PROGRAMME

'DONNEES ET GEOMETRIES'

```
// EXEC FORTRAN
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DIMENSION XB(40),YB(40),ZB(40)
      DIMENSION IDENT(60),ID(40,4),X(40),Y(40),Z(40),ATNO(40),BOD(40)
      DIMENSION BOA(40),PAN(40),TAU(40)
      DIMENSION NENT(40),NSORT(40)
      DIMENSION RCOOR(120,3),ITRI(40),JTRI(40),IAXE(40),JAXE(40)
      DIMENSION MTAB(40),LTAB(40)
      DOUBLE PRECISION RCOOR
      DOUBLE PRECISION ZD
      DOUBLE PRECISION XYZ,PIGN
      DOUBLE PRECISION X,Y,Z,ATNO,BOD,BOA,PAN,TAU
      DOUBLE PRECISION R,XB,YB,ZB
      DOUBLE PRECISION REP,REP1,REP2
      DOUBLE PRECISION PXY,PYZ,PZX
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON ATNO,BOD,BOA,PAN,TAU,X,Y,Z
      COMMON NAT,NATOM,NTRI,NM1,NM2,NOR,NPROXY,NCOORD,NSYMET
      COMMON IDENT,IUNIT,IPUNCH,ID,ITRI,IAXE
      COMMON NENT,NSORT
      COMMON NVNUM
  106 FORMAT(/2X,11(1X,F7.4))
  108 FORMAT(/1X,I2,3(2X,F11.8))
  110 FORMAT(3A8,A7)
  402 FORMAT('OLA DIMENSION DONNEE AU TABLEAU DZETA EST DIFFERENTEDE
     1 8 OU 6')
 1049 FORMAT('OENERGIE DE REPULSION NUCLEAIRE')
 1051 FORMAT('0 AVEC H=1   C=6  N=7  O=8   REP= ',F15.8)
 1050 FORMAT('0 AVEC H=1  C=4  N=5  O=6    REP= ',F15.8)
 4013 FORMAT('OERREUR SUR LES DISTANCES  LA DISTANCE ',I2,'-',I2,
     1'VAUT',F11.7,'AU LIEU DE ',F11.7)
 4016 FORMAT('OERREUR SUR LES COORDONNEES LES POINTS ',I2,'ET',I2,
     1'NE SE CORRESPONDENT PAS')
 6679 FORMAT('OAXE X, AXE Y',F12.8,'  AXE Y,AXE Z',F12.8,
     1'   AXE Z , AXE X ',F12.8)
 6681 FORMAT(1H1)
 7779 FORMAT('OCOSINUS RESPECTIF DES AXES DU TRIEDRE DE REFERENCE')
 7783 FORMAT(16I5)
 7784 FORMAT('ONUMEROTATION D ENTREE')
 7785 FORMAT(1H0,15I5)
 7786 FORMAT('ONUMEROTATION DE SORTIE')
      CALL PRELEC(LEC)
```

```
    1 CONTINUE
      CALL LECTUR
            CALL DIST(X,Y,Z,R,NAT)
      IF(NVNUM)5011,5011,4000
 4000 CONTINUE
      WRITE(IMP,7784)
      WRITE(IMP,7785)(NENT(I),I=1,NAT)
      WRITE(IMP,7786)
      WRITE(IMP,7785)(NSORT(I),I=1,NAT)
      CALL PERMUT(NSORT,ITRI,IAXE,X,Y,Z,R,ID,BOD,NE,NPROXY,NAT,NATOM
 5011 CONTINUE
      IF(NPROXY)4080,4010,4080
 4080 CONTINUE
  122 DO 12 I=2,NAT
      J=ID(I,4)
      PIGN=R(I,J)
 4011 IF(DABS(PIGN-BOD(I))-1.D-3)4010,4010,4012
 4012 WRITE(IMP,4013)I,J,PIGN,BOD(I)
      STOP
   12 CONTINUE
 4010 CONTINUE
      N522=NPROXY
      NM12=NM1+NM2
      IF(NOR)401,401,5200
 5200 CONTINUE
      CALL TRANS(X,Y,Z,NAT,NOR)
  401 IF(NM12)403,403,404
  404       CALL ROTAX(X,Y,Z,PXY,PYZ,PZX,NAT,NM1,NM2)
      WRITE(IMP,7779)
      WRITE(IMP,6679)PXY,PYZ,PZX
  403 CONTINUE
      IF(NSYMET)4031,4030,4031
 4031 CONTINUE
      DO 4030 ISYMET=1,NSYMET
            CALL VSYMET(X,Y,Z,NA,ISORT,IP1,IP2)
      IF(ISORT)4014,4015,4014
 4015 WRITE(IMP,4016)IP1,IP2
      WRITE(IMP,108)IP1,X(IP1),Y(IP1),Z(IP1)
      WRITE(IMP,108)IP2,X(IP2),Y(IP2),Z(IP2)
      STOP
 4014 CONTINUE
 4030 CONTINUE
            CALL ECRICD(X,Y,Z,R,IDENT,NAT,IUNIT,IMP,IPERF,IPUNCH)
C     ------------------------------------------------------------
C                 COMMENTAIRE
```

```
C                 ----------
C    REP1   CALCUL DE L'ENERGIE DE REPULSION NUCLEAIRE  DANS LE
C    CAS OU CHAQUE ATOME PORTE UNE CHARGE EGALE A SON NUMERO ATOMIQUE
C    ------------------------------------------------------------------
C    REP2   CALCUL DE L'ENERGIE DE REPULSION NUCLEAIRE  DANS LE
C   CAS OU CHAQUE ATOME  PORTE LA CHARGE NUCLEAIRE Q'IL AURAIT S'IL N'ET
C    ENTOURE QUE PAR SES ELECTRONS DE VALENCE
C    ------------------------------------------------------------------
            CALL PERN1(MTAB,LTAB,NATOM)
      CALL EREPNU(MTAB,NATOM,REP1)
      CALL EREPNU(LTAB,NATOM,REP2)
      WRITE(IMP,1049)
      WRITE(IMP,1050)REP1
      WRITE(IMP,1051)REP2
      CALL CLASCO(X,Y,Z,ITRI,IAXE,NAT,RCOOR)
      IF(NDIMZD-8)70,71,72
   70 IF(NDIMZD-6)74,73,74
   73 CONTINUE
      CALL PREC46
      GO TO 75
   74 WRITE(IMP,402)
      GO TO 75
   71 CONTINUE
      CALL PREC48
      GO TO 75
