30 RETURN
      END
      SUBROUTINE PAXES(DCODE,ITYPE)
C     ***** ITYPE .LT.0 FOR COVARIANCE MATRIX IN Q *****
C     ***** ITYPE .GT.0 FOR ELLIPSOID QUADRATIC FORM IN Q *****
C     ***** XABSF(ITYPE)=1 BASED ON TRICLINIC COORDINATE SYSTEM *****
C     ***** =2 OR 3 FOR WORKING OR REFERENCE CARTESIAN SYSTEMS *****
C     ***** CONTRAVARIANT EIGENVECTORS FOR Q IN COLUMNS OF PAC *****
C     ***** CHECK ATOM CODE *****
      DIMENSION X(3)
      REAL*8 DCODE,D100,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100=100.
      D100K=100000.
      IT=IABS(ITYPE)-1
      KS=DMOD(DCODE,D100)
      IF(NSYM-KS)105,115,115
  105 NG=4
      GO TO 300
  115 II=DCODE/D100K
      IF(NATOM-II)125,130,130
  125 NG=5
      GO TO 300
  130 IF(II)125,125,135
C     ***** CRYSTALLOGRAPHIC SYMMETRY ROTATION *****
  135 CALL TMM(PA(1,1,II),FS(1,1,KS),PAT)
      IF(IT-1)160,145,155
C     ***** TRANSFORM TO CARTESIAN SYSTEMS *****
  145 CALL TMM(PAT,AAWRK,PAC)
      GO TO 175
  155 CALL TMM(PAT,AAREV,PAC)
      GO TO 175
  160 IF(ITYPE)162,155,170
C     ***** TRANSFORM TO TRICLINIC SYSTEM *****
  162 DO 165 J=1,9
  165 PAC(J,1)=PAT(J,1)
```

```
      GO TO 175
  170 CALL MM(AA,PAT,PAC)
C     ***** FORM DIAGONAL MATRIX OR ITS INVERSE *****
  175 DO 205 J=1,3
      T1=EV(J,II)
      IF(ITYPE)195,195,185
  185 X(J)=1./(T1*T1)
      GO TO 205
  195 X(J)=T1*T1
  205 RMS(J)=T1
C     ***** FORM QUADRATIC FORM *****
      DO 245 I=1,3
      DO 245 J=I,3
      T1=0.0
      DO 225 K=1,3
  225 T1=T1+PAC(I,K)*PAC(J,K)*X(K)
      Q(J,I)=T1
  245 Q(I,J)=T1
  300 RETURN
      END
      SUBROUTINE PLTXY(X,Y)
C     ***** PLOT COORD. AND CLOSEST EDGE AFTER PROJECTION *****
      DIMENSION X(3),Y(2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      T4=1.
      T1=1.
      IF(VIEW)125,125,110
  110 T4=VIEW-X(3)
      IF(T4)115,115,120
  115 Y(1)=-99.
      Y(2)=-99.
      GO TO 130
  120 T1=VIEW/T4
  125 Y(1)=X(1)*T1+XO(1)
      Y(2)=X(2)*T1+XO(2)
      T1=XLNG(1)-ABS(Y(1)*2.-XLNG(1))
      T2=XLNG(2)-ABS(Y(2)*2.-XLNG(2))
      EDGE=AMIN1(T1,T2)*.5
      IF(T4-VIEW*.5)130,300,300
  130 EDGE=-99.
  300 RETURN
      END
      SUBROUTINE PLOT(x,y,ipen)
      common /ns/ npf,ndraw,norient,nvar
      if (ndraw .eq. 0) return
      if (ndraw .eq. 1) call pensc(x,y,ipen)
      if (ndraw .eq. 2) call penps(x,y,ipen)
      if (ndraw .eq. 3) call penhp(x,y,ipen)
      if (ndraw .eq. 9) call pensc(x,y,ipen)
      return
      end
      SUBROUTINE PRELIM
```

```
C     ***** DATA INPUT ROUTINE *****
      REAL*8 TD
      DIMENSION B(9)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      character*80 card
      character*24 sympart(3)
      dimension fsym(3,4)
      character*36 maksym
C     ***** CELL DIMENSIONS *****
      D100K=100000.
  106 FORMAT(i1,f8.6,5F9.6)
      READ (IN,107)card
  107 format(a)
      write (NED,107) card
      READ (card,106)iflag,(A(I),I=1,6)
      T1=ABS(A(4))-1.
      DO 125 J=1,3
      IF(T1)115,110,110
C     ***** CELL ANGLES IN DEGREES *****
  110 A(J+6)=A(J+3)
      A(J+3)=COS(A(J+6)*1.745329E-2)
      GO TO 120
C     ***** COSINES OF CELL ANGLES *****
  115 A(J+6)=ARCCOS(A(J+3))
C     ***** STORE IDEMFACTOR MATRIX *****
  120 AID(J,J)=1.
      AID(J+1,1)=0.
      AID(J+5,1)=0.
C     ***** STORE METRIC TENSOR *****
  125 AA(J,J)=A(J)**2
      AA(1,2)=A(1)*A(2)*A(6)
      AA(1,3)=A(1)*A(3)*A(5)
      AA(2,3)=A(2)*A(3)*A(4)
      AA(2,1)=AA(1,2)
      AA(3,1)=AA(1,3)
      AA(3,2)=AA(2,3)
C     ***** INVERT METRIC TENSOR *****
      CALL AXEQB(AA,BB,AID,3)
C     ***** CALCULATE RECIPROCAL CELL PARAMETERS *****
      DO 128 J=1,3
  128 B(J)=SQRT(BB(J,J))
      B(6)=BB(1,2)/(B(1)*B(2))
      B(5)=BB(1,3)/(B(1)*B(3))
      B(4)=BB(2,3)/(B(2)*B(3))
      DO 130 J=1,3
  130 B(J+6)=ARCCOS(B(J+3))
```

```
C     ***** WAS INPUT FOR REAL OR RECIPROCAL CELL *****
      IF(A(1)-1.)135,150,150
  135 DO 140 J=1,9
      T1=AA(J,1)
      AA(J,1)=BB(J,1)
      BB(J,1)=T1
      T1=A(J)
      A(J)=B(J)
  140 B(J)=T1
C     ***** WRITE OUT CELL PARAMETERS *****
  143 FORMAT(1H010X,22HDIRECT CELL PARAMETERS/1H 15X,1HA14X,1HB14X,1HC14
     1X,5HALPHA10X,4HBETA11X,5HGAMMA)
  145 FORMAT(1H ,10X,F9.5,2F15.6,3F15.3/1H ,48X,6HCOSINE,F12.8,2F15.8)
  147 FORMAT(1H010X,26HRECIPROCAL CELL PARAMETERS/1H 15X,2HA*13X,2HB*13X
     1,2HC*13X,6HALPHA*9X,5HBETA*10X,6HGAMMA*)
  150 IF (NOUT.GE.0)
     &WRITE (NOUT,143)
      IF (NOUT.GE.0)
     &WRITE (NOUT,145)(A(I),I=1,3),(A(I),I=7,9),(A(I),I=4,6)
      IF (NOUT.GE.0)
     &WRITE (NOUT,147)
      IF (NOUT.GE.0)
     &WRITE (NOUT,145)(B(I),I=1,3),(B(I),I=7,9),(B(I),I=4,6)
C     ***** STORE STANDARD VECTORS *****
      CALL AXES(AID,AID(1,2),REFV,0)
      CALL MM(AA,REFV,AAREV)
      DO 160 I=1,3
      DO 160 J=1,3
      AAWRK(J,I)=AAREV(J,I)
      Q(J,I)=REFV(I,J)
  160 WRKV(J,I)=REFV(J,I)
C     ***** READ AND WRITE SYMMETRY TRANSFORMATIONS *****
  171 FORMAT(1H0,10X,24HSYMMETRY TRANSFORMATIONS/1H ,14X,3HNO.)
c 171 FORMAT(1H010X,24HSYMMETRY TRANSFORMATIONS/1H 14X,3HNO.12X,13HTRANS
c    1FORMED X18X,13HTRANSFORMED Y18X,13HTRANSFORMED Z)
  173 FORMAT(I1,F14.10,3F3.0,2(F15.10,3F3.0))
  175 FORMAT(1H ,13X,I2,3(F13.6,F4.0,2H X,F4.0,2H Y,F4.0,2H Z))
  176 FORMAT(1H ,13X,I2,5x,a)
  177 FORMAT(1H1,10X,18A4)
      IF (NOUT.GE.0)
     &WRITE (NOUT,171)
      LINES=14
      DO 190 I=1,96
      LINES=MOD(LINES+1,56)
      READ (IN,107)card
      write (NED,107) card
      if(iflag.eq.0)READ(card,173)IS,(TS(J,I),(FS(K,J,I),K=1,3),J=1,3)
      if (iflag.eq.1) then
         read (card,1771) is
 1771    format(i1)
         ipart=1
         do 1772 jk=1,3
         do 1772 kl=1,24
 1772    sympart(jk)(kl:kl)=' '
         jk=2
 1773    if (card(jk:jk).eq.' ') go to 1776
         lm=1
         do 1774 kl=jk,80
            if(card(kl:kl).eq.' ' .or. card(kl:kl).eq.',') then
               jk=kl
               go to 1775
               end if
               sympart(ipart)(lm:lm)=card(kl:kl)
               lm=lm+1
 1774       continue
 1775       ipart=ipart+1
 1776       jk=jk+1
            if (jk.lt.80) go to 1773
            do 1777 isymp=1,3
               call tepsym(sympart(isymp),i,isymp)
 1777       continue
         end if
         do 178 j=1,3
            fsym(j,4)=ts(j,i)
            do 178 k=1,3
  178          fsym(j,k)=fs(k,j,i)
         IF(LINES)185,180,185
  180    IF (NOUT.GE.0)
        &WRITE (NOUT,177)(TITLE(J),J=1,18)
         IF (NOUT.GE.0)
        &WRITE (NOUT,171)
c *** ORTEP II symmetry output
c 185 IF (NOUT.GE.0)
c       &WRITE (NOUT,175)I,(TS(J,I),(FS(K,J,I),K=1,3),J=1,3)
  185 if (nout.ge.0) WRITE (NOUT,176)I,maksym(1,fsym)
C     ***** NON-CRYSTALLOGRAPHIC HELIX-SYMMETRY INPUT *****
      IF(FS(3,3,I)-5.)188,186,186
  186 T1=FS(1,3,I)/FS(3,3,I)
      TS(3,I)=TS(3,I)+T1
      T1=AMOD(T1*FS(2,3,I),1.)*6.28318531
      T2=COS(T1)
      T1=SIN(T1)
      DO 187 J=1,9
  187 VT(J,1)=AID(J,1)
      VT(1,1)=T2
      VT(2,2)=T2
      VT(2,1)=-T1
      VT(1,2)=T1
      CALL MM(VT,Q,PAC)
      CALL MM(AAREV,PAC,FS(1,1,I))
  188 IF(IS)195,190,195
  190 CONTINUE
      NG=1
      CALL ERPNT(0.D0,0)
      I=96
  195 NSYM=I
  196 ISW=IS
      NATOM=0
C     ***** POSITIONAL AND THERMAL PARAMETERS *****
  207 FORMAT(11H0 NO. ATOM 8X,1HX10X,1HY10X,1HZ13X,3HB118X,3HB228X,3HB33
     18X,3HB128X,3HB138X,10HB23   TYPE)
  209 FORMAT(1H ,I3,1X,A6,3F11.6,5X,6F11.6,F5.0)
  210 FORMAT(1H ,I3,1X,A6,3F11.6,5X,2F11.6,4F11.0,F5.0)
  211 FORMAT(A6,3X,6F9.0)
  213 FORMAT(I1,F8.0,5F9.0,7X,F2.0)
      LINES=LINES+2
      IF(LINES-56)220,215,215
  215 LINES=-1
      GO TO 225
  220 IF (NOUT.GE.0)
     &WRITE (NOUT,207)
      if (isw.eq.2) then
```

