This is the Fortran code for generating coördinates of molecules numerically using the CDNT framework.

```
C
C
C   PROGRAM:  VERSION 3, WHICH WILL CALCULATE THE CARTESIAN
C   COORDINATES FOR THE ATOMS IN A MOLECULE.  THIS PROGRAM IS DESIGNED
C   TO PUNCH A COMPLETE INPUT DECK FOR USE WITH GAMESS AND MELD
C
      PROGRAM
      IMPLICIT REAL*8(A-H,O-Z), INTEGER*2(\$)
      REAL*4 R,CHECK
      INTEGER*2 NOTE,NULL,STAR,STAR1,STAR2,ARROW,BOND,REF,ORIGIN,
     .  CMMT,ATOM
      LOGICAL*1 TRAN,DIST,NODIST,NOCHCK,COORD1,COORD2,COORD3,JUMP,
     .  NOSKIP,NOTEST,KEY,COORD
C
      COMMON/PARMS/RIJ,ZETA,ETA,EULER1,EULER2,EULER3,X1,X2,X3,Y1,Y2,
     .  Y3,Z1,Z2,Z3,RDCX,RDCY,RDCZ,II,JJ,KK,LL,MM,COORD1,COORD2,COORD3
      COMMON/ABCXYZ/AX,AY,AZ,BX,BY,BZ,CX,CY,CZ
      DIMENSION DC(3,77),C(3,75),ANGLE(76),TEMP(10),PARM(3),EULER(3),
     .  X(3),Y(3),Z(3),RDC(3),V1(3),V2(3),R1(3),R2(3),RBOND(2)
      DIMENSION R(12)
      DIMENSION IPARM(5),JJJ(2)
      DIMENSION BOND(76),REF(76),ORIGIN(76),CMMT(75),ATOM(40),\$R(12)
      DIMENSION KEY(75),COORD(3),ICHG(75),RCHG(75)
      CHARACTER*2 SYMBOL(36) /' H','HE','LI','BE',' B',' C',' N',' O',
     . ' F','NE',26*'  '/
      CHARACTER*2 TMP
C
      EQUIVALENCE (PARM(1),RIJ),(EULER(1),EULER1),(X(1),V1(1),X1),
     .  (Y(1),V2(1),Y1),(Z(1),R1(1),Z1),(RDC(1),R2(1),RDCX),
     .  (TEMP(1),R(1),ATOM(1),KEY(1)),(TEMP(10),\$TEMP),(IPARM(1),II),
     .  (COORD(1),COORD1)
C
      DATA RADDEG,ANGLE(76)/Z3F477D1A894A74E2,90.D0/,IDECK,NLESS/0,0/
      DATA STAR,STAR1,STAR2,NULL,ARROW/' *','* ','**','  ','<='/
      DATA BOND(76),REF(76),ORIGIN(76)/77,76,76/
      DATA ZERO/0.0D0/
C
C   DEFINE THE TRIGONOMETRIC FUNCTIONS FOR ANGLES IN DEGREES.
C
      SIN(DEG)=DSIN(DEG*RADDEG)
      COS(DEG)=DCOS(DEG*RADDEG)
C     RADDEG=0.1745329251994329D-01
```

```
C
C
      CALL ERRSET(215,1,1,0,0,0)
C
C   READ TROL CARD.
C
  101 READ (5,102,END=199) NBOND,NDROP,NTYPE,TRAN,DIST
  102 FORMAT (3I3,1X,3L1,2X,F10.6)
C
C   CHECK CONTROL VARIABLES.
C
      NATOM=NBOND+1
      NNATOM=NATOM-NDROP
      IF(NATOM.LT.2.OR.NATOM.GT.75.OR.NNATOM.LT.0.)
     .  GO TO 10
      NODIST=.NOT.DIST
      IF(RMIN.LT.1.D-09)  RMIN=0.5D0
      IDECK=IDECK+1
C
C   IF TRAN=.TRUE., READ COORDINATE TRANSFORMATION PARAMETERS.