   72 WRITE(IMP,402)
   75 CONTINUE
      MATOM=3*NATOM
      WRITE(IDISC)((RCOOR(IDERR,JDERR),JDERR=1,3),IDERR=1,MATOM)
   99 STOP
      END
```

```
// EXEC FORTRAN
      SUBROUTINE PRELEC(LEC)
      LEC=1
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE LECTUR
      DIMENSION IDENT(60),ID(40,4),X(40),Y(40),Z(40),ATNO(40),BOD(40),
     1BOA(40),PAN(40),TAU(40)
      DIMENSION ZD(4,8)
      DIMENSION COEFO(6),COEFN(6),COEFC(6),COEFH(6)
      DIMENSION NE(40)
      DIMENSION R(40,40)
      DIMENSION NENT(40),NSORT(40),ITRI(40),IAXE(40)
      DIMENSION COEF(4,3,8)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DIMENSION Q(32)
      DOUBLE PRECISION Q
      DOUBLE PRECISION COEFO,COEFN,COEFC,COEFH
      DOUBLE PRECISION COEF
      DOUBLE PRECISION OZ,ZN,CZ,HZ,OW,WN,CW,HW
      DOUBLE PRECISION ZD,R,ATNO,BOD,BOA,PAN,TAU,X,Y,Z
      DOUBLE PRECISION TKO,SO,ALFAO
      INTEGER OAVSP,OAVS
      INTEGER OAV2P
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON ATNO,BOD,BOA,PAN,TAU,X,Y,Z
      COMMON NAT,NATOM,NTRI,NM1,NM2,NOR,NPROXY,NCOORD,NSYMET
      COMMON IDENT,IUNIT,IPUNCH,ID,ITRI,IAXE
      COMMON NENT,NSORT
      COMMON NVNUM
   99 FORMAT(I5)
  100 FORMAT(60A1)
  101 FORMAT(1H1,60A1)
  102 FORMAT(I2,3X,F5.1,F10.8,F15.8,3X,4I2,4X,2F10.6,6X,2I2)
  103 FORMAT(2X,'ATOME NO.',1X,'NUMERO ATOMIQUE',3X,'BOD',8X,'BOA',9X,'K
     1',2X,'L',2X,'M',2X,'N',8X,'PAN',10X,'TAU',/)
  104 FORMAT(4X,I2,8X,F4.1,9X,F6.4,7X,F10.6,2X,4I2,2X,2(3X,F8.4),4X,2I2/
     1)
 4082 FORMAT(8F10.6)
 4083 FORMAT(16I5)
 5510 FORMAT(4F10.4)
 5511 FORMAT(8F9.6)
 5512 FORMAT(8F10.6)
 7100 FORMAT(16I5)
 7101 FORMAT(16I2)
 7775 FORMAT(16(I2,5X))
 7776 FORMAT('0 POUR EFFECTUER LE PROGRAMME PROXYZ ')
```

```
 7777 FORMAT('ONAT    NATOM  NTRI   NCOORD NPROXY NOR    NM1    NM2    ',
     1' NVNUM  NSYMET IUNIT  IPUNCH')
 7778 FORMAT('OLES AXES SONT DETERMINES DE LA MANIERE SUIVANTE')
 7779 FORMAT('OL ORIGINE EST L ATOME NUMERO',I2)
 7780 FORMAT('OL AXE DES X EST DETERMINE PAR LES ATOMES ',I2,'ET',I2)
 7781 FORMAT('OL AXE DES Y EST DETERMINE PAR LES ATOMES ',I2,'ET',I2)
 7782 FORMAT('OL AXE DES Z EST PREPENDICULAIRE AU PLAN XY ET LE TRIEDRE
     1XYZ EST DIRECT')
 7783 FORMAT(16I5)
 7784 FORMAT('ONUMEROTATION D ENTREE')
 7785 FORMAT(1HO,15I5)
 7786 FORMAT('ONUMEROTATION DE SORTIE')
 7787 FORMAT(3F20.14,16X,2I2)
 8000 FORMAT(1HO,3F15.8,10X,2I2)
C    ------------------------------------------------------------------
C                COMMENTAIRE
C                -----------
C    LEC    EST LE NUMERO  DE L UNITE DE LECTURE
C   IPERF    EST L UNITE DE PERFORATION DE CARTES
C    IMP  EST LE NUMERO DE L UNITE D ECRITURE SUR IMPRIMANTE
C    IDISC    EST LE NUMERO  DE LA PREMIERE UNITE DE BANDE OU DISQUE
C   JDISC     EST LE NUMERO  DE LA SECONDE  UNITE DE BANDE OU DISQUE
C    KDISC    EST LE NUMERO  DE LA TROISIEME UNITE DE BANDE OU DISQUE
C   LDISC  EST LE NUMERO DE LA QUATRIEME UNITE DE BANDE OU DISQUE
C   MDISC  EST LE NUMERO DE LA CINQUIEME UNITE DE BANDE OU DISQUE
C   NDISC  EST LE NUMERO DE LA SIXIEME   UNITE DE BANDE OU DISQUE
C
C    SUR LA 360/40  DE GRENOBLE C.E.N.G.  ON A UTILISE
C    LEC=1
C    IPERF=2
C    IMP=3
C    NDISC=4
C    MDISC=5
C    LDISC=6
C    IDISC=7
C    JDISC=8
C    KDISC=9
C
C
C      NESP
C      ----
C                NOMBRE DE TYPES D ATOMES DIFFERENTS QUE PEUT TRAITER
C   LE PROGRAMME
C      NOUS NOUS SOMMES LIMITES A QUATRE TYPES
C    OXYGENE  DE TYPE  NUMERO  =  NTYPO
```

```
C    AZOTE     DE TYPE  NUMERO  =  NTYPN
C   CARBONE    DE TYPE  NUMERO  =  NTYPC
C   OXYGENE    DE TYPE  NUMERO  =  NTYPH
C
C    CONDITIONS
C    ---------
C   ON DOIT PRENDRE NTYPH DIFFERENT DE 1
C    LE PRODUIT DE NTYPH PAR LUI-MEME DOIT ETRE DIFFERENT DU PRODUIT
C   DE DEUX  AUTRES TYPES (RESP O,N,C)
C
C   NOUS AVONS UTILISE LA NUMEROTATION SUIVANTE
C    NTYPO = 1
C    NTYPN = 2
C    NTYPC = 3
C    NTYPH = 4
C
C
C
C   OAVS
C   -----
C   NOMBRE D'ORBITALES DE VALENCE DE TYPE S
                      F    N