```
          iu=18
          call gtafil(iu)
        end if
C     ***** MAXIMUM NUMBER OF ATOMS EQUALS MAXATM *****
  225 MATOM=NATOM+1
      DO 245 I=MATOM,MAXATM
      LINES=MOD(LINES+1,56)
      IF(ISW.EQ.1) GO TO 226
C     ***** CALL SPECIAL PURPOSE READIN ROUTINE *****
      CALL readin(iu,CHEM(I),IDENT(1,I),IDENT(2,I),P(1,I),P(2,I),P(3,I)
     1,IT1,IS,PA(1,1,I),PA(2,1,I),PA(3,1,I),PA(1,2,I),PA(2,2,I)
     2,PA(3,2,I),PA(1,3,I))
      T1=IT1
      K=IT1+1
      GO TO 229
  226 continue
      READ (IN,107)card
      write (NED,107) card
      READ (card,211)CHEM(I),V1(1),V1(2),(P(J,I),J=1,3),T1
      IDENT(1,I)=V1(1)
      IDENT(2,I)=V1(2)
      K=1.+T1
      IF(FLOAT(K-1)-T1)227,228,227
  227 K=1
  228 continue
      READ (IN,107)card
      write (NED,107) card
      READ (card,213)IS,(PA(J,1,I),J=1,7)
  229 IF(LINES)230,230,232
  230 IF (NOUT.GE.0)
     &WRITE (NOUT,177)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,207)
  232 IF(PA(3,1,I)-10000.)235,234,234
  234 IF (NOUT.GE.0)
     &WRITE (NOUT,210)I,CHEM(I),(P(J,I),J=1,3),(PA(J,1,I),J=1,7)
      GO TO 238
  235 IF (NOUT.GE.0)
     &WRITE (NOUT,209)I,CHEM(I),(P(J,I),J=1,3),(PA(J,1,I),J=1,7)
  238 GO TO (244,239,241,242,244),K
C     ***** TYPE 1 POSITIONAL PARAMETERS (ANGSTROMS) *****
  239 DO 240 J=1,3
  240 P(J,I)=P(J,I)/A(J)
      GO TO 244
C     ***** TYPE 2 POSITIONAL PARAMETERS, STANDARD CARTESIAN *****
  241 V1(1)=P(1,I)
      V1(2)=P(2,I)
      GO TO 243
C     ***** TYPE 3 POSITIONAL PARAMETERS *****
C     ***** CYLINDRICAL COORDINATES REFERRED TO STANDARD CARTESIAN *****
  242 T2=P(2,I)*.01745329252
      V1(1)=V1(1)+P(1,I)*COS(T2)
      V1(2)=V1(2)+P(1,I)*SIN(T2)
  243 V1(3)=P(3,I)
      CALL VM(V1,Q,P(1,I))
  244 IF(IS)246,245,246
  245 CONTINUE
      if (isw.eq.2) close(iu)
      NG=2
      CALL ERPNT(0.D0,0)
      I=MAXATM
```

```
  246 NATOM=I
C     ***** CONVERT TEMP FACTOR COEF TO STANDARD TYPE ZERO *****
      NG1=0
      DO 450 I=1,NATOM
      T1=PA(1,1,I)
c     interim fix for IBM AIX
      K9=7
      K=1.+PA(K9,1,I)
      IF(T1)255,250,255
  250 T1=.1
      GO TO 405
  255 T6=.0506605918
      GO TO(270,260,265,265,270,260,400,405,270,260,270,450),K
C     ***** TYPE 1 *****
  260 DO 262 J=4,6
  262 PA(J,1,I)=PA(J,1,I)*.5
      GO TO 270
C     ***** TYPES 2 AND 3 (BASE 2 SYSTEMS) *****
  265 T6=.351152464
      IF(K-4)270,260,270
C     ***** TYPES 0 THROUGH 5 *****
  270 IF(PA(2,1,I))400,400,272
  272 DO 300 J=1,3
      DO 300 L=J,3
      T2=T6
      IF(K-5)285,275,275
  275 IF(K-6)280,280,281
C     ***** TYPES 4 AND 5 *****
  280 T2=B(J)*B(L)*T2*.25
      GO TO 285
C     ***** TYPES 8 AND 9 (U(I,J) TENSOR SYSTEMS) *****
  281 T2=B(J)*B(L)
      IF(K-11)285,282,282
C     ***** TYPE 10, (CARTESIAN TENSOR SYSTEM) *****
  282 T2=1.0
  285 IF(J-L)290,287,290
  287 VT(J,J)=T2*PA(J,1,I)
      GO TO 300
  290 M=J+L+1
      VT(J,L)=T2*PA(M,1,I)
      VT(L,J)=VT(J,L)
  300 CONTINUE
C     ***** FIND PRINCIPAL AXES *****
      IF(K-11)310,305,305
  305 CALL MM(VT,Q,PAC)
      CALL MM(REFV,PAC,VT)
  310 CALL MM(VT,AA,DA)
      CALL EIGEN(DA,RMS,PAT)
C     ***** ARE EIGENVALUES POSITIVE *****
      IF(RMS(1))325,325,320
  320 IF(NG)350,360,330
  325 NG=3
  330 NG1=1
      CALL ERPNT(DBLE(I)*D100K+55501.D0,0)
C     ***** 3 EQUAL EIGENVALUES, USE REFERENCE VECTORS *****
  340 T3=SIGN(SQRT(ABS(RMS(1)+RMS(2)+RMS(3))/3.),RMS(1))
      DO 345 J=1,3
      DO 342 K=1,3
  342 PA(J,K,I)=REFV(J,K)
  345 EV(J,I)=T3
      GO TO 450
```

```
  350 IF(NG+6)340,340,352
C     ***** TWO EQUAL EIGENVALUES *****
  352 N=NG+5
      CALL UNITY(PAT(1,N),V1,-1)
      DO 354 K=1,3
      IF(ABS(VMV(V1,AA,REFV(1,K)))-.58)356,354,354
  354 CONTINUE
  356 CALL MM(AA,DA,VT)
      CALL AXES(V1,REFV(1,K),DA,-1)
      DO 359 K=1,3
      L=MOD(N+K-2,3)+1
      DO 358 J=1,3
  358 PA(J,L,I)=DA(J,K)
  359 EV(L,I)=SIGN(SQRT(ABS(VMV(DA(1,K),VT,DA(1,K)))),RMS(L))
      GO TO 450
C     ***** MAKE EIGENVECTORS 1 ANGSTROM LONG *****
  360 CALL AXES(PAT(1,1),PAT(1,3),PA(1,1,I),-1)
  370 NG=0
C     ***** SQRT EIGENVALUE = RMS DISPLACEMENT *****
      DO 375 J=1,3
      T2=RMS(J)
  375 EV(J,I)=SIGN(SQRT(ABS(T2)),T2)
      GO TO 450
C     ***** TYPE 6 (ISOTROPIC TEMP FACTOR) *****
  400 T1=SQRT(T1*1.266515E-2)
C     ***** TYPE 7 (DUMMY SPHERE OR ELLIPSOID OF REVOLUTION) *****
  405 IF(PA(2,1,I))409,409,406
C     ***** ELLIPSOID OF REVOLUTION FOR PASS OR PALE *****
  406 EV(1,I)=T1
      EV(2,I)=PA(2,1,I)
      EV(3,I)=PA(2,1,I)
      GO TO 411
C     ***** SPHERE FOR PEAK OR PIT, OR A GENERAL SPHERE ATOM *****
  409 DO 410 J=1,3
  410 EV(J,I)=T1
  411 IF(PA(3,1,I))430,430,415
C     ***** FIRST DEFINED VECTOR FOR SPHERE OR CRITICAL POINT *****
  415 DO 417 J=1,2
      TD=PA(J+2,1,I)
      CALL ATOM(TD,VT(1,J))
      IF(NG)416,417,416
  416 CALL ERPNT(TD,0)
      GO TO 430
  417 CONTINUE
      CALL DIFV(VT(1,2),VT(1,1),V1)
      T11=SQRT(VMV(V1,AA,V1))
      DO 418 J=1,3
  418 V1(J)=V1(J)/T11
C     ***** SECOND DEFINED VECTOR FOR SPHERE OR CRITICAL POINT *****
      DO 420 J=3,4
      TD=PA(J+2,1,I)
      IF(TD.EQ.0.0)GO TO 422
      CALL ATOM(TD,VT(1,J))
      IF(NG)419,420,419
  419 CALL ERPNT(TD,0)
      GO TO 430
  420 CONTINUE
      CALL DIFV(VT(1,4),VT(1,3),V2)
      T11=SQRT(VMV(V2,AA,V2))
      DO 421 J=1,3
  421 V2(J)=V2(J)/T11
C     ***** CHECK FOR NEARLY PARALLEL UNIT VECTORS *****
      IF(ABS(VMV(V1,AA,V2)).LT.0.9) GO TO 429
C     ***** SUBSTITUTE BEST REFERENCE VECTOR *****
  422 T22=1.0
      J22=0
      DO 424 J=1,3
      T11=ABS(VMV(V1,AA,REFV(1,J)))
      IF(T22.LE.T11)GO TO 424
      T22=T11
      J22=J
  424 CONTINUE
      DO 425 J=1,3
  425 V2(J)=REFV(J,J22)
  429 CALL AXES(V1,V2,PA(1,1,I),-1)
      GO TO 450
C     ***** REFERENCE VECTORS FOR SPHERE *****
  430 DO 435 J=1,9
  435 PA(J,1,I)=REFV(J,1)
  450 NG=0
C     ***** WRITE OUT RMS VALUES *****
      LINES=LINES+2
      IF(LINES-56)458,458,455
  455 LINES=-1
      GO TO 460
  458 IF (NOUT.GE.0)
     &WRITE (NOUT,461)
  460 DO 465 I=1,NATOM
      LINES=MOD(LINES+1,56)
      IF(LINES)465,462,465
  461 FORMAT(10H0NO. ATOM 8X,1HX10X,1HY10X,1HZ13X,7HRMSD 1 4X,7HRMSD 2 4
     1X,7HRMSD 3 )
  462 IF (NOUT.GE.0)
     &WRITE (NOUT,177)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,461)
  463 FORMAT(1H ,I3,1X,A6,3F11.6)
  465 IF (NOUT.GE.0)
     &WRITE (NOUT,209)I,CHEM(I),(P(J,I),J=1,3),(EV(J,I),J=1,3
     1)
      IF(NG1)999,999,470
  470 CALL EXITNG(NG)
  999 RETURN
      END
      SUBROUTINE PRIME
C     ****GENERAL INITIALIZATION OF PRIME PARAMETERS****
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      BRDR=0.5
C     ****CALCULATE CONSTANTS****
```