C
      IF(TRAN)  READ (5,103) (EULER(I),COORD(I),X(I),Y(I),Z(I),I=1,3)
  103 FORMAT (F19.15,L1,3F10.6)
C
  111 FORMAT (10A8)
 1111 FORMAT ('0',10A8)
      READ(5,111) TEMP
      WRITE(6,1111) TEMP
C
C   PRINT ATOM COUNT AND OPTIONS.
C
      WRITE (6,112) NATOM,NNATOM
  112 FORMAT ('0'//T10,'ATOM COUNT:     TOTAL ATOMS =',I3,5X,
     .  'PERTINENT ATOMS =',I3/////T10,'OPTIONS IN EFFECT:')
      IF(DIST)  WRITE (6,113)
  113 FORMAT (T20,'DISTANCE MATRIX')
      WRITE (6,114)
  114 FORMAT (T20,'INTERATOMIC DISTANCES FOR PERTINENT ATOMS')
      IF(.NOT.NOCHCK)  WRITE (6,115) RMIN
  115 FORMAT ('+',T62,'CHECKED  (RMIN =',F10.6,')')
      IF(NOCHCK)  WRITE (6,116)
  116 FORMAT ('+',T62,'NOT CHECKED')
  117 IF(.NOT.TRAN)  GO TO 127
      WRITE (6,118)
  118 FORMAT (T20,'COORDINATE TRANSFORMATION')
C
```

```
C   PROVISION FOR COORDINATE TRANSFORMATION VIA EULER ANGLE ROTATIONS.
C
      IF(COORD2)  GO TO 120
      COS2=COS(EULER2)
      SIN2=SIN(EULER2)
      \$ELR2=NULL
      IF(DABS(SIN2).LT.1.D-08)  GO TO 121
      IF(COORD1)  GO TO 119
      COS1=COS(EULER1)
      SIN1=SIN(EULER1)
      \$ELR1=NULL
      GO TO 123
  119 RR=DSQRT(X1**2+Y1**2)
      COS1=X1/RR

      SIN1=Y1/RR
      GO TO 122
  120 RR1=DSQRT(X2**2+Y2**2)
      RR2=DSQRT(RR1**2+Z2**2)
      COS2=Z2/RR2
      SIN2=RR1/RR2
      EULER2=ARCCOS(COS2)
      \$ELR2=STAR2
      IF(SIN2.LT.1.D-08)  GO TO 121
      COS1=-Y2/RR1
      SIN1=X2/RR1
      GO TO 122
  121 COS1=1.D0
      SIN1=0.D0
  122 EULER1=ARCCOS(COS1)
      IF(SIN1.LT.-1.D-08)  EULER1=-EULER1
      \$ELR1=STAR2
  123 IF(COORD3)  GO TO 124
      COS3=COS(EULER3)
      SIN3=SIN(EULER3)
      \$ELR3=NULL
      GO TO 125
  124 RR=DSQRT(X3**2+Y3**2+Z3**2)
      COS3=(COS1*X3+SIN1*Y3)/RR
      SIN3=(-SIN1*X3+COS1*Y3)/RR
      IF(DABS(Z3).GT.1.D-09)  SIN3=COS2*SIN3+SIN2*(Z3/RR)
      EULER3=ARCCOS(COS3)
      IF(SIN3.LT.1.D-08)  EULER3=-EULER3
      \$ELR3=STAR2
C
C   **  CONSTRUCT THE TRANSFORMATION MATRIX.
```

```
C
  125  AX=COS1*COS3-SIN1*COS2*SIN3
       AY=SIN1*COS3+COS1*COS2*SIN3
       AZ=SIN2*SIN3
       BX=-COS1*SIN3-SIN1*COS2*COS3
       BY=-SIN1*SIN3+COS1*COS2*COS3
       BZ=SIN2*COS3
       CX=SIN1*SIN2
       CY=-COS1*SIN2
       CZ=COS2
C
C   **  PRINT THE TRANSFORMATION MATRIX AND TEST ITS ORTHONORMALITY.