C   1 ORBITALE S
C   L'ORBITALE 1S  POUR L'ATOME D'HYDROGENE
C L'ORBITALE 2S  POUR LES ATOMES  DE CARBONE  D'AZOTE ET D'OXYGENE
C
C    OAV2P
C   --------
C   NOMBRE D'ORBITALES DE VALENCE DE TYPE 2P
C    DANS  CE PROGRAMME  ON A PRIS  TROIS  ORBITALES  2P
C    POUR LES ATOMES DE CARBONE  D'AZOTE ET D'OXYGENE
C   OAVSP
C -----
C   NOMBRE D'ORBITALES ATOMIQUES CONSIDEREES DANS CE PROGRAMME
C    POUR LES ATOMES DE CARBONE  D'AZOTE ET D'OXYGENE
C      OAVSP=OAVS+OAV2P
C
C
C     TABLEAUX  OZ    ZN    CZ    HZ
C     ----------------
C      TABLEAUX  DES ENERGIES D'INTERACTIONS CALCULEES A PARTIR DES
C      INTEGRALES DE COULOMB ET D'ECHANGES ENTRE LES DIFFERENTES
C      ORBITALES ATOMIQUES D'UN ATOME DE TYPE DONNE
C      CES TABLEAUX  CONCERNENT RESPECTIVENENT LES ATOMES D'OXYGENE
C     D'AZOTE  DE CARBONE  ET D'HYDROGENE
```

```
C
C      TABLEAUX  OW    WN    CW    HW
C      ------------------------------
C       TABLEAUX  DES ENERGIES DE COEUR  DES ORBITALES ATOMIQUES D'UN TYPE
C      D'ATOME DONNE  (RESPECTIVEMENT  O,N,C,H )
C
C   ON APPELLE ' ENERGIE DE COEUR '  L'ENERGIE QUE POSSEDERAIT UN
C    ELECTRON DANS UNE ORBITALE ATOMIQUE DONNEE D'UN ATOME DONNEE
C    ET DANS L'ETAT DE VALENCE APPROPRIE
C
C
C      TABLEAU  ZD(32)
C      ---------------
C                        TABLEAU  DES EXPOSANTS ORBITAUX RELATIFS A UNE
C   BASE DANS LAQUELLE L'ORBITALE  2S  EST ORTHOGONALE A L'ORBITALE 1S
C  ET OU  LES ORBITALES 2P (OU 1S,2S,2P) SONT DEVOLLOPEES
C   EN  'DOUBLE DZETA '
C    CE TABLEAU  EST CONSTRUIT  DE LA MANIERE SUIVANTE
C
C   CHAQUE LIGNE  CORRESPOND A UN TYPE D'ATOME DIFFERENT
C    IL DOIT DONC AVOIR  NESP  LIGNES
C    CHAQUE LIGNE  COMPORTANT NDIMZD  VALEURS
C   NDIMZD  EST EGAL A  4   SI SEULE L'ORBITALE  2P EST DEVELLOPEE
C    EN DOUBLE DZETA
C    CHAQUE  LIGNE  DU TABLEAU   ZD  CONTIENT  ALORS LES VALEURS
C  DZETA 1S   DZETA 2S   DZETA 2P1   DZETA 2P2
C
C   NDIMZD  EST EGAL  A  6  SI LES ORBITALES  1S,2S,2P  SON DEVELLOPEES
C   EN DOUBLE DZETA
C    CHAQUE  LIGNE  DU TABLEAU   ZD  CONTIENT  ALORS LES VALEURS
C  DZETA 1S1   DZETA 1S2   DZETA 2S1   DZETA 2S2   DZETA 2P1   DZETA 2P2
C
C
C     COEFO  COEFN  COEFC  COEFH
C     ---------------------------
C                        TABLEAUX DES COEFFICIENTS RELATIFS
C   A LA DECOMPOSITION EN DOUBLE DZETA  DES ORBITALES ATOMIQUES
C   DE BASE (RESPECTIVEMENT POUR LES ATOMES O,N,C,H)
C
C    COEF   MATRICE (NESP,3,NDIMZD)  CONSTITUEE A L'AIDE DE
C     CES COEFFICIENTS
C
C
C     IDENT
C     -----
```

```
C           TABLEAU ALPHANUMERIQUE  SERVANT A IDENTIFIER LE PROBLEME
C
C
C                                COORDONNEES
C                                ===========
C     CE PROGRAMME UTILISE  LES COORDONNEES DES ATOMES DE LA MOLECULE
C      AINSI QUE LES COORDONNEES DES POINTS DEFINISSANT LES TRIEDRES
C      LOCAUX CENTRES SUR LES ATOMES POSSEDANT DES ORBITALES DE TYPE 2P
C      CHAQUE TRIEDRE LOCAL EST DEFINI PAR L'ATOME SUR LEQUEL IL EST
C      CENTRE ET PAR DEUX POINTS  P1 ET P2
C    ON APELLE
C      NATOM  LE NOMBRE D'ATOMES DE LA MOLECULE
C      -----
C      NTRI LE NOMBRE DE POINTS TOTAL NECESSAIRES POUR DEFINIR
C      -------
C      LES TRIEDRES CENTRES SUR LES ATOMES POSSEDANT DES ORBITALES 2P
C   ON NE DEFINIRA QUE DEUX AXES PAR TRIEDRE
C   LE TROISIEME AXE EST PRIS ORTHOGONAL AUX DEUX PREMIERS
C LE TRIEDRE  ETANT  DIRECT
C      NAT =NATOM + NTRI
C       ---
C
C      LE PROGRAMME  OFFRE DEUX POSSIBILITES
C    1)   NCOORD=1   NPROXY=0
C         ---------------------
C     SOIT  RENTRER DIRECTEMENT SUR CARTES PERFOREES LES COORDONNEES
C      DES ATOMES ET DES POINTS DEFINISSANT LES TRIEDRES,CES COORDONNEES
C      ETANT CALCULEES DANS UN TRIEDRE DE REFERENCE  TA
C    ON DOIT AVOIR  NAT CARTES PERFOREES  CONSTITUEES DE DEUX PAQUET
C         LES  NAT   PREMIERES CARTES
C         ---------------------------
C      CHAQUE CARTE COMPORTE LES COORDONNEES D'UN ATOME DE LA MOLECULE
C      PERFOREES EN 3 F20.