```
      DO 2950 I=1,5
 2950 CONT(I)=SQRT(1./(2.*(1.+COS(3.141593   /2.**I))))
c     DISP=.005
      disp=0.
      FORE=.866
      ITILT=0
      LATM=0
      MAXATM=505
      NCD=0
      NG=0
      DO 3000 J=1,3
 3000 ORGN(J) = 0.0
      RES(1)=.75
      RES(2)=.5*res(1)
      RES(3)=.25*res(2)
      SCAL1=1.0
      SCAL2=1.54
      SCL=1.54
      DO 3005 I=1,3
      SYMB(I,I)=1.
      SYMB(I+1,1)=0.
 3005 SYMB(I+5,1)=0.
      TAPER=.375
      THETA=0.0
      VIEW=0.0
      XLNG(1)=10.5
      XLNG(2)=8.0
      XO(1)=5.25
      XO(2)=4.0
      XO(3)=0.0
C     ***** INITIATE OVERLAP ROUTINES *****
      CALL LAP500(0)
      RETURN
      END
      SUBROUTINE PROJ(D,DP,X,XO,VIEW,I1,I2,I3)
C     ***** 3D CARTESIAN TO 2D PLOTTER COORDINATES *****
      DIMENSION D(3,129),DP(2,129),X(3),XO(3)
      T3=VIEW-X(3)
      DO 145 I=I1,I2,I3
      T1=D(1,I)+X(1)
      T2=D(2,I)+X(2)
      IF(VIEW)135,135,120
  120 T4=VIEW/(T3-D(3,I))
      T1=T1*T4
      T2=T2*T4
  135 DP(1,I)=T1+XO(1)
  145 DP(2,I)=T2+XO(2)
      RETURN
      END
      SUBROUTINE RADIAL(ND)
C     ***** GENERATE ELLIPSE FROM TWO CONJUGATE VECTORS *****
C     ***** ORTHONORMAL VECTORS PRODUCE 8-128 SPOKED CIRCLE *****
C     ***** ND DENOTES NUMBER OF SUBDIVISIONS (1 TO 5) *****
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      DO 115 J=1,3
      T1=DA(J,1)
      D(J,1)=T1
      D(J,129)=T1
      D(J,65)=-T1
      T1=DA(J,2)
      D(J,33)=T1
  115 D(J,97)=-T1
      DO 135 K=1,ND
      T1=CONT(K)
      KDEL=2**(6-K)
      KDEL1=KDEL+1
      KDEL2=KDEL/2
      DO 135 L=KDEL1,65,KDEL
      J=L-KDEL
      M=L-KDEL2
      DO 135 N=1,3
      T2=(D(N,L)+D(N,J))*T1
      D(N,M)=T2
  135 D(N,M+64)=-T2
      RETURN
      END
      subroutine readin(iu,chem,id1,id2,x1,x2,x3,it,is,b1,b2,b3,b4,
     1                 b5,b6,btype)
      integer*2 id1,id2
      character*1 chain
      character*3 res
      character*4 atom
      character*6 rec
      character*8 chem
      b1=.1
      b2=0
      b3=0
      b4=0
      b5=0
      b6=0
      btype=7.
      id1=0
      id2=0
      it=2
c     ***** read the pdb file *****
      read (iu,201) rec,iserno,atom,res,chain,id2,x1,x2,x3,occ,tf
  201 format(a6,i5,1x,a4,1x,a3,1x,a1,i4,4x,3f8.0,2f6.0)
      id1=9
      if (atom.eq.' N  ') id1=1
      if (atom.eq.' CA ') id1=2
      if (atom.eq.' C  ') id1=3
      if (atom.eq.' O  ') then
         id1=4
         b1=.15
      end if
      chem=atom(2:4)//res
      is=0
      read (iu,202,end=203) rec
  202 format(a6)
      backspace(iu)
      return
  203 is=1
      return
```

```
      end
      SUBROUTINE recycle
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      common /ns/ npf,ndraw,norient,nvar

      if (ndraw.eq.2 .or. ndraw.eq.3) call getpap

c *** find 201(203) instruction
      do 545 i=1,nque
         if (que(i)(7:9).eq.'201' .or. que(i)(7:9).eq.'203') then
            next=i
            go to 570
         end if
  545 continue
C *** ZERO ATOMS ARRAY AND RETURN TO EXECUTE NEXT INSTRUCTION ***
  570 LATM=0
      DO 580 I=1,500
      ATOMID(I)=0.
      DO 580 J=1,3
  580 ATOMS(J,I)=0.
      RETURN
      END
      SUBROUTINE SCRIBE(Y,NPEN)
      DIMENSION Y(2),YO(2)
C     ***** SUBROUTINE WHICH LINKS WITH THE PLOTTER-SPECIFIC SUBROUTINES
      IF(NPEN-3)210,205,205
C     ***** KEEP TRACK OF COORDINATES FOR LAST PEN-UP LOCATION *****
  205 YO(1)=Y(1)
      YO(2)=Y(2)
      NPO=0
      RETURN
C     ***** CALL MECHANICAL PLOTTER PLOTTING SUBROUTINE *****
  210 IF(NPO)225,220,225
  220 CONTINUE
      CALL PLOT(YO(1),YO(2),3)
  225 CONTINUE
      CALL PLOT(Y(1),Y(2),2)
      NPO=1
      RETURN
      END
      SUBROUTINE SEARC
      DIMENSION NW(6),DX(3),S1D(200),S2(200),U(3),V(3),W(2,4),WW(2,3)
      DIMENSION X(4),Y(3),Z(3)
      REAL*8 DZMIN,DZMAX,DZSTO,S1D,TD1,TD2,TD3,TD4,D10K,D100K
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      logical featur
      featur=.false.
      ifeat=ain(6)
      if ((ifeat.eq.1 .or. ifeat.eq.2) .and.
     &    (mod(nj2,10).eq.5 .or. mod(nj2,10).eq.6)) featur=.true.
C     ***** OBTAIN PROBLEM PARAMETERS *****
      D10K=10000.
      D100K=100000.
      IF (NOUT.GE.0)
     &WRITE (NOUT,20)
   20 FORMAT(1H0 9X,82H        FROM ATOMS    TO ATOMS     WITH RADIUS OR
     1, IF A BOX, WITH SEMIDIMENSIONS  /11X,46HCODE   (MIN  MAX)   (MIN
     2 MAX)                7X,1HA8X,1HB8X,1HC)
      IF(AIN(1)-D10K)100,100,101
  100 ITOM1=AIN(1)
      SYITOM=55501.
      GO TO 103
  101 ITOM1=AIN(1)/D100K
      SYITOM=DMOD(AIN(1),D100K)
  102 IF(DABS(AIN(2))-D10K)103,103,104
  103 ITOM2 = DABS(AIN(2))
      SYITO2=SYITOM
      GO TO 105
  104 ITOM2=DABS(AIN(2))/D100K
      SYITO2=DMOD(DABS(AIN(2)),D100K)
  105 ITAR1=AIN(3)
      IF(ITAR1)108,108,110
  108 ITAR1=1
  110 ITAR2=AIN(4)
      DMAX=AIN(5)
      IF(DMAX)115,115,120
  115 DMAX=4.
      AIN(5)=DMAX
  120 DMX=DMAX*DMAX
      TEM=.01
      KFUN=NJ*100+MOD(NJ2,10)
      K=NJ*100+NJ2
      I0=SYITOM
      I02=SYITO2
      LATOM=LATM
  121 FORMAT(1H0,10X,2I3,I5,I4,I5,2I4,18X,3F9.3/1H )
      IF (NOUT.GE.0)
     &WRITE (NOUT,121)K,ITOM1,I0,ITOM2,I02,
     1ITAR1,ITAR2,(AIN(J),J=5,7)
  124 FORMAT(1H ,15X,2I5,I8,I5,2F9.3)
      IF(NCD)130,130,125
  125 IF (NOUT.GE.0)
     &WRITE (NOUT,124) ((KD(J,I),J=1,4),(CD(J,I),J=1,2) ,I=1,
     1NCD)
  130 DO 135 J=1,4
      W(1,J)=99.
  135 W(2,J)=-99.
```