C
       WRITE (6,126) AX,AY,AZ,EULER1,\$ELR1,COS1,SIN1,BX,BY,BZ,
     .  EULER2,\$ELR2,COS2,SIN2,CX,CY,CZ,EULER3,\$ELR3,COS3,SIN3
  126  FORMAT ('0'//T10,'COORDINATE TRANSFORMATION:'///T27,
     .  'TRANSFORMATION MATRIX',T78,'EULER ANGLE',T100,'COSINE',T117,
     .  'SINE'/3(/T10,3F16.8,15X,F16.8,A2,2F16.8))
       DET=AX*(BY*CZ-BZ*CY)+AY*(BZ*CX-BX*CZ)+AZ*(BX*CY-BY*CX)
       IF(DABS(DET-1.D0).GT.1.D-06)  GO TO 20
C
C   **  TRANSFORM THE DIRECTIONAL COSINES OF THE REFERENCE VECTORS.
C
       DC(1,76)=AZ
       DC(2,76)=BZ
       DC(3,76)=CZ
       DC(1,77)=-AX
       DC(2,77)=-BX

       DC(3,77)=-CX
       GO TO 128
C
C   SET THE DIRECTIONAL COSINES FOR THE REFERENCE VECTORS IN THE
C   ABSENCE OF ANY COORDINATE TRANSFORMATION.
C
  127  DC(1,76)=0.D0
       DC(2,76)=0.D0
       DC(3,76)=1.D0
       DC(1,77)=-1.D0
       DC(2,77)=0.D0
       DC(3,77)=0.D0
C
C   INITIALIZE THE BOND COUNT AND ATOM VERIFICATION PARAMETERS.
C
  128  IBOND=0
       JUMP=.FALSE.
```

```
      DO 129  I=1,NNATOM
      KEY(I)=.FALSE.
      CMMT(I)=NULL
  129 CONTINUE
      IF(NDROP.EQ.0)  GO TO 131
      NGO=NNATOM+1
      DO 130  I=NGO,NATOM
      KEY(I)=.FALSE.
      CMMT(I)=STAR2
  130 CONTINUE
C
C   PRINT HEADING FOR BOND DEFINITIONS.
C
  131 WRITE (6,132)
  132 FORMAT ('0'//T10,'BOND DEFINITIONS:'///T11,'ATOMS',T23,'BOND',
     .  T35,'BOND',T45,'DIHEDRAL',T65,'COORDINATES OF I',T100,
     .  'COORDINATES OF J',T126,'CHECK ON'/T3,'BOND',T11,'I',T15,'J',
     .  T22,'LENGTH',T34,'ANGLE',T46,'ANGLE',T61,'X',T72,'Y',T83,'Z',
     .  T96,'X',T107,'Y',T118,'Z',T127,'LENGTH'/)
C
C   READ THE COORDINATES OF THE INITIAL ATOM.
C
      READ (5,133) XX,YY,ZZ
  133 FORMAT (3F10.6)
C
C   READ THE BOND DEFINITION CARD.
C
  134 IF(IBOND.GE.NBOND)  GO TO 158
      IBOND=IBOND+1
      READ (5,135) IPARM,PARM
  135 FORMAT (5I2,3F10.6)
      IF(KK.NE.0)  GO TO 144
      IMAX=MAX0(II,JJ)
  136 IF(IMAX.GT.NATOM)  GO TO 30
C
C   INSERT THE COORDINATES OF THE INITIAL ATOM.
C
      IF(JUMP)  GO TO 137
      KEY(II)=.TRUE.
      BOND(II)=76
      C(1,II)=XX
      C(2,II)=YY

      C(3,II)=ZZ
      REF(II)=76
      JUMP=.TRUE.
```

```
C
C   "STANDARD" BOND DEFINITION.
C
  137  IF(.NOT.KEY(II).OR.KEY(JJ))  GO TO 40
       IF(RIJ.LT.1.D-09)  GO TO 50
       KEY(JJ)=.TRUE.
       ORIGIN(IBOND)=II
       BOND(JJ)=IBOND
       ANGLE(IBOND)=ZETA
       IREF1=BOND(II)
       IREF=REF(II)
       SIN1=SIN(ZETA)
       COS1=COS(ZETA)
C
C   **  TEST "COLLINEARITY" OF THE BOND.