C      LA NUMEROTATION DE CES ATOMES EST DONNEE PAR LEUR PLACE DANS CE
C      PAQUET DE CARTES
C
C           LES  NTRI  CARTES SUIVANTES
C          ----------------------------
C      SUR CHAQUE CARTE  ON DOIT TROUVER
C      LES COORDONNEES DU POINT DEFINISSANT UN TRIEDRE
C      DEUX NOMBRES  ITRI  ET  IAXE  PERFORE EN COLONNE  77 78  ET 79 80
C      ITRI INDIQUE LE NUMERO DE L'ATOME SUR LEQUEL EST CENTRE LE TRIEDRE
C      IAXE EST EGAL A  1   OU   A 2  SUIVANT QUE LE POINT  DEFINIT
C      LE PREMIER AXE DU TRIEDRE  OU LE SECOND
C   EXEMPLE
```

```
C      1.0                0.0        0.0
C     INDIQUE QUE LE POINT DEFINIT L'AXE NUMERO 2 DU TRIEDRE LOCAL
C      CENTRE SUR L'ATOME NUMERO 5
C    2)   NCOORD=0   NPROXY=1
C         ~~~~~~~~~~~~~~~~~~~
C    ON PEUT FAIRE APPEL AU SOUS PROGROMME  PROXYZ  ECRIT  PAR    KUSNETS
C
C
C      NOR
C      ~~~
C    SI ON DESIRE FAIRE UNE TRANSLATION DE L'ORIGINE DU TRIEDRE DE REFER.
C    NOR= NUMERO DE L ATOME PRIS COMME NOUVELLE ORIGINE
C    SINON NOR=0
C
C
C      NM1  NM2
C      ~~~~~~~~
C      SI ON DESIRE EFFECTUER UNE ROTATION DU TRIEDRE DE REFERENCE
C    NM1 ET NM2  SONT LES NUMEROS DES POINTS DEFINISSANT LES DEUX NOUVEAUX
C    AXES  PAR RAPPORT A L ORIGINE
C      NM1  ET  NM2 DOIVENT ETRE PRIS DANS L'ENSEMBLE (1,2,...NAT)
C    AXE NUMERO 1 = AXE JOIGNANT L ORIGINE AU POINT NUMERO NM1
C    AXE NUMERO 2 = AXE JOIGNANT L ORIGINE AU POINT NUMERO NM2
C    SINON ON DOIT AVOIR NM1=0  NM2=0
C
C
C      NVNUM
C      ~~~~~
C      SI ON DESIRE INTRODUIRE UNE NUMEROTATION QUE CELLE QUI EST DONNEE
C     PAR LA SUCCESSION DES CARTES DONNEES  DONNER A NVNUM  UNE VALEUR
C    NON NULLE   RENTRER DANS LE TABLEAU NSORT(32) LA NOUVELLE
C     (1,2,    NAT)
C      SININ   NVNUM=0
C
C
C    NSYMET
C    ~~~~~~
C     NSYMET  DIFFEREND  DE ZERO   INDIQUE  QUE  L'ON  FAIT APPEL
C     (NSYMET) FOIS  AU SOUS-PROGRAMME  VSYMET
C     QUI VERIFIE SI UN COUPLE DE POINTS DONNE VERIFIE UNE SYMETRIE DONNEE
C     SINON  NSYMET =0
C
C
C      IUNIT
C      ~~~~~
```

```
C      CONTROLE D'UNITE
C      IUNIT=1  LES COORDONNEES SERONT ECRITES EN ANGSTROM
C      IUNIT=1  LES COORDONNEES SERONT ECRITES EN UNITE DE BOHR
C      IUNIT=2  LES COORDONNEES SERONT ECRITES DANS LES DEUX SYSTEMES
C
C
C      IPUNCH
C      ------
C      CONTOLE  DE PERFORATION DES COORDONNEES
C      IPUNCH =0 PAS DE PERFORATION DES COORDONNEES
C      IPUNCH = 1  PERFORATION DES COORDONNEES  DANS LE SYSTEME D'UNITE
C      DEFINI PAR  IUNIT
C
C
C  KSACK
C -------
C  KSACK=1  INDIQUE  QUE L'ON  UTILISE L'EXTRAPOLATION DE SACK
C   SINON  KSACK=0
C
C
C      NIT
C      ---
C      NOMBRE D'ITERATIONS DEMANDEES
C    POUR  L'UTILISATION  DE LA METHODE D'ESSAI-ERREUR SUR LES INDICES
C   DE POPULATION AVEC LESQUELS  ON CONSTRUIT L'HAMILTONIEN EFFECTIF
C
C
C      TKO
C      ---
C        PARAMETRE SEMI-EMPIRIQUE  UTILISE POUR LA FORMULE
C    DE WOLFSBERG-HELMHOLZ (PARAMETRE K)
C
C
C      ALFAO   SO
C      ----------
C                        DEUX PARAMETRES INTERVENANT DANS LA TRANSLATION DU
C    ZERO D'ENERGIE
C    ALFAO  CORRESPOND APPROXIVEMENT A L'ENERGIE D'IONISATION  D'UN
C    ELECTRON  'SIGMA'  APPARTENANT A UN CARBONE TRIGONAL
C
C  DANS CETTE VERSION ON A PRIS  ALFAO=-8.64    TKO=3     SO=0.3
C
C
C  TABLEAU  Q(32)
C ----------
```

```
C  TABLEAU  CONTENANT  LES  INDICES DE POPULATION  DE DEPART
C
C
C      NELEC
C      -----
C                    NOMBRE D'ELECTRONS A PLACER SUR LES
C ORBITALES MOLECULAIRES
C
C
C      NDO
C      ---
C        NOMBRE D'ORBITALES MOLECULAIRES DOUBLEMENT OCCUPEES
C
C
C      NSO
C      ---
C        NOMBRE D'ORBITALES MOLECULAIRES SIMPLEMENT OCCUPEES
C    RESTRICTION  DE CE PROGRAMME
C   ----------------------------