```
      inum=0
      DO 153 KI=ITAR1,ITAR2
      i=ki
  136 if (featur) then
         inum=inum+1
         i=inum
         if (ident(ifeat,i).ne.ki) go to 154
      end if
      TD=FLOAT(I)*D100K
      CALL ATOM(TD,X)
      IF(NG)140,145,140
  140 CALL ERPNT(TD,KFUN)
      GO TO 600
  145 X(4)=X(1)-X(2)
      DO 155 J=1,4
      TEM=X(J)
      IF(W(2,J)-TEM)148,150,150
  148 W(2,J)=TEM
  150 IF(TEM-W(1,J))152,155,155
  152 W(1,J)=TEM
  155 CONTINUE
  154 if (featur .and. inum.lt.natom) go to 136
  153 CONTINUE
      KFUN2=MOD(KFUN,10)
      GO TO (165,165,160,156,165,165),KFUN2
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES TRICLINIC BOX *****
  156 DO 158 J=1,3
  158 DX(J)=AIN(J+4)
      GO TO 170
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES RECTANGULAR BOX *****
  160 DO 162 J=1,3
      DX(J)=0.
      DO 162 I=1,3
      T9=AIN(I+4)
  162 DX(J)=DX(J)+ABS(REFV(J,I)*T9)
      GO TO 170
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES DMAX SPHERE *****
  165 T1=1.-A(4)*A(4)-A(5)*A(5)-A(6)*A(6)+2.*A(4)*A(5)*A(6)
      DO 168 J=1,3
  168 DX(J)=SQRT((1.-A(J+3)**2)/T1)*DMAX/A(J)
C     ***** START SEARCH AROUND REFERENCE ATOMS *****
  170 LIST=0
      LAST=0
      M1=ITOM1
      N1=ITOM2
      IF(KFUN2-5)186,172,172
C     ***** CONVOLUTE AND REITERATIVE CONVOLUTE INSTRUCTIONS *****
  172 IF(LATM)174,174,176
C     ***** FAULT, NO ENTRIES IN ATOMS LIST *****
  174 NG=12
      CALL ERPNT(0.D0,KFUN)
      GO TO 600
C     ***** CHECK FOR REFERENCE ATOMS IN ATOMS LIST *****
  176 IF(LATM-LAST)600,600,177
  177 LIST=LAST
      LAST=LATM
  178 LIST=LIST+1
      IF(LAST-LIST)505,180,180
  180 TD1=ATOMID(LIST)
      IF(LAST-LIST.LE.0 .OR. AIN(8).EQ.0.D0) GO TO 184
C     ***** FIND SMALLEST ATOM NUMBER IN REMAINDER OF ATOMS LIST *****
      LISTP1=LIST+1
      DO 182 J=LISTP1,LAST
      IF(TD1.LE.ATOMID(J)) GO TO 182
      DO 181 I=1,3
      T1=ATOMS(I,J)
      ATOMS(I,J)=ATOMS(I,LIST)
  181 ATOMS(I,LIST)=T1
      TD1=ATOMID(J)
      ATOMID(J)=ATOMID(LIST)
      ATOMID(LIST)=TD1
  182 CONTINUE
  184 ITOM=TD1/D100K
      if (featur) then
         if (ident(ifeat,itom).lt.itom1 .or.
     &       ident(ifeat,itom).gt.itom2) go to 178
      else
         IF(ITOM.LT.ITOM1 .OR. ITOM.GT.ITOM2) GO TO 178
      end if
      SYITOM=DMOD(TD1,D100K)
      SYITO2=SYITOM
      M1=ITOM
      N1=ITOM
C     ***** SET INITIAL RUN PARAMETERS *****
  186 M2=AMOD(SYITOM,100.)
      M5=AMOD(SYITOM/100.,1000.)
      M3=M5/100
      M4=MOD(M5/10,10)
      M5=MOD(M5,10)
C     ***** SET TERMINAL RUN PARAMETERS *****
      N2=AMOD(SYITO2,100.)
      N5=AMOD(SYITO2/100.,1000.)
      N3=N5/100
      N4=MOD(N5/10,10)
      N5=MOD(N5,10)
C     ***** START SEARCH AROUND REFERENCE ATOMS *****
      DO 500 L5=M5,N5
      DO 500 L4=M4,N4
      DO 500 L3=M3,N3
      DO 500 L2=M2,N2
      DO 500 ITOM=M1,N1
      TD3=DBLE(ITOM)*D100K+DBLE(L3*10000+L4*1000+L5*100+L2)
      CALL ATOM(TD3,Y)
      IF(NG)188,190,188
  188 CALL ERPNT(TD3,KFUN)
      GO TO 500
C     ***** K=SYMMETRY EQUIVALENT POSITION *****
  190 NUM=0
      DO 400 K=1,NSYM
C     ***** SUBTRACT SYMMETRY TRANSLATION FROM REFERENCE ATOM *****
      DO 192 J=1,3
  192 U(J)=Y(J)-TS(J,K)
C     ***** DETERMINE LIMITING CELLS TO BE SEARCHED *****
C     ***** FIRST,MOVE THE BOX THROUGH THE SYMMETRY OPERATION *****
      DO 200 J=1,3
      DO 200 L=1,2
      WW(L,J)=0.0
      DO 200 I=1,3
      TEM=FS(I,J,K)
      IF(TEM)194,200,196
  194 N=MOD(L,2)+1
      GO TO 198
```

```
  196 N=L
  198 WW(L,J)=WW(L,J)+W(N,I)*TEM
  200 CONTINUE
C     ***** CHECK FOR MIXED INDEX TRANSFORMATION *****
      DO 215 J=1,2
      TEM=FS(1,J,K)
      IF(TEM+FS(2,J,K))215,201,215
  201 IF(TEM)203,215,207
  203 WW(1,J)=W(2,4)*TEM
      WW(2,J)=W(1,4)*TEM
      GO TO 215
  207 WW(1,J)=W(1,4)*TEM
      WW(2,J)=W(2,4)*TEM
  215 CONTINUE
C     ***** MOVE 4 CELLS AWAY THEN MOVE BACK UNTIL PARALLELEPIPED AROUND
C         REF ATOM AND BOX AROUND TRANSFORMED ASYM UNIT INTERSECT *****
      N=0
      DO 235 J=1,3
      DO 225 I=1,2
      N=N+1
      TT=(U(J)-WW(I,J))*FLOAT(I*2-3)-DX(J)
      TEM=5.0
  221 TEM=TEM-1.0
      IF(TEM+TT)225,225,221
  225 NW(N)=TEM*FLOAT(I*2-3)+5.
C     ***** IF NO POSSIBILITY OF A HIT, GO TO NEXT SYMMETRY OPER *****
      IF(NW(N)-NW(N-1))400,235,235
  235 CONTINUE
      LL=NW(1)
      LU=NW(2)
      ML=NW(3)
      MU=NW(4)
      NL=NW(5)
      NU=NW(6)
C     ***** L CELL TRANSLATIONS IN X *****
      DO 396 L=LL,LU
      V(1)=U(1)+FLOAT(L-5)
C     ***** M CELL TRANSLATIONS IN Y *****
      DO 396 M=ML,MU
      V(2)=U(2)+FLOAT(M-5)
C     ***** N CELL TRANSLATIONS IN Z *****
      DO 396 NN=NL,NU
      V(3)=U(3)+FLOAT(NN-5)
C     ***** I = TARGET ATOM *****
      inum=0
      DO 396 KI=ITAR1,ITAR2
      i=ki
  244 if (featur) then
         inum=inum+1
         i=inum
         if (ident(ifeat,i).ne.ki) go to 395
      end if
      DO 250 J=1,3
      TEM=0.0
      DO 245 II=1,3
  245 TEM=TEM+FS(II,J,K)*P(II,I)
C     ***** SEE IF WITHIN PARALLELEPIPED*****
      TEM=TEM-V(J)
      IF(DX(J)-ABS(TEM))395,250,250
  250 X(J)=TEM
      GO TO (255,255,252,265,255,255),KFUN2
C     ***** SEE IF WITHIN MODEL BOX *****
  252 CALL VM(X,AAREV,V1(2))
      DO 253 J=2,4
      IF(AIN(J+3)-ABS(V1(J)))395,253,253
  253 CONTINUE
      GO TO 265
C     ***** SEE IF WITHIN SPHERE *****
  255 DSQ=VMV(X,AA,X)
      IF(DMX-DSQ)395,256,256
  256 IF(DSQ-.0001)258,260,260
  258 IF(KFUN-402)395,260,260
  260 TEM=SQRT(DSQ)
      IF(AIN(8))265,265,261
C     *****SELECT ONLY FIRST ASYMMETRIC UNIT ENCOUNTERED *****
  261 IF(LATM)265,265,262
  262 DZMIN=DBLE(I)*D100K
      DZMAX=DZMIN+D100K
      DO 264 J=1,LATM
      DZSTO=ATOMID(J)
      IF(DZSTO-DZMIN)264,263,263
  263 IF(DZMAX-DZSTO)264,264,395
  264 CONTINUE


C     ***** SELECT VECTORS ACCORDING TO CODES IF ANY *****
  265 if(ncd.le.0) go to 277
c     if logc=0, screening conditions are ORed
c     if logc=1, screening conditions are ANDed
      logc=ain(8)
  268 DO 275 J=1,NCD
      norg=itom
      ntar=i
      if (kd(5,j).eq.1) then
         norg=ident(1,itom)
         ntar=ident(1,i)
      end if
      if (kd(5,j).eq.2) then
         norg=ident(2,itom)
         ntar=ident(2,i)
      end if
      if (logc.eq.0) then
         if (kd(2,j).gt.0) then
            if (norg.lt.kd(1,j) .or. norg.gt.kd(2,j)) go to 275
         end if
         if (kd(4,j).gt.0) then
            if (ntar.lt.kd(3,j) .or. ntar.gt.kd(4,j)) go to 275
         end if
         if (cd(2,j).gt.0.) then
            if (tem.lt.cd(1,j)  .or. tem.gt.cd(2,j)) go to 275
         end if
         go to 277
      end if
      if (logc.eq.1) then
         if (kd(2,j).gt.0) then
            if (norg.lt.kd(1,j) .or. norg.gt.kd(2,j)) go to 276
         end if
         if (kd(4,j).gt.0) then
            if (ntar.lt.kd(3,j) .or. ntar.gt.kd(4,j)) go to 276
         end if
         if (cd(2,j).gt.0.) then
            if (tem.lt.cd(1,j)  .or. tem.gt.cd(2,j)) go to 276
```

```
          end if
          if (j.eq.ncd) go to 277
       end if
  275 CONTINUE