C
       IF(DABS(SIN1).GT.1.D-08)  GO TO 138
       SENSE=1.D0
       IF(COS1.GT.1.D-06)  SENSE=-1.D0
       RDCX=SENSE*DC(1,IREF1)
       RDCY=SENSE*DC(2,IREF1)
       RDCZ=SENSE*DC(3,IREF1)
       REF(JJ)=IREF*IDINT(SENSE)
       GO TO 141
C
C   **  DETERMINE THE DIRECTIONAL COSINES OF THE BOND VECTOR BY
C   **  CALLING SUBROUTINE LOCAL.
C
  138  AA=SIN1*COS(ETA)
       BB=SIN1*SIN(ETA)
       CC=-COS1
       IF(IREF.GT.0)  GO TO 139
       IREF=-IREF
       ZREF=180.D0-ANGLE(IREF)
       GO TO 140
  139  ZREF=ANGLE(IREF)
  140  IATOM=ORIGIN(IREF)
       IREF2=BOND(IATOM)
       CALL LOCAL(ZREF,DC(1,IREF1),DC(2,IREF1),DC(3,IREF1),
     .  DC(1,IREF2),DC(2,IREF2),DC(3,IREF2))
       RDCX=AX*AA+BX*BB+CX*CC
       RDCY=AY*AA+BY*BB+CY*CC
       RDCZ=AZ*AA+BZ*BB+CZ*CC
       REF(JJ)=IBOND
C
C   **  PERFORM CHECK ON BOND LENGTH.
```

```
C
  141  CHECK=RIJ*DSQRT(RDCX**2+RDCY**2+RDCZ**2)
C
C   **  DETERMINE THE COORDINATES FOR THE ATOM BEING DEFINED.
C
       DO 142  I=1,3
       RR=RDC(I)
       DC(I,IBOND)=RR
       C(I,JJ)=C(I,II)+RR*RIJ
  142  CONTINUE

C
C   **  PRINT THE BOND DEFINITION PARAMETERS.
C
       WRITE (6,143) IBOND,NULL,II,JJ,PARM,(C(I,II),I=1,3),
     .  (C(J,JJ),J=1,3),CHECK
  143  FORMAT (T4,I2,A2,2I4,3F12.4,2(2X,3F11.4),F11.4)
       GO TO 134
C
C   BOND DEFINITION VIA BISECTOR FEATURE OPTION.
C   NOTE:  IF II=JJ, THEN ONLY ONE ATOM IS BEING DEFINED.
C
  144  IMAX=MAX0(II,JJ,KK,LL,MM)
       IF(IMAX.GT.NATOM)  GO TO 30
       IF(KEY(II).OR.KEY(JJ).OR..NOT.KEY(KK).OR..NOT.KEY(LL).OR.
     .  .NOT.KEY(MM))  GO TO 40
       WRITE (6,145) IPARM,PARM
  145  FORMAT (T3,'-----  BISECTOR FEATURE OPTION  -----',4X,2I4,1X,
     .  3I4,1X,2(F10.4,'*'),F12.4,'*')
       IF(II.NE.JJ)  GO TO 146
       IF(RIJ.LT.1.D-09)  GO TO 50
       NOTE=STAR1
       COS1=1.D0
       SIN1=0.D0
       NDEF=1
       GO TO 147
  146  IF(RIJ.LT.1.D-09)  RIJ=1.09D0
       IF(ZETA.LT.1.D-09)  ZETA=1.09D0
       ALPHA=ETA/2.D0
       IF(ALPHA.LT.1.D-08)  ALPHA=54.7356103172453D0
       NOTE=STAR2
       COS1=COS(ALPHA)
       SIN1=SIN(ALPHA)
       NDEF=2
       JJJ(2)=JJ
       RBOND(2)=ZETA
```

```
  147  JJJ(1)=II
       RBOND(1)=RIJ
C
C   **  DEFINE THE BISECTOR PLANE.