C    NSO  NE PEUT ETRE EGAL QU'A  0  OU  1
C ------------------------------------------------------------------------------
C
      READ(LEC,7100)IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      READ(LEC,7100)NESP
      READ(LEC,7100)NTYPO,NTYPN,NTYPC,NTYPH
      READ(LEC,7100)OAVS,OAV2P
      OAVSP=OAVS+OAV2P
 8003 CONTINUE
      DO 7000 I=1,OAVSP
      READ(LEC,5510)(OZ(I,J),J=1,OAVSP)
 7000 CONTINUE
      DO 7001 I=1,OAVSP
      READ(LEC,5510)(ZN(I,J),J=1,OAVSP)
 7001 CONTINUE
      DO 7002 I=1,OAVSP
      READ(LEC,5510)(CZ(I,J),J=1,OAVSP)
 7002 CONTINUE
      READ(LEC,5510)HZ
      READ(LEC,5510)OW
      READ(LEC,5510)WN
      READ(LEC,5510)CW
      READ(LEC,5510)HW
      READ(LEC,7100)NDIMZD
      DO 4081 I=1,NESP
      READ(LEC,4082)(ZD(I,J),J=1,NDIMZD)
```

```
      ZD(I,NDIMZD+1)=ZD(I,NDIMZD-1)
      ZD(I,NDIMZD+2)=ZD(I,NDIMZD)
4081  CONTINUE
      NDIMZD=NDIMZD+2
      DO 7003 I1=1,NESP
      DO 7003 J1=1,3
      READ(LEC,5512)(COEF(I1,J1,K1),K1=1,NDIMZD)
7003  CONTINUE
      WRITE(JDISC)NDIMZD
      WRITE(JDISC)OZ,ZN,CZ,HZ
      WRITE(JDISC)OW,WN,CW,HW
      WRITE(JDISC)COEF
    1 CONTINUE
      READ(LEC,100)IDENT
      READ(LEC,7783)NAT,NATOM,NTRI,NCOORD,NPROXY,NOR,NM1,NM2,NVNUM,
     1NSYMET,IUNIT,IPUNCH
1000  WRITE(IMP,101)IDENT
      WRITE(IMP,7777)
      WRITE(IMP,7775)NAT,NATOM,NTRI,NCOORD,NPROXY,NOR,NM1,NM2,NVNUM
     1,NSYMET,IUNIT,IPUNCH
      MA=NAT-NTRI
      IF(NPROXY)8252,8252,8253
8253  CONTINUE
      WRITE(IMP,7776)
      WRITE(IMP,7778)
      IF(NOR)90,91,90
   91 KOR=1
      WRITE(IMP,7779)KOR
      GO TO 94
   90 CONTINUE
      WRITE(IMP,7779)NOR
   94 CONTINUE
      NM12=NM1+NM2
      IF(NM12)92,93,92
   93 KNM1=2
      KNM2=3
      WRITE(IMP,7780)KOR,KNM1
      WRITE(IMP,7781)KOR,KNM2
      GO TO 95
   92 CONTINUE
      WRITE(IMP,7780)NOR,NM1
      WRITE(IMP,7781)NOR,NM2
   95 CONTINUE
      WRITE(IMP,7782)
      DO 3  NA=1,NAT
```

```
    3 READ(LEC,102) NA,ATNO(NA),BOD(NA),BOA(NA),(ID(NA,J),J=1,4)
     1,PAN(NA),TAU(NA),ITRI(NA),IAXE(NA)
C    ------------------------------------------------------------------
C                COMMENTAIRE
C                -----------
C NA EST PERFORE EN COLONNE 1ET2
C ATNO(NA) EN COLONNES 6 A 10
C BOD(NA) EN COLONNES  11 A20
C  BOA(NA) EN COLONNES 21 A 35
CID(NA,J) EN COLONNES 39 40,41 42,43 44,45 46
C PAN(NA) EN COLONNES 51 A60
C TAU(NA)  EN COLONNES 61 A 70
C ITRI(NA) EN COLONNES 77 78
C IAXE(NA) EN COLONNES 79 80
C    ------------------------------------------------------------------
 2000 WRITE(IMP,103)
      DO 4  NA=1,NAT
    4 WRITE(IMP,104) NA,ATNO(NA),BOD(NA),BOA(NA),(ID(NA,J),J=1,4)
     1,PAN(NA),TAU(NA),ITRI(NA),IAXE(NA)
      CALL PROXYZ
       GO TO 6683
 8252 CONTINUE
C    ------------------------------------------------------------------
C                COMMENTAIRE
C                -----------
C    X(I)   Y(I)   Z(I)   SONT PERFORES EN 3F20.14
C    POUR  I = 1 A NAT
C    ------------------------------------------------------------------
       DO 8254 IVVV=1,NAT
      READ(LEC,7787)X(IVVV),Y(IVVV),Z(IVVV),ITRI(IVVV),IAXE(IVVV)
      WRITE(IMP,8000)X(IVVV),Y(IVVV),Z(IVVV),ITRI(IVVV),IAXE(IVVV)
 8254 CONTINUE
 6683 CONTINUE
      READ(LEC,4083)(NE(I),I=1,NATOM)
      WRITE(IMP,4083)MA
      WRITE(IMP,4083)(NE(I),I=1,NATOM)
      IF(NVNUM)6685,6685,6684
 6684 CONTINUE
C    ------------------------------------------------------------------
C                COMMENTAIRE
C                -----------
C  NSORT(I)   TABLEAU  CONTENENT LA NOUVELLE NUMEROTATION DEFINIE
C   SUR LES ATOMES ET LES AXES DES TRIEDRES LOCAUX
C    ------------------------------------------------------------------
      DO 4005 I=1,NAT
```

```
4005 NENT(I)=I