  276 GO TO 395
  277 TD=D100K*DBLE(I)+DBLE((1110-L*100-M*10-NN)*100+K)
      IF(KFUN-402)278,325,325
C     ***** DETERMINE CORRECT POSITION IN SORTED VECTOR TABLE *****
  278 IF(NUM)317,317,279
  279 DO 315 II=1,NUM
      TT=S2(II)-TEM
      IF(ABS(TT)-0.0001)297,297,281
  281 IF(TT)315,297,283
C     ***** MOVE LONGER VECTORS TOWARD END OF TABLE *****
  283 IF(200-NUM)287,287,289
  287 NUM=199
  289 IJ=NUM
      DO 295 J=II,NUM
      S1D(IJ+1)=S1D(IJ)
      S2(IJ+1)=S2(IJ)
  295 IJ=IJ-1
      GO TO 319
C     ***** CHECK FOR DUPLICATE VECTORS IF DISTANCES ARE EQUAL *****
  297 CALL ATOM(S1D(II),Z)
      DO 305 J=1,3
      IF(ABS(X(J)+Y(J)-Z(J))-0.0001)305,305,315
  305 CONTINUE
      GO TO 395
  315 CONTINUE
      IF(200-NUM)320,320,317
C     ***** STORE THE RESULT IN VECTOR TABLE *****
  317 II=NUM+1
  319 NUM=NUM+1
      S1D(II)=TD
      S2(II)=TEM
  320 IF(KFUN-106)395,325,325
C     ***** STORE RESULT IN ATOMS TABLE *****
  325 DO 330 J=1,3
  330 V1(J)=X(J)+Y(J)
      CALL STOR(TD)
  395 if (featur .and. inum.lt.natom) go to 244
  396 CONTINUE
  400 CONTINUE
C     ***** PRINT OUT DISTANCES *****
  421 FORMAT(1H010X,20HVECTORS FROM ATOM  (I3,1H,I5,1H)6X,8HTO ATOMSI4,
     18H THROUGH,I4)
      I0 = DMOD(TD3,D100K)
      IF (NOUT.GE.0)
     &WRITE (NOUT,421)ITOM,I0,ITAR1,ITAR2
      IF(NUM)500,500,423
  423 DO 435 I=1,NUM
      TD2=S1D(I)
      I1=TD2/D100K
      I2=TD2-DBLE(I1)*D100K
      CALL ATOM(TD2,Z)
      IF(I-1)432,432,434
  427 FORMAT(1H 13X,2(A6,1X),39X,1H(I3,1H,I5,1H)3F7.4,7X,3HD =F6.3)
  429 FORMAT(1H 13X,2(A6,1X),2(3H  (I3,1H,I5,1H)3F7.4,3X),4X,3HD =F6.3)
  432 IF (NOUT.GE.0)
     &WRITE (NOUT,429)CHEM(ITOM),CHEM(I1),ITOM,I0,(Y(J),J=1,3
     1),I1,I2,(Z(J),J=1,3),S2(I)
      GO TO 435
  434 IF (NOUT.GE.0)
     &WRITE (NOUT,427)CHEM(ITOM),CHEM(I1),I1,I2,(Z(J),J=1,3),
     1S2(I)
  435 CONTINUE
C     ***** CALCULATE ANGLES ABOUT REF ATOM IF CODE IS 102 *****
  437 IF(KFUN-102)500,451,500
  441 FORMAT(1H0,10X,18HANGLES AROUND ATOM,I5)
  451 IF (NOUT.GE.0)
     &WRITE (NOUT,441)ITOM
      L=NUM-1
      IF(L)500,500,457
  457 DO 465 I=1,L
      TD2=S1D(I)
      T3=S2(I)
      I1=TD2/D100K
      I2=TD2-DBLE(I1)*D100K
      CALL ATOM(TD2,X)
      CALL DIFV(X,Y,U)
      CALL MV(AA,U,V2)
      M=I+1
      DO 465 J=M,NUM
      TD4=S1D(J)
      J1=TD4/D100K
      J2=TD4-DBLE(J1)*D100K
      CALL ATOM(TD4,Z)
      CALL DIFV(Z,Y,V)
      F=ARCCOS(VV(V,V2)/(T3*S2(J)))
      CALL DIFV(X,Z,V3)
      F1=SQRT(VMV(V3,AA,V3))
  460 FORMAT(1H 13X,3(A6,1X),7X,3(2H (I3,1H,I5,1H)),12X,3HD =F6.3,7X,3HA
     1 =F6.2)
      IF (NOUT.GE.0)
     &WRITE (NOUT,460)CHEM(I1),CHEM(ITOM),CHEM(J1),I1,I2,ITOM
     1,I0,J1,J2,F1,F
  465 CONTINUE
  495 CONTINUE
  500 CONTINUE
      IF(LAST-LIST)505,505,178
  505 IF(KFUN2-6)600,176,600
  600 IF(KFUN-106)610,605,610
  605 LATM=LATOM
  610 RETURN
      END
      SUBROUTINE SIMBOL(W,W2,HGT,ITXT,THT,N)
      DIMENSION W(3),ITXT(72),DS(10),DC(10)
c     DIMENSION IPTR(64),NKNT(64),IXYT(349)
      DIMENSION IPTR(90),NKNT(90),IXYT(556)
      CHARACTER*1 ITXT
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans
      DATA IPTR
     1/312, 32, 41, 54, 47, 20, 20, 54, 64, 70,103,108, 17,115,118,128
     2,140,128,140,194,166,102,208,120, 27,211,216,  0,  0,  0,  0,237
     3,  0,150,328,301,168,223, 92,328, 88,188,180,  1, 82,180, 82, 30
     4,128,239,244,252,269,277,258,284,192,289, 76, 76, 11,  6, 14,153
     5,350,360,370,378,388,398,407,421,428,436,444,450,455,465,472,484
     6,494,504,510,521,526,533,536,541,546,550/
      DATA NKNT
```

```
     1/ 16,  9, 12,  8,  7,  7,  6, 10,  6,  6,  5,  7,  3,  5,  4,  9
     2,  7, 12, 10, 12,  4,  6,  3,  5,  5,  5,  7,  0,  0,  0,  0,  2
     3,  0,  7,  9, 11, 12, 14, 10,  4,  4,  4,  8,  5,  6,  2,  5,  2
     4,  9,  5,  8, 13,  8,  9, 11,  5, 16, 12, 11, 12,  3,  5,  3, 13
     5, 10, 10,  8, 10, 10,  9, 14,  7,  8,  8,  6,  5, 10,  7,  9, 10
     6, 10,  6, 11,  5,  7,  3,  5,  5,  4,  7/
C        @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
C        P   Q   R   S   T   U   V   W   X   Y   Z                   _
C            !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
C        0   1   2   3   4   5   6   7   8   9   :   ;   <   =   >   ?
C        a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p
C        q   r   s   t   u   v   w   x   y   z
      DATA IXYT
     1/44,48,46,26,66,24,64,99,66,26,68,26,64,24,66,28,29,22,62,69
     2,29,26,56,26,22,62,29,62,99,22,69,22,25,65,25,28,39,59,68,62
     3,63,65,56,26,56,67,68,59,29,22,52,63,68,68,59,39,28,23,32,52
     4,63,65,55,22,29,26,66,69,62,32,52,42,49,39,59,35,36,46,45,35
     5,99,42,32,33,43,42,31,69,58,53,62,62,37,38,49,58,25,24,33,43
     6,64,29,23,32,52,63,69,29,22,25,69,99,47,62,22,29,45,69,62,29
     7,22,46,62,69,47,69,99,68,59,39,28,23,32,52,63,68,99,44,62,22
     8,29,59,68,67,56,26,56,65,62,49,44,99,32,43,52,32,99,44,46,56
     9,67,68,59,39,28,29,69,49,42,99,23,53,64,65,56,36,27,38,68,25
     A,65,45,63,27,45,23,67,29,38,33,22,56,67,68,59,39,28,27,36,56
     B,65,63,52,32,23,25,36,29,42,69,29,47,42,47,69,29,69,22,62,99
     C,36,56,38,28,29,39,38,99,69,22,99,53,63,62,52,53,15,75,38,49
     D,42,32,52,28,39,59,68,66,24,22,62,28,39,59,68,67,56,36,56,65
     E,63,52,32,23,28,39,59,68,29,24,64,54,59,52,42,62,23,32,52,63
     F,65,56,26,29,69,68,43,42,23,32,52,63,68,59,39,28,26,35,55,66
     G,24,64,54,53,57,56,66,26,36,37,33,66,57,47,36,35,44,54,65,67
     H,58,38,27,24,33,53,64,57,49,59,57,99,37,29,39,37,22,32,12,22
     I,23,21,22,31,13,22,33,11,22
c 349
     a,62,67,66,57,37,26,23,32,52,63
     b,22,29,26,37,57,66,63,52,32,23
     c,63,52,32,23,26,37,57,66
     d,62,69,66,57,37,26,23,32,52,63
     e,63,52,32,23,26,37,57,66,65,35
     f,32,36,26,46,36,38,49,59,68
     g,64,67,66,57,37,26,24,33,53,64,62,51,31,22
c 420
     h,22,29,26,37,57,66,62
     i,32,52,42,47,37,99,48,49
     j,32,41,51,62,67,99,68,69
     k,22,29,24,57,35,62
     l,32,52,42,49,39
     m,22,27,26,37,46,42,46,57,66,62
c 458
     n,22,27,26,37,57,66,62
     o,63,52,32,23,26,37,57,66,63,99,67,45
     p,21,27,26,37,57,66,64,53,33,24
     q,61,67,66,57,37,26,24,33,53,64
     r,22,27,26,37,57,66
c 500 82
     s,23,32,52,63,64,55,35,26,37,57,66
c    t,42,48,46,36,56
     t,42,49,47,27,67
     u,67,62,63,52,32,23,27
     v,27,42,67
c    w,27,22,45,62,67
     w,27,32,46,52,67
     x,22,67,99,27,62
     y,27,43,67,31
c    y,27,44,67,21,21
c    y,27,42,31,42,67
     z,62,22,67,27,99,35,55/
      DATA RAD/0.01745329/
      if (ndraw.eq.9) then
         write (npf,21) n,nvar,w(1)+xtrans,w(2)+ytrans,8.*hgt,tht
   21    format('TXT',i2,1x,i2,' 1',4(1x,f10.6))
         write (npf,22) (itxt(k),k=1,n)
   22    format(80a1)
         return
      end if
C-----TEST FOR SPECIAL CASE OF CENTERED SYMBOL
      IF(N.LE.0) GO TO 400
C-----SET UP TABLE OF INCREMENTS BASED ON HGT AND THT
      IF(THT.EQ.0.0) GO TO 120
      TH=RAD*THT
      ST=SIN(TH)
      CT=COS(TH)
      GO TO 130
  120 ST=0.0
      CT=1.0
  130 D=HGT/7.0
      DST=D*ST
      DCT=D*CT
      DS(1)=-DST
      DC(1)=-DCT
      DO 145 I=2,10
      DS(I)=DS(I-1)+DST
      DC(I)=DC(I-1)+DCT
  145 CONTINUE
C-----START LOOP THROUGH THE N CHARACTERS OF ITXT
      XO=0.0
      YO=0.0
      DO 370 J=1,N
      ITXTJ=ICHAR(ITXT(J))
      if (itxtj.ge.97.and.itxtj.le.122) then
          ich=itxtj-32
          go to 221
      end if
C-----MASK IT TO SIX BITS AND ADD ONE. PICK UP POINTER AND COUNTER
  220 ICH=MOD(ITXTJ,64)+1
  221 IP=IPTR(ICH)
      NK=NKNT(ICH)
C-----TEST FOR SPACE OR UNDEFINED CHARACTER
      IF(NK.EQ.0) GO TO 360
C-----START LOOP THROUGH SEGMENTS OF CHARACTER. LIFT PEN INITIALLY
      IPEN=3
      DO 350 K=1,NK
      IXY=IXYT(IP)
C-----LIFT PEN IF SPECIAL INDICATOR IS FOUND
      IF(IXY.NE.99) GO TO 300
      IPEN=3
      GO TO 340
  300 IX=IXY/10
      IY=IXY-10*IX
      DX=XO+DC(IX)-DS(IY)
      DY=YO+DC(IY)+DS(IX)
      CALL DRAW(W,DX,DY,IPEN)
C-----PUT PEN DOWN TO DRAW NEXT SEGMENTS
      IPEN=2
```