C
       DO 148  I=1,3
       CREF=C(I,LL)
       R1(I)=CREF-C(I,KK)
       R2(I)=CREF-C(I,MM)
  148  CONTINUE
       RR1=DSQRT(R1(1)**2+R1(2)**2+R1(3)**2)
       RR2=DSQRT(R2(1)**2+R2(2)**2+R2(3)**2)
       RDOT=DABS(R1(1)*R2(1)+R1(2)*R2(2)+R1(3)*R2(3))/(RR1*RR2)
       IF(DABS(RDOT-1.D0).LT.1.D-06)  GO TO 60
       DO 149  I=1,3
       V1(I)=(R1(I)/RR1+R2(I)/RR2)/2.D0
  149  CONTINUE
       V2(1)=R1(2)*R2(3)-R1(3)*R2(2)
       V2(2)=R1(3)*R2(1)-R1(1)*R2(3)
       V2(3)=R1(1)*R2(2)-R1(2)*R2(1)
       VV1=DSQRT(V1(1)**2+V1(2)**2+V1(3)**2)
       VV2=DSQRT(V2(1)**2+V2(2)**2+V2(3)**2)
       DO 150  I=1,3
       V1(I)=V1(I)/VV1

       V2(I)=V2(I)/VV2
  150  CONTINUE
C
C   **  DETERMINE THE LOCAL COORDINATE SYSTEM.
C
       IREF1=BOND(LL)
       IREF=REF(LL)
       IF(IREF.GT.0)  GO TO 151
       IREF=-IREF
       ZREF=180.D0-ANGLE(IREF)
       GO TO 152
  151  ZREF=ANGLE(IREF)
  152  IATOM=ORIGIN(IREF)
       IREF2=BOND(IATOM)
       CALL LOCAL(ZREF,DC(1,IREF1),DC(2,IREF1),DC(3,IREF1),
     .  DC(1,IREF2),DC(2,IREF2),DC(3,IREF2))
C
C   **  DETERMINE THE DIRECTIONAL COSINES OF THE BOND VECTORS AND
C   **  THE COORDINATES OF THE ATOMS BEING DEFINED.
C
       DO 157  N=1,NDEF
```

```
      IF(N.EQ.1)  GO TO 153
      IBOND=IBOND+1
      SIN1=-SIN1
  153 JJ=JJJ(N)
      KEY(JJ)=.TRUE.
      ORIGIN(IBOND)=LL
      BOND(JJ)=IBOND
      RIJ=RBOND(N)
      DO 154  I=1,3
      RR=COS1*V1(I)+SIN1*V2(I)
      RDC(I)=RR
      DC(I,IBOND)=RR
      C(I,JJ)=C(I,LL)+RR*RIJ
  154 CONTINUE
C
C   **  PERFORM THE CHECK ON THE BOND LENGTH.
C
      CHECK=RIJ*DSQRT(RDCX**2+RDCY**2+RDCZ**2)
C
C   **  DETERMINE THE BOND ANGLE AND DIHEDRAL ANGLE.
C
      COS2=-(CX*RDCX+CY*RDCY+CZ*RDCZ)
      ZETA=ARCCOS(COS2)
      SIN2=SIN(ZETA)
      IF(DABS(SIN2).GT.1.D-08)  GO TO 155
      IREF=REF(LL)
      IF(COS2.GT.1.D-06)  IREF=-IREF
      REF(JJ)=IREF
      ETA=0.D0
      GO TO 156
  155 COS3=(AX*RDCX+AY*RDCY+AZ*RDCZ)/SIN2
      SIN3=(BX*RDCX+BY*RDCY+BZ*RDCZ)/SIN2
      ETA=ARCCOS(COS3)
      IF(SIN3.LT.-1.D-08)  ETA=-ETA
      REF(JJ)=IBOND
  156 ANGLE(IBOND)=ZETA
C
C   **  PRINT THE BOND DEFINITION PARAMETERS.

C
      WRITE (6,143) IBOND,NOTE,LL,JJ,PARM,(C(I,LL),I=1,3),
     .  (C(J,JJ),J=1,3),CHECK
  157 CONTINUE
      GO TO 134
C
C   PRINT THE COORDINATES OF THE ATOMS.