     READ(LEC,7783)(NSORT(I),I=1,NAT)
6685 CONTINUE
     NDIM=0
     DO 7200 I=1,MA
     IF(NE(I)-NTYPH)7202,7201,7202
7201 NDIM=NDIM+OAVS
     GO TO 7200
7202 NDIM=NDIM+OAVSP
7200 CONTINUE
     WRITE(IMP,4083)NDIM
     READ(LEC,7100)KSACK
     WRITE(IMP,4083)KSACK
     READ(LEC,7100)NIT
     WRITE(IMP,4083)NIT
     READ(LEC,5512)TKO,SO,ALFAO
     READ(LEC,5512)(Q(I),I=1,NDIM)
     WRITE(IMP,5512)(Q(I),I=1,NDIM)
     READ(LEC,7100)NELEC,NDO,NSO
     WRITE(IMP,7100)NELEC,NDO,NSO
     WRITE(KDISC)LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
     WRITE(KDISC)NESP
     WRITE(KDISC)NTYPO,NTYPN,NTYPC,NTYPH
     WRITE(KDISC)OAVSP,OAVS
     WRITE(KDISC)NDIM
     WRITE(KDISC)KSACK
     WRITE(KDISC)NIT
     WRITE(KDISC)TKO,SO,ALFAO
     WRITE(KDISC)(Q(I),I=1,NDIM)
     WRITE(KDISC)NELEC,NDO,NSO
8002 CONTINUE
     RETURN
     END
```

```
// EXEC FORTRAN
      SUBROUTINE PROXYZ
      DIMENSION  PS(3,3),PSM(3,3)
      DIMENSION IDENT(60),ID(40,4),X(40),Y(40),Z(40),ATNO(40),BOD(40),
     1BOA(40),PAN(40),TAU(40),ALP(40),BET(40),GAM(40),R(40,40)
      DIMENSION ZD(4,8)
      DIMENSION NE(40)
      DIMENSION NENT(40),NSORT(40),ITRI(40),IAXE(40)
      DOUBLE PRECISION ZD,R,ATNO,BOD,BOA,PAN,TAU,X,Y,Z
      DOUBLE PRECISION RAD,PI,PI2,T30,T90,T270,ZETA,TAUP,SISI,COCO
      DOUBLE PRECISION ROP,RO,TAU9,CTAU9,CBOA,SPACT,RHO,POPOF,ETA,CSI
      DOUBLE PRECISION BOC,ETAP,CTAU,A,B,C,D,E,F,G,BC,X11,Y11,Z11,SOM
      DOUBLE PRECISION PXY,PYZ,PZX,XS,YS,ZS,P,Q,V,W,T,S
      DOUBLE PRECISION PIS6
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON ATNO,BOD,BOA,PAN,TAU,X,Y,Z
      COMMON NAT,NATOM,NTRI,NM1,NM2,NOR,NPROXY,NCOORD,NSYMET
      COMMON IDENT,IUNIT,IPUNCH,ID,ITRI,IAXE
      COMMON NENT,NSORT
      COMMON NVNUM
 6668 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 310' )
 6669 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 311' )
 6670 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 312' )
 6671 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 313' )
 6672 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 314' )
 6673 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 315' )
 6674 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 316' )
 6675 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 317' )
 6676 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 318' )
 6677 FORMAT(2X,'ATOME NUMERO',I3 ,3X ,'TRAITEMENT PAR 319' )
C     ------------------------------------------------------------------
C       TOUT LES ANGLES SONT CONVERTIS EN RADIANS
C     ------------------------------------------------------------------
      RAD=0.017453292519943
      PI=RAD*180.
      PI2=RAD*360.
      PIS6=PI/6.
      T30=RAD*30.
      T90=RAD*90.
      T270=RAD*270.
      DO 9  NA=1,NAT
      BOA(NA)=RAD*BOA(NA)
```

```
      TAU(NA)=RAD*TAU(NA)
      PAN(NA)=RAD*PAN(NA)
    9 CONTINUE
C     ALB,BET,GAM SONT LES COSINUS DIRECTEURS DES ATOMES
C     OBTENTION DES COORDONNEES DES ATOMES 1,2,3
      ALP(1)=0.
      BET(1)=0.
      GAM(1)=-1.
      ALP(2)=-DCOS(PIS6)
      BET(2)=0.
      GAM(2)=-0.5
      ALP(3)=0.
      BET(3)=0.
      GAM(3)=-1.
      X(1)=0.
      Y(1)=0.
      Z(1)=0.
      X(2)=0.
      Y(2)=0.
      Z(2)=BOD(2)
      GAM(4)=DCOS(BOA(3))
      BET(4)=0.
      IF(BOA(3)-T90)6,2,2
    6 ALP(4)=DCOS(T90-BOA(3))
      GO TO 500
    2 ALP(4)=DCOS(BOA(3)-T90)
  500 X(3)=BOD(3)*ALP(4)
      Y(3)=0.
      Z(3)=BOD(3)*GAM(4)
      DO 8  NA=4,NAT
      K=ID(NA,1)
      L=ID(NA,2)
      M=ID(NA,3)
      N=ID(NA,4)
      IF(K-3)461,460,461
  460 IF(L-4)461,462,461
  462 GAM(3)=1.
      ALP(4)=-ALP(4)
      GAM(4)=-GAM(4)
  461 IF(L-40)464,463,464
  463 GAM(L)=1.
      ALP(L)=0.
      BET(L)=0.