```
  340 IP=IP+1
  350 CONTINUE
C-----MOVE ORIGIN TO NEXT CHARACTER POSITION
  360 XO=XO+DC(8)
      YO=YO+DS(8)
  370 CONTINUE
      RETURN
c *** Only one centered symbol (*) is available in ORTEP-III.
C-----PLOT ONE SPECIFIC CENTERED SYMBOL. SET UP TABLE OF INCREMENTS
  400 DCT=HGT/2.0
      DC(1)=-DCT
      DC(2)= 0.0
      DC(3)= DCT
C-----MOVE TO SYMBOL WITH PEN UP OR DOWN, DEPENDING ON N
      IPEN=3
      IF(N.LE.-2) IPEN=2
C-----LOOP THROUGH SEGMENTS OF CENTERED SYMBOL
      DO 440 K=337,349
      IXY=IXYT(K)
      IX=IXY/10
      IY=IXY-10*IX
      CALL DRAW(W,DC(IX),DC(IY),IPEN)
C-----PUT PEN DOWN TO DRAW REMAINING SEGMENTS
      IPEN=2
  440 CONTINUE
      RETURN
      END
      SUBROUTINE SPARE(INST)
C     ***** THIS SUBROUTINE MAY BE USED FOR NEW INSTRUCTIONS *****
      RETURN
      END
      SUBROUTINE STOR(TD1)
C     ***** STORE IN OR REMOVE FROM ATOMS ARRAY *****
      REAL*8 TD1
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IF(LATM)481,481,450
  450 IF(500-LATM)455,455,460
  455 IF(NJ2-10)490,490,460
  460 L=LATM
C     ***** CHECK FOR POSITIONAL DUPLICATION *****
      DO 480 K=1,L
      DO 465 J=1,3
      IF(ABS(V1(J)-ATOMS(J,K))-0.001)465,465,480
  465 CONTINUE
      IF(NJ2-10)490,490,470
C     ***** ATOM REMOVAL BY TABLE PUSHDOWN *****
  470 LATM=LATM-1
      DO 475 I=K,LATM
      ATOMID(I)=ATOMID(I+1)
      DO 475 J=1,3
  475 ATOMS(J,I)=ATOMS(J,I+1)
      GO TO 490
```

```
  480 CONTINUE
  481 IF(NJ2-10)482,490,490
C     ***** STORE ATOM *****
  482 IF(499-LATM)490,483,485
  483 NG=16
      CALL ERPNT (TD1,400)
  485 LATM=LATM+1
      DO 486 J=1,3
  486 ATOMS(J,LATM)=V1(J)
      ATOMID(LATM)=TD1
  490 RETURN
      END
      subroutine tepsym(txt,num,kk)
c *** parses character string representation of symmetry operators
c *** and stores information
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      character*24 txt
      logical twodig

c *** convert txt to upper case
      do 101 i=1,24
         iascii=ichar(txt(i:i))
         if (iascii.ge.97.and.iascii.le.122) txt(i:i)=char(iascii-32)
  101 continue

c *** default value of ts if no fraction specified
  202 ts(kk,num)=0.

c *** look for and interpret a/b style fraction
      n=index(txt,'/')
      if (n.gt.0) then

c        get denominator
         k=ichar(txt(n+1:n+1))-48
         m=ichar(txt(n+2:n+2))-48
         if (m.ge.0 .and. m.le.9) then
            iden=k * 10 + m
         else
            iden=k
         end if

c        get numerator
         twodig=.false.
         ksign=1
         k=ichar(txt(n-1:n-1))-48
         if (n-2.ge.1) then
            m=ichar(txt(n-2:n-2))-48
            if (m.ge.0 .and. m.le.9) twodig=.true.
            if (txt(n-2:n-2).eq.'-') ksign=-1
            if (n-3.ge.1) then
               if (txt(n-3:n-3).eq.'-') ksign=-1
            end if
```

```
         end if
         if (twodig) then
            inum=ksign * (m * 10 + k)
         else
            inum=ksign * k
         end if

         ts(kk,num)=float(inum)/float(iden)
      end if

c *** look for and interpret decimal style fraction
      n=index(txt,'.')
      if (n.gt.0) then

c        get post decimal point portion
         k=ichar(txt(n+1:n+1))-48
         m=ichar(txt(n+2:n+2))-48
         if (m.ge.0 .and. m.le.9) then
            ts(kk,num)=float(k) * .1 + float(m) * .01
         else
            ts(kk,num)=float(k) * .1
         end if

c        get sign
         ksign=1
         if (n-1.ge.1) then
            if (txt(n-1:n-1).eq.'-') ksign=-1
         end if
         if (n-2.ge.1) then
            if (txt(n-2:n-2).eq.'-') ksign=-1
         end if

         ts(kk,num)=float(ksign) * ts(kk,num)
      end if

c *** interpret xyz portion of symmetry operation
      do 303 i=1,24
         if (txt(i:i).eq.'X') then
            fs(1,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(1,kk,num)=-1.
            end if
         end if
         if (txt(i:i).eq.'Y') then
            fs(2,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(2,kk,num)=-1.
            end if
         end if
         if (txt(i:i).eq.'Z') then
            fs(3,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(3,kk,num)=-1.
            end if
         end if
303   continue

      return
      end
      SUBROUTINE TMM(X,Y,Z)
C     Z = TRANSPOSED (TRANSPOSE(X) * (Y) )
C     Z(3,3)=X(3,3)*Y(3,3)
      DIMENSION X(3,3),Y(3,3),Z(3,3)
      X11=X(1,1)
      X12=X(1,2)
      X13=X(1,3)
      X21=X(2,1)
      X22=X(2,2)
      X23=X(2,3)
      X31=X(3,1)
      X32=X(3,2)
      X33=X(3,3)
      Y11=Y(1,1)
      Y12=Y(1,2)
      Y13=Y(1,3)
      Y21=Y(2,1)
      Y22=Y(2,2)
      Y23=Y(2,3)
      Y31=Y(3,1)
      Y32=Y(3,2)
      Y33=Y(3,3)
      Z(1,1)=X11*Y11+X21*Y21+X31*Y31
      Z(1,2)=X12*Y11+X22*Y21+X32*Y31
      Z(1,3)=X13*Y11+X23*Y21+X33*Y31
      Z(2,1)=X11*Y12+X21*Y22+X31*Y32
      Z(2,2)=X12*Y12+X22*Y22+X32*Y32
      Z(2,3)=X13*Y12+X23*Y22+X33*Y32
      Z(3,1)=X11*Y13+X21*Y23+X31*Y33
      Z(3,2)=X12*Y13+X22*Y23+X32*Y33
      Z(3,3)=X13*Y13+X23*Y23+X33*Y33
      RETURN
      END
      subroutine uinput(in,nout)
c *** user input routine
      common /ns/ npf,ndraw,norient,nvar
      common /dfl/ infile,idraw,iorient,iout,ext,atomfi,fpaplen
      character*60 fname,txtans,infile,atomfi
      character*4 ext
      character*1 answer

      call dflts
      iflag=0
c *** get the input file name and open the file or "exit" ***
      if (iargc().eq.1) then
         call getarg(1,fname)
         open(in,file=fname,status='old',err=110)
         go to 135
      end if
  110 fname=infile
      ipos=index(fname,' ')
      write (*,115) fname(1:ipos-1)
  115 format(' Enter instruction set file name or "exit" [',a,']:  ',$)
      read (*,120) txtans
  120 format(a)
      if (txtans(1:4).eq.'exit' .or. txtans(1:4).eq.'EXIT')
     *    call exitng(0)
      if (txtans(1:1) .ne. ' ') fname=txtans
      open(in,file=fname,status='old',err=125)
      go to 135
  125 ipos=index(fname,' ')
      fname=fname(1:ipos-1)
      write (*,130) fname(1:ipos-1)
```