```

```
C
  158  WRITE (6,159)
  159  FORMAT ('0'//T10,'COORDINATES OF THE ATOMS:'///T10,'ATOM',T24,
     .  'X',T39,'Y',T54,'Z',T74,'ATOM',T88,'X',T103,'Y',T118,'Z'/)
       NSTOP=NATOM/2
       NREF=NATOM-NSTOP
       DO 161  N=1,NSTOP
       NN=NREF+N
       WRITE (6,160) N,CMMT(N),(C(I,N),I=1,3),NN,CMMT(NN),
     .  (C(J,NN),J=1,3)
  160  FORMAT (T10,I3,A2,3F15.8,T74,I3,A2,3F15.8)
  161  CONTINUE
       IF(NREF.GT.NSTOP)  WRITE (6,160) NREF,CMMT(NREF),
     .  (C(I,NREF),I=1,3)
C
       READ(5,995) (ICHG(I),I=1,NNATOM)
995    FORMAT(40I2)
       DO  996  I=1,NNATOM
996    RCHG(I)=DFLOAT(ICHG(I))
C
C
C   IF DIST=.TRUE., DETERMINE AND PRINT THE DISTANCE MATRIX.
C   IF NOCHCK=.FALSE., CHECK INTERATOMIC DISTANCES BETWEEN THE
C   PERTINENT ATOMS.
C
  167  IF(NODIST.AND.NOCHCK)  GO TO 101
       WRITE (6,168)
  168  FORMAT ('0'/)
       IF(NODIST)  GO TO 170
       WRITE (6,169)
  169  FORMAT (T10,'DISTANCE MATRIX:')
       NCOL=12
       NLIM=NATOM

       GO TO 171
  170  NCOL=NNATOM
       NLIM=NNATOM
  171  NOSKIP=NODIST
C
       NSTOP=0
  172  NGO=NSTOP+1
       NSTOP=NSTOP+NCOL
       NSTOP=MIN0(NSTOP,NLIM)
C
       IF(NODIST)  GO TO 175
       WRITE (6,173) (N,N=NGO,NSTOP)
```

```
  173  FORMAT ('0'/T4,12(8X,I2))
       WRITE (6,174)
  174  FORMAT (' ')
  175  IF(NGO.LE.NNATOM)  NOTEST=.FALSE.
       IF(NOCHCK)  NOTEST=.TRUE.
       DO 185  I=NGO,NLIM
       XX=C(1,I)
       YY=C(2,I)
       ZZ=C(3,I)
       IF(NOTEST)  GO TO 176
  176  NNSTOP=MIN0(NSTOP,I)
       KCOL=0
       DO 182  J=NGO,NNSTOP
       IF(I.EQ.J)  GO TO 179
       RR=DSQRT((C(1,J)-XX)**2+(C(2,J)-YY)**2+(C(3,J)-ZZ)**2)
C
C   **  CHECK FOR DISTANCES LESS THAN 1.0.
C
       II=RR+1.D-08
       IF(II.NE.0)  GO TO 180
       \$RR=STAR2
C
C   **  CHECK FOR DISTANCES LESS THAN RMIN.
C
       IF(RR.GE.RMIN)  GO TO 181
       IF(NOTEST)  GO TO 181
       NLESS=NLESS+1
       IF(NOSKIP)  GO TO 177
       NOSKIP=.TRUE.
       WRITE (6,174)
  177  WRITE (6,178) J,I,RR
  178  FORMAT (T25,'***** DISTANCE BETWEEN ATOMS',I3,' AND',I3,' IS ',
     .  'LESS THAN RMIN  (DISTANCE =',F9.6,') *****')
       \$RR=ARROW
       GO TO 181
  179  RR=0.D0
  180  \$RR=NULL
  181  IF(NODIST)  GO TO 182
       KCOL=KCOL+1
       R(KCOL)=RR
       \$R(KCOL)=\$RR
  182  CONTINUE
       IF(NODIST)  GO TO 185
       IF(.NOT.NOSKIP)  GO TO 183
       NOSKIP=.FALSE.
       WRITE (6,174)
```

```
  183  WRITE (6,184) I,(R(K),\$R(K),K=1,KCOL)

  184  FORMAT (T4,I2,2X,12(F8.4,A2))
  185  CONTINUE
       IF(NLIM.GT.NSTOP)  GO TO 172
       IF(NLESS.NE.0)  GO TO 70
C
C   PUNCH THE COORDINATES OF THE PERTINENT ATOMS.