C     ------------------------------------------------------------------
C     CAS OU L'ANGLE  BOA(NA) VAUT  180  DEGRES
```

```
C     ------------------------------------------------------------------
  464 IF(BOA(NA)-PI)123,124,123
  124 ALP(M)=ALP(K)
      BET(M)=BET(K)
      GAM(M)=GAM(K)
      GO TO 11
  123 IF(BOA(NA)-T90)400,400,401
  400 ZETA=BOA(NA)
      GO TO 402
  401 ZETA=PI-BOA(NA)
  402 IF(TAU(NA)-T90)404,404,403
  403 IF(TAU(NA)-PI)405,405,408
  408 IF(TAU(NA)-T270)406,406,407
C     ------------------------------------------------------------------
C       TAUP EST L'ANGLE DE LA PROJECTION DE M DANS LE PLAN PERPENDICULAIRE
C     AU VECTEUR K      AVEC LE PLAN(K,L)
C     ------------------------------------------------------------------
  404 TAUP=TAU(NA)
      GO TO 409
  405 TAUP=PI-TAU(NA)
      GO TO 409
  406 TAUP=TAU(NA)-PI
      GO TO 409
  407 TAUP=PI2-TAU(NA)
C       ROP EST L'ANGLE QUE FAIT M AVE LE PLAN(K-L)
  409 SISI=DSIN(ZETA)*DSIN(TAUP)
      COCO=DSQRT(1.-SISI**2)
      ROP=DATAN(SISI/COCO)
      IF(TAU(NA)-T90)421,421,420
  420 IF(TAU(NA)-PI)422,422,425
  425 IF(TAU(NA)-T270)423,423,424
  421 RO=ROP
      GO TO 427
  422 RO=PI-ROP
      GO TO 427
  423 RO=ROP+PI
      GO TO 427
  424 RO=PI2-ROP
C     ------------------------------------------------------------------
C       TAU9 EST L'ANGLE DE M AVEC LE PRODUIT VECTORIEN K.L
C     ------------------------------------------------------------------
  427 TAU9=T90-RO
      CTAU=DCOS(ROP)
      CTAU9=DCOS(TAU9)
      CBOA=DCOS(BOA(NA))
```

```
      SPACT=CTAU9*(DSIN(PAN(NA)))
C     ------------------------------------------------------------------
C       OBTENTION DE L'ANGLE ENTRE K ET LA PROJECTION DE M SUR LE PLAN (K-
C       PAR SON COSINUS
C     ------------------------------------------------------------------
      IF(TAU(NA)-T90)130,131,132
  131 RHO=0.0
      GO TO 199
  133 RHO=BOA(NA)
      GO TO 140
  130 IF(TAU(NA))127,136,127
  136 IF(BOA(NA)-T90)133,133,137
  137 RHO=PI-BOA(NA)
      GO TO 140
  132 IF(TAU(NA)-PI)126,136,126
  126 IF(TAU(NA)-T270)127,131,127
  127 COCO=DCOS(ZETA)/DCOS(ROP)
      SISI=DSQRT(1.-COCO**2)
      POPOF=DATAN(SISI/COCO)
      IF(POPOF)8001,8000,8000
 8001 RHO=POPOF+PI
      GO TO 8002
 8000 RHO=POPOF
 8002 CONTINUE
  140 IF(PAN(NA)-T90)70,71,72
   70 IF(RHO-T90)73,201,73
   71 IF(RHO-T90)202,200,202
   72 IF(RHO-T90)74,202,74
   73 IF(RO-ROP)776,75,776
  776 IF(RO-(PI2-ROP))76,75,76
   74 IF(RO-ROP)778,77,778
  778 IF(RO-(PI2-ROP))73,77,78
   75 IF(BOA(NA)-T90)203,203,79
   76 IF(BOA(NA)-T90)79,203,203
   77 IF(BOA(NA)-T90)80,202,202
   78 IF(BOA(NA)-T90)202,202,80
   79 IF(PAN(NA)-RHO)201,202,202
   80 IF((PAN(NA)+RHO)-PI)203,204,204
C       ETA EST L'ANGLE DE L AVEC LA PROJECTION DE M DANS LE PLAN(K-L)
  200 ETA=0.
      GO TO 207
  201 ETA=RHO-PAN(NA)
      GO TO 207
  199 IF(BOA(NA)-T90)601,602,601
  601 IF(PAN(NA)-T90)202,602,203
```

```
C      CSI EST LE COSINUS DE L'ANGLE DES VECTEURS L ET M
  602 CSI=0.
      GO TO 620
  202 ETA=PAN(NA)-RHO
      GO TO 604
  203 ETA=PI-(PAN(NA)+RHO)
  604 IF(RHO)205,605,205
  605 IF(BOA(NA)-T90)210,211,211
  210 BOC=BOA(NA)
      GO TO 212
  211 BOC=PI-BOA(NA)
  212 CSI=-(DCOS(BOC))*DCOS(ETA)
      GO TO 209
  204 ETA=(PAN(NA)+RHO)-PI
  205 IF(ETA-T90)207,207,206
  206 ETAP=PI-ETA
      GO TO 208
  207 ETAP=ETA
  208 CSI=-CTAU*DCOS(ETAP)
  209 IF(TAU(NA)-T90)621,611,612
  612 IF(TAU(NA)-T270)622,611,621
  621 IF(ETA-T90)611,611,608
  622 IF(ETA-T90)608,611,611
  611 IF(RHO)620,609,620
  609 IF(BOA(NA)-T90)607,606,606
  606 IF(PAN(NA)-T90)620,608,608
  607 IF(PAN(NA)-T90)608,608,620
  608 CSI=-CSI
C     ------------------------------------------------------------------