```
  130 format(/' "',a,'" does not exist'/)
      go to 110
c *** determine where ortep drawing should go ***
  135 write (*,140) idraw
  140 format(' Drawing to (1) Screen, (2) Postscript file, (3) HPGL file
     *, or (0) Omit [',i1,']:  ',$)
      read (*,145) answer
  145 format(a)
      if (answer .eq. ' ') then
         ndraw=idraw
      else
         read (answer,155) ndraw
      end if
      if (ndraw.lt.0.or.(ndraw.gt.3.and.ndraw.ne.9)) then
         write(6,*) 'invalid selection'
         go to 135
      end if
      if (ndraw.eq.0.or.ndraw.eq.1.or.ndraw.eq.9) go to 149
      go to 1451
c *** need to get this information if printing from editor
      entry getpap
      iflag=1
c *** determine orientation of drawing ***
 1451 write (*,1452) iorient
 1452 format(' (1) Portrait or (2) Landscape orientation [',i1,']:  ',$)
      read (*,145) answer
      if (answer .eq. ' ') then
         norient=iorient
      else
         read (answer,155) norient
      end if
      if (norient.lt.1.or.norient.gt.2) then
         write(6,*) 'invalid selection'
         go to 1451
      end if
c *** determine paper length for postscript landscape
      if (ndraw.eq.2.and.norient.eq.2) then
         write (*,1453) fpaplen
 1453    format(' How tall is printer page in inches? [',f5.2,']:  ',$)
         read (*,120) txtans
         if (txtans(1:1) .ne. ' ') read (txtans,1454) fpaplen
 1454    format(f10.0)
         nvar=fpaplen*1000.
      end if
c *** if called from recycle, return there
      if (iflag.eq.1) return
c *** determine where ortep output should go ***
  149 write (*,150) iout
  150 format(' Text output to (1) File, (2) Screen, or (0) Omit [',i1,
     *       ']:  ',$)
      read (*,145) answer
      if (answer .eq. ' ') then
      nou=iout
      else
      read (answer,155) nou
      end if
  155 format(i1)
c *** set output unit number ***
      nout=-4
      if (nou .eq. 1) nout=4
      if (nou .eq. 2) nout=6
c *** if output goes to a file; get its name and open the file ***
      if (nout .eq. 4) then
         ipos=index(fname,'.')
         if(ipos .ne. 0) then
            fname=fname(1:ipos-1)//ext
            go to 160
         end if
         ipos=index(fname,' ')
         fname=fname(1:ipos-1)//ext
  160    ipos=index(fname,' ')
         write (*,165) fname(1:ipos-1)
  165    format(' Enter output file name [',a,']:  ',$)
         read (*,120) txtans
         if (txtans(1:1) .ne. ' ') fname=txtans
         open(nout,file=fname,status='old',err=170)
         go to 175
  170    open(nout,file=fname,status='new')
      end if
  175 continue
      return
c *** get file name of an external file containing atomic parameters
      entry gtafil(iu)
  210 fname=atomfi
      ipos=index(fname,' ')
      write (*,215) fname(1:ipos-1)
  215 format(' Enter atom parameter file name or "exit" [',a,']:  ',$)
      read (*,120) txtans
      if (txtans(1:4).eq.'exit' .or. txtans(1:4).eq.'EXIT')
     *    call exitng(0)
      if (txtans(1:1) .ne. ' ') fname=txtans
      open(iu,file=fname,status='old',err=225)
      return
  225 ipos=index(fname,' ')
      write (*,130) fname(1:ipos-1)
      go to 210
c *** ask user about using editor
      entry go2edtr
      write (*,303)
c 303 format(/,' Edit instruction set? (Y)es or (N)o [N]:  ',$)
  303 format(/,' (1) Save drawing as Postscript file',/,
     &' (2) Save drawing as HPGL file',/,
     &' (3) Redraw structure on screen',/,
     &' (4) Edit instruction set',/,
     &' [Quit]:  ',$)
      read (*,304) answer
  304 format(a)
      if (answer.eq.'1') then
         ndraw=2
         call recycle
      else
         if (answer.eq.'2')  then
            ndraw=3
            call recycle
         else
            if (answer.eq.'3') then
               ndraw=1
               call recycle
            else
               if (answer.eq.'4') call editr
            end if
         end if
```

```
      end if
      return
      end
      SUBROUTINE UNITY(X,Z,ITYPE)
      DIMENSION X(3),Y(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      Y(1)=X(1)
      Y(2)=X(2)
      Y(3)=X(3)
      IF(ITYPE)125,125,105
  105 T1=SQRT(Y(1)*Y(1)+Y(2)*Y(2)+Y(3)*Y(3))
      GO TO 145
  125 T1=SQRT(Y(1)*(Y(1)*AA(1,1)+Y(2)*(AA(1,2)+AA(2,1))+Y(3)*(AA(1,3)+A
     1A(3,1)))+Y(2)*(Y(2)*AA(2,2)+Y(3)*(AA(2,3)+AA(3,2)))+Y(3)*Y(3)*AA(3
     2,3))
  145 IF(T1)155,155,175
  155 NG=5
      GO TO 300
  175 Z(1)=Y(1)/T1
      Z(2)=Y(2)/T1
      Z(3)=Y(3)/T1
  300 RETURN
      END
      SUBROUTINE VM(Y,X,Z)
C     TRANSPOSED VECTOR * MATRIX
C     Z(3)=Y(3)*X(3,3)
      DIMENSION X(3,3),Y(3),Z(3)
      Y1=Y(1)
      Y2=Y(2)
      Y3=Y(3)
      Z(1)=X(1,1)*Y1+X(2,1)*Y2+X(3,1)*Y3
      Z(2)=X(1,2)*Y1+X(2,2)*Y2+X(3,2)*Y3
      Z(3)=X(1,3)*Y1+X(2,3)*Y2+X(3,3)*Y3
      RETURN
      END
      FUNCTION VMV(X1,Q,X2)
C     TRANSPOSED VECTOR * MATRIX * VECTOR
C     VMV=X1(3)*Q(3,3)*X2(3)     TO  EVALUATE QUADRATIC OR BILINEAR FORM
      DIMENSION X1(3),Q(3,3),X2(3)
      VMV=X1(1)*(X2(1)*Q(1,1)+X2(2)*Q(1,2)+X2(3)*Q(1,3))
     &   +X1(2)*(X2(1)*Q(2,1)+X2(2)*Q(2,2)+X2(3)*Q(2,3))
     &   +X1(3)*(X2(1)*Q(3,1)+X2(2)*Q(3,2)+X2(3)*Q(3,3))
      RETURN
      END
      FUNCTION VV(X,Y)
C     TRANSPOSED VECTOR * VECTOR
C     VV=X(3)*Y(3)
      DIMENSION X(3),Y(3)
      VV=X(1)*Y(1)+X(2)*Y(2)+X(3)*Y(3)
      RETURN
      END
      SUBROUTINE XYZ(DQA,X,ITYPE)
```

```
C     ***** ITYPE .GT.0 CART. COORD. FROM ATOM CODE WORD *****
C     ***** XABSF(ITYPE) .LE.2 FOR WORKING SYSTEM *****
C     ***** XABSF(ITYPE) .GT.2 FOR REFERENCE SYSTEM *****
C     ***** ITYPE .LE.0 USES TRICLINIC COORD. XT *****
      DIMENSION X(3)
      REAL*8 DQA
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IT=IABS(ITYPE)-2
      NG1=NG
      NG=0
      IF(ITYPE)10,10,5
    5 CALL ATOM(DQA,XT)
      IF(NG)30,10,30
   10 T1=0.
      DO 15 J=1,3
      T2=XT(J)-ORGN(J)
      V1(J)=T2
   15 T1=T1+ABS(T2)
      IF(T1-.0001)20,20,40
   20 NG=NG1
   30 DO 35 J=1,3
   35 X(J)=0.
      GO TO 300
   40 IF(IT)45,45,60
C     ***** RELATIVE TO WORKING SYSTEM *****
   45 DO 55 I=1,3
      T1=0.
      DO 50 J=1,3
   50 T1=T1+V1(J)*AAWRK(J,I)
   55 X(I)=T1*SCAL1
      GO TO 300
C     ***** RELATIVE TO REFERENCE SYSTEM *****
   60 DO 70 I=1,3
      T1=0.
      DO 65 J=1,3
   65 T1=T1+V1(J)*AAREV(J,I)
   70 X(I)=T1*SCAL1
  300 RETURN
      END
c ****************************************************************
c *** DUMMY SCREEN OUTPUT (MAY BE REPLACED WITH SCREEN DRIVER CODE)
c ****************************************************************

c     subroutine initsc
c     return
c     end

c     subroutine penwsc(penw)
c     return
c     end

c     subroutine colrsc(icolor)
```

```
c     return
c     end

c     subroutine pensc(x,y,ipen)
c     return
c     end

c     subroutine endsc
c     return
c     end

c *** end of dummy screen output
c ****************************************************************

c ****************************************************************
c *** PGPLOT CODE FOR SCREEN OUTPUT
c *** if PGPLOT is implemented, use the subroutines here
c *** instead of the ones in the DUMMY SCREEN OUTPUT section
c     PGPLOT is a free graphics library developed by T. J. Pearson at
c     the California Institute of Technology. Information about PGPLOT
c     can be found on the World Wide Web at
c     http://astro.caltech.edu/~tjp/pgplot
c     or via e-mail to tjp@astro.caltech.edu.
c ****************************************************************

      subroutine initsc
      character*10 outdev
      common /ns/ npf,ndraw,norient,nvar
      integer pgbeg

      xwid=11.
      yhgt=8.5

c *** The following is for PGPLOT on an X-windows system.
      outdev = '/XWINDOW'
c *** The following is for PGPLOT on an MS-DOS system.
c     outdev = '/MS'
c *** The following is for PGPLOT on a Macintosh system.
c     outdev = '/MAC'

      open(npf,status='scratch')

      if (pgbeg(0,' ',1,1) .ne. 1) call exitng(8)

c     switch black and white
      call pgscr(0,1.,1.,1.)
      call pgscr(1,0.,0.,0.)

c     set up drawing window
      call pgpage
      call pgqch(osize)
      call pgsch(0.)
      call pgvstd
      call pgwnad(0.,xwid,0.,yhgt)
      call pgsch(osize)
      call pgbox('BCT',1.,0,'BCT',1.,0)
      call pgsci(1)
      call pgsfs(2)
      call pgrect(10.4,11.,8.2,8.5)
      call pgtext(10.5,8.3,'EXIT')
      return
      end

      subroutine colrsc(icolor)
c *** set plot color
c *** in ORTEP icolor=0 => black
c *** PGPLOT is set up for 1=black
      common /ns/ npf,ndraw,norient,nvar
      icol=icolor
      if (icol.eq.0) icol=1
      nvar=icol
      if (ndraw.eq.1) call pgsci(icol)
      if (ndraw.eq.9) write (npf,111) icol
  111 format('COL',1x,i2)
      return
      end

      subroutine penwsc(penw)
c *** change pen width
c *** PGPLOT measures pen width in 200ths of an inch
      common /ns/ npf,ndraw,norient,nvar
      ipenw=nint(.001*penw*200.)
      if (ipenw.le.0) ipenw=1
      if (ipenw.gt.200) ipenw=200
      if (ndraw.eq.1) call pgslw(ipenw)
      if (ndraw.eq.9) write (npf,111) ipenw
  111 format('WID',1x,i3)
      return
      end

      subroutine pensc(x,y,ipen)
c *** move the pen
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar

      if (ipen.eq.2) then
         if (ndraw.eq.1) call pgdraw(x+xtrans,y+ytrans)
         if (ndraw.eq.9) write (npf,111) x+xtrans,y+ytrans
  111    format('LIN',2(1x,f10.6))
      end if
      if (ipen.eq.3) then
         if (ndraw.eq.1) call pgmove(x+xtrans,y+ytrans)
         if (ndraw.eq.9) write (npf,112) x+xtrans,y+ytrans
  112    format('MOV',2(1x,f10.6))
      end if

      return
      end

      subroutine endsc
      common /ns/ npf,ndraw,norient,nvar

      call curssc
      close(npf)

c *** tell user to hit <enter> key
      call pgsci(0)
      call pgsfs(1)
      call pgrect(7.5,11.,8.2,8.5)
      call pgsci(1)
      call pgsfs(2)
```