C
       DO  997  N=1,NNATOM
       DO  998  I=1,3
998    C(I,N) =C(I,N)/(0.529177249D0)
997    CONTINUE
       DO  9999  N=1,NNATOM
       TMP=SYMBOL(ICHG(N))
       IF(N.GT.9) GO TO 9998
       WRITE(7,999) TMP,N,(C(I,N),I=1,3),RCHG(N),ZERO
       GO TO 9999
9998   WRITE(7,9898) TMP,N,(C(I,N),I=1,3),RCHG(N),ZERO
9999   CONTINUE
999    FORMAT(A2,I1,2X,3F15.10,5X,F5.2,2X,F5.2)
9898   FORMAT(A2,I2,1X,3F15.10,5X,F5.2,2X,F5.2)
       GO TO 101
C
C   PRINT ERROR (TERMINATION) MESSAGES.
C
   10  WRITE (6,15) NBOND,NDROP,NADD
   15  FORMAT ('1',T10,'***ERROR***  INVALID TROL VARIABLE',
     .  ':'/T23,'NBOND =',I3,5X,'NDROP =',I3,5X,'NADD =',I3)
       STOP 20
   20  WRITE (6,25) DET
   25  FORMAT ('0',T10,'***ERROR***  TRANSFORMATION MATRIX NOT ORTHO',
     .  'NORMAL:'/T23,'DET =',F12.8)
       STOP 20
   30  WRITE (6,35) IPARM
   35  FORMAT ('0',T10,'***ERROR***  INVALID ATOM INDEX NUMBER:'/T23,
     .  'I =',I3,5X,'J =',I3,5X,'K =',I3,5X,'L =',I3,5X,'M =',I3)
       STOP 20
   40  WRITE (6,45) IPARM
   45  FORMAT ('0',T10,'***ERROR***  ATOM DEFINITION CONFLICT:'/T23,
     .  'I =',I3,5X,'J =',I3,5X,'K =',I3,5X,'L =',I3,5X,'M =',I3)
       STOP 20
   50  WRITE (6,55)
   55  FORMAT ('0',T10,'***ERROR***  UNSPECIFIED BOND LENGTH')
       STOP 20
   60  WRITE (6,65) KK,LL,MM
```

```
   65  FORMAT ('0',T10,'***ERROR***  BISECTOR PLANE CANNOT BE ',
     .  'DEFINED (ATOMS',2(I3,','),I3,' NEAR COLLINEAR)')
       STOP 20
   70  WRITE (6,75) NLESS
   75  FORMAT ('0',T10,'***ERROR*** ',I5,' INTERATOMIC DISTANCE(S) ',
     .  'LESS THAN RMIN')
       STOP 20
  199  STOP
       END
C        <<<<< SUBROUTINE: LOCAL >>>>>
       SUBROUTINE LOCAL(ZREF,DCX1,DCY1,DCZ1,DCX2,DCY2,DCZ2)
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON/ABCXYZ/AX,AY,AZ,BX,BY,BZ,CX,CY,CZ
       DATA RADDEG/Z3F477D1A894A74E2/
C
C   DEFINE THE TRIGONOMETRIC FUNCTIONS FOR ANGLES IN DEGREES.
C
       CSC(DEG)=1.D0/DSIN(DEG*RADDEG)
       COT(DEG)=DCOTAN(DEG*RADDEG)
       ACOT=-COT(ZREF)
       ACSC=-CSC(ZREF)
       AX=ACOT*DCX1+ACSC*DCX2
       AY=ACOT*DCY1+ACSC*DCY2
       AZ=ACOT*DCZ1+ACSC*DCZ2
       BX=AZ*DCY1-AY*DCZ1
       BY=AX*DCZ1-AZ*DCX1
       BZ=AY*DCX1-AX*DCY1
       CX=DCX1
       CY=DCY1
       CZ=DCZ1
       RETURN

       END
C        <<<<< FUNCTION: ARCCOS >>>>>
       DOUBLE PRECISION FUNCTION ARCCOS(COSINE)
       IMPLICIT REAL*8(A-H,O-Z)
       DATA RADDEG/Z3F477D1A894A74E2/
       ICOS=COSINE
       IF(ICOS.NE.0)  COSINE=ICOS
       ARCCOS=DARCOS(COSINE)/RADDEG
       RETURN
       END
```