```
      call pgrect(7.5,11.,8.2,8.5)
      call pgsci(1)
      call pgtext(7.6,8.3,'Hit <RETURN> or <ENTER> key')

      call pgend

      return
      end

      subroutine curssc
c *** correlate screen cursor position with atom positions and display results
      character ch
      character*21 str
      integer pgcurs
      character*6 label,alabel
      character*9 tomid,atomid
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar

      call pgsfs(1)
      call pgscf(1)
      call pgsch(1.)

c *** get cursor position
  1   junk = pgcurs(x,y,ch)

      if (ch.eq.'x' .or. ch.eq.'X') return
      if (ch.eq.'d' .or. ch.eq.'D') return
      if (x.ge.10.4 .and. x.le.11. .and. y.ge.8.2 .and. y.le.8.5) return
      if (ichar(ch).eq.13) return

c *** initial values for variables
      xpt = x
      ypt = y
      adiffx = .0625
      adiffy = .0625
      odiffx = adiffx
      odiffy = adiffy
      atomid = '         '
      alabel = '      '
      iflag = 0
      nflag = 0

      rewind(npf)

  2   read(npf,3,end=4) label,tomid,xx,yy
  3   format(11x,a6,3x,a9,4x,2f8.0)
      diffx = abs(xx-xpt)
      diffy = abs(yy-ypt)
      if (diffx.le.adiffx .and. diffy.le.adiffy) nflag=nflag+1
      if (diffx.le.odiffx .and. diffy.le.odiffy) then
         atomid = tomid
         alabel = label
         odiffx = diffx
         odiffy = diffy
      end if
      go to 2

  4   if (nflag.eq.0) write(str,5)
      if (nflag.eq.1) write(str,6) alabel,atomid
      if (nflag.gt.1) write(str,7) alabel,atomid
  5   format('Not near atom center')
  6   format(a6,1x,a9)
  7   format(a6,1x,a9,' + ??')

c *** erase rectangle
      call pgsci(0)
      call pgsfs(1)
      call pgrect(0.,2.8,8.2,8.5)
c *** redraw empty rectangle
      call pgsci(1)
      call pgsfs(2)
      call pgrect(0.,2.8,8.2,8.5)

c *** print atom information in rectangle
      call pgtext(0.1,8.3,str)

      go to 1

      end

c *** end of PGPLOT specific routines
c ******************************************************************

c ******************************************************************
c *** HPGL FILE OUTPUT
c ******************************************************************

      subroutine inithp
      common /ns/ npf,ndraw,norient,nvar
      character ESC
      character*10 fname

      do 11 i=1,999
         write (fname, 10) i
   10    format('TEP',i3.3,'.PRN')
         open(unit=npf,file=fname,status='old',err=12)
         close(npf)
   11 continue
   12 open(unit=npf,file=fname,status='new')
      write (*,13) fname
   13 format(/,' HPGL file name:  ',a)

      ESC=char(27)

      write (npf,21) ESC
      write (npf,22) ESC
      write (npf,23)
      if (norient.eq.2) write (npf,24)
   21 format(a1,'E')
   22 format(a1,'%0B')
   23 format('IN;'/'SP1;'/'PW.15;')
   24 format('RO90.;')
      return
      end

      subroutine colrhp(icolor)
      common /ns/ npf,ndraw,norient,nvar
c *** set plot color
c *** in ORTEP icolor=0 => black
c *** plotter pen 1=black
      icol=icolor
```

```
      if (icol.eq.0) icol=1
      write (npf,21) icol
   21 format('SP',i1,';')
      return
      end

      subroutine penwhp(penw)
      common /ns/ npf,ndraw,norient,nvar
      if (penw.eq.0.) then
         penw=.15
      else
         penw=penw*.0252
      end if
      write (npf,21) penw
   21 format('PW',f5.2,';')
      return
      end

      subroutine penhp(x,y,ipen)
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans

      ix = nint((x + xtrans) * 1000.)
      iy = nint((y + ytrans) * 1000.)

      if (ipen.eq.2) write (npf,101) ix,iy
  101 format('PD',i4,',',i4,';')
      if (ipen.eq.3) write (npf,102) ix,iy
  102 format('PU',i4,',',i4,';')
      return
      end


      subroutine endhp
      common /ns/ npf,ndraw,norient,nvar
      character ESC

      ESC=char(27)
      write (npf,31)
   31 format('PU;',/,'SP0;',/,'PG;',/,'IN;')
      write (npf,34) ESC
   34 format(a1,'%0A')
      write (npf,35) ESC
   35 format(a1,'E')
      close(npf)
      return
      end

c *** end of HPGL specific routines
c ***************************************************************

c ***************************************************************
c *** POSTSCRIPT FILE OUTPUT
c ***************************************************************

      subroutine initps
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt
      character*10 fname

c *** initialize variables to calculate bounding box

      ixmin=20000
      ixmax=0
      iymin=20000
      iymax=0

      do 11 i=1,999
         write (fname, 10) i
   10    format('TEP',i3.3,'.PRN')
         open(unit=npf,file=fname,status='old',err=12)
         close(npf)
   11 continue
   12 open(unit=npf,file=fname,status='new')
      write (*,13) fname
   13 format(/,' Postscript file name:  ',a)

      ixt=0
      iyt=0
      write (npf,21)
      write (npf,22)
      write (npf,23)
      if (norient.eq.2) then
         write (npf,24)
         iyt=nvar
      else
         write (npf,25)
      end if
      write (npf,26)
      write (npf,27)
      write (npf,28)
      write (npf,29)
      write (npf,30)
      write (npf,31)
      write (npf,32) ixt,iyt
      if (norient.eq.2) write (npf,33)
      write (npf,34)
      write (npf,35)
   21 format('%!PS-Adobe-3.0 EPSF-3.0')
   22 format('%%Creator: ORTEP-III')
   23 format('%%BoundingBox: (atend)',/,'%%Pages: 1')
   24 format('%%Orientation: Landscape')
   25 format('%%Orientation: Portrait')
   26 format('%%BeginProlog')
   27 format('/m {moveto} def')
   28 format('/l {lineto} def')
   29 format('%%EndProlog',/,'%%Page: 1 1')
   30 format('%%BeginPageSetup')
   31 format('0.072 0.072 scale')
   32 format(i6,1x,i6,' translate')
   33 format('-90 rotate')
   34 format('0 setgray 1 setlinecap 5 setlinewidth')
   35 format('%%EndPageSetup')
      return
      end

      subroutine colrps(icolor)
      common /ns/ npf,ndraw,norient,nvar
      write(npf,101)
  101 format('stroke')
      if (icolor.eq.0) write(npf,1)
      if (icolor.eq.1) write(npf,1)
      if (icolor.eq.2) write(npf,2)
```

```
      if (icolor.eq.3) write(npf,3)
      if (icolor.eq.4) write(npf,4)
      if (icolor.eq.5) write(npf,5)
      if (icolor.eq.6) write(npf,6)
      if (icolor.eq.7) write(npf,7)
    1 format('0 setgray')
    2 format('1 0 0 setrgbcolor')
    3 format('0 1 0 setrgbcolor')
    4 format('0 0 1 setrgbcolor')
    5 format('0 1 1 setrgbcolor')
    6 format('1 0 1 setrgbcolor')
    7 format('1 1 0 setrgbcolor')
      return
      end

      subroutine penwps(penw)
      common /ns/ npf,ndraw,norient,nvar
      write(npf,101)
  101 format('stroke')
      if (penw.eq.0.) penw=5.
      write(npf,102) penw
  102 format(f10.2,1x,'setlinewidth')
      return
      end

      subroutine penps(x,y,ipen)
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt

      ix = nint((x + xtrans) * 1000.)
      iy = nint((y + ytrans) * 1000.)

      if (ipen.eq.2) write (npf,101) ix,iy
  101 format(i6,1x,i6,1x,'l')
      if (ipen.eq.3) write (npf,102) ix,iy
  102 format('stroke'/i6,1x,i6,1x,'m')

c *** variables to calculate bounding box
      if (ix.lt.ixmin) ixmin=ix
      if (ix.gt.ixmax) ixmax=ix
      if (iy.lt.iymin) iymin=iy
      if (iy.gt.iymax) iymax=iy

      return
      end

      subroutine endps
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt

      write (npf,25)
   25 format('stroke'/'showpage')

c *** calculate bounding box
      if (norient.eq.1) then
         ixmn=float(ixmin+ixt)*.072 - 2
         iymn=float(iymin+iyt)*.072 - 2
         ixmx=float(ixmax+ixt)*.072 + 2
         iymx=float(iymax+iyt)*.072 + 2
      else
         ixmn=float(ixt+iymin)*.072 - 2
         iymn=float(iyt-ixmax)*.072 - 2
         ixmx=float(ixt+iymax)*.072 + 2
         iymx=float(iyt-ixmin)*.072 + 2
      end if
      if (ixmn.lt.0) ixmn=0
      if (iymn.lt.0) iymn=0

c *** put bounding box at end of postscript file
      write (npf,26) ixmn,iymn,ixmx,iymx
   26 format('%%BoundingBox: ',4(i6,1x))

      write (npf,27)
   27 format('%%Trailer'/'%%EOF')
      close(npf)

      return
      end

c *** end of Postscript specific routines
c ***************************************************************

c ****************************************************************
c *** DEFAULT VALUES FOR ORTEP INPUT AND OUTPUT OPTIONS ARE SET HERE
c *** USED IN SUBROUTINE UINPUT
c ****************************************************************

      subroutine dflts
      common /dfl/ infile,idraw,iorient,iout,ext,atomfi,fpaplen
      character*60 infile,atomfi
      character*4 ext

c *** name of default input file
      infile='TEP.IN'

c *** where ortep drawing output should go
c *** 1: Screen, 2: Postscript file, 3: HPGL file
      idraw=1

c *** orientation of drawing
c *** 1: portrait, 2: landscape
      iorient=1

c *** height of page
      fpaplen=11.

c *** where ortep text output should go
c *** 1: file, 2: screen, 0: omit
      iout=0

c *** text output filename extension
      ext='.out'

c *** default name of external atom parameter file
      atomfi='ATOMS.DAT'

      return
      end
```