C      COSINUS DIRECTEURS DE M
C     ------------------------------------------------------------------
  620 IF(DABS(ALP(K))-3.D-7)81,81,82
   81 IF(DABS(BET(K))-3.D-7)83,83,84
   82 IF(DABS(GAM(L))-3.D-7)85,85,828
  828 IF(DABS(BET(K))-3.D-7)89,89,310
   83 IF(DABS(BET(L))-3.D-7)86,86,311
   84 IF(DABS(GAM(L))-3.D-7)87,87,312
   85 IF(DABS(BET(K))-3.D-7)89,89,88
   86 IF(DABS(GAM(L))-3.D-7)313,313,314
   87 IF(DABS(BET(L))-3.D-7)316,316,90
   88 IF(DABS(GAM(K))-3.D-7)318,318,310
   89 IF(DABS(ALP(L))-3.D-7)898,898,310
  898 IF(DABS(BET(L))-3.D-7)319,319,317
   90 IF(DABS(GAM(K))-3.D-7)315,315,312
  310 A=BET(K)*GAM(L)-GAM(K)*BET(L)
```

```
      B=GAM(K)*ALP(L)-ALP(K)*GAM(L)
      C=ALP(K)*BET(L)-BET(K)*ALP(L)
      S=B*ALP(K)-A*BET(K)
      P=(-SPACT*ALP(K)+A*CBOA)/S
      Q=(A*GAM(K)-C*ALP(K))/S
      V=(-CBOA-P*BET(K))/ALP(K)
      W=(Q*BET(K)+GAM(K))/ALP(K)
      T=CSI-V*ALP(L)-P*BET(L)
      D=Q*BET(L)-W*ALP(L)+GAM(L)
  331 GAM(M)=T/D
      ALP(M)=V-W*GAM(M)
      BET(M)=P+Q*GAM(M)
      WRITE(IMP,6668)NA
      GO TO 11
  311 BET(M)=(BET(L)*(GAM(K)*CSI+GAM(L)*CBOA)-GAM(K)*ALP(L)*SPACT)/(GAM(
     1K)*(ALP(L)**2+BET(L)**2))
      ALP(M)=(SPACT+ALP(L)*BET(M)*GAM(K))/(GAM(K)*BET(L))
      GAM(M)=-CBOA/GAM(K)
      WRITE(IMP,6669)NA
      GO TO 11
  312 D=BET(K)*GAM(L)-GAM(K)*BET(L)
      E=(-SPACT*BET(K)+CBOA*GAM(K)*ALP(L))/(D*BET(K))
      F=BET(K)*(E*ALP(L)-CSI)-CBOA*BET(L)
      G=(ALP(L)**2+D**2)/D
      GAM(M)=(-F)/G
      ALP(M)=E +ALP(L)*GAM(M)/(D*BET(K))
      BET(M)=(-CBOA-GAM(K)*GAM(M))/BET(K)
      WRITE(IMP,6670)NA
      GO TO 11
  313 GAM(M)=(-CBOA)/GAM(K)
      BET(M)=(-SPACT)/(GAM(K)*ALP(L))
      ALP(M)=CSI/ALP(L)
      WRITE(IMP,6671)NA
      GO TO 11
  314 GAM(M)=-CBOA/GAM(K)
      BET(M)=-SPACT/(ALP(L)*GAM(K))
  325 ALP(M)=(CSI+CBOA*GAM(L)/GAM(K))/ALP(L)
      WRITE(IMP,6672)NA
      GO TO 11
  315 BET(M)=(-CBOA)/BET(K)
      GAM(M)=SPACT/(ALP(L)*BET(K))
  335 ALP(M)=(BET(K)*CSI+CBOA*BET(L))/(ALP(L)*BET(K))
      WRITE(IMP,6673)NA
      GO TO 11
  316 GAM(M)=(-CBOA*GAM(K)*ALP(L)+BET(K)*SPACT)/ALP(L)
```

```
      ALP(M)=CSI/ALP(L)
      BET(M)=(-CBOA-GAM(M)*GAM(K))/BET(K)
      WRITE(IMP,6674)NA
      GO TO 11
  317 GAM(M)=ALP(K)*SPACT-GAM(K)*CBOA
      BET(M)=CSI/BET(L)
      ALP(M)=(-CBOA-GAM(M)*GAM(K))/ALP(K)
      WRITE(IMP,6675)NA
      GO TO 11
  318 BC=ALP(K)*BET(L)-BET(K)*ALP(L)
      GAM(M)=SPACT/BC
      BET(M)=(ALP(K)*CSI+CBOA*ALP(L))/BC
      ALP(M)=(-CBOA-BET(K)*BET(M))/ALP(K)
      WRITE(IMP,6676)NA
      GO TO 11
  319 GAM(M)=CSI/GAM(L)
      BET(M)=SPACT/(ALP(K)*GAM(L))
  345 ALP(M)=(-CBOA-GAM(K)*GAM(M))/ALP(K)
      WRITE(IMP,6677)NA
   11 IF(K-3)581,580,581
  580 IF(L-4)581,582,581
  582 GAM(3)=-1.
      ALP(4)=-ALP(4)
      GAM(4)=-GAM(4)
  581 IF(L-40)584,583,584
  583 GAM(L)=-1.
  584 X(NA)=X(N)+BOD(NA)*ALP(M)
      Y(NA)=Y(N)+BOD(NA)*BET(M)
    8 Z(NA)=Z(N)+BOD(NA)*GAM(M)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE TRANS(X,Y,Z,NAT,NOR)
      DIMENSION X(40),Y(40),Z(40)
      DOUBLE PRECISION X,Y,Z,X11,Y11,Z11
      DO 900 IV=1,NAT
      X11=X(NOR)
      Y11=Y(NOR)
      Z11=Z(NOR)
      X(IV)=X(IV)-X11
      Y(IV)=Y(IV)-Y11
  900 Z(IV)=Z(IV)-Z11
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE ROTAX(X,Y,Z,PXY,PYZ,PZX,NAT,NM1,NM2)
      DIMENSION X(40),Y(40),Z(40),PS(3,3),PSM(3,3)
      DOUBLE PRECISION X,Y,Z,PS,PSM,SOM,PXY,PYZ,PZX,XS,YS,ZS
      PS(1,1)=X(NM1)
      PS(2,1)=Y(NM1)
      PS(3,1)=Z(NM1)
      PS(1,2)=X(NM2)
      PS(2,2)=Y(NM2)
      PS(3,2)=Z(NM2)
      PS(1,3)=Y(NM1)*Z(NM2)-Z(NM1)*Y(NM2)
      PS(2,3)=Z(NM1)*X(NM2)-X(NM1)*Z(NM2)
      PS(3,3)=X(NM1)*Y(NM2)-Y(NM1)*X(NM2)
      DO 901    IV=1,3
      SOM=0.0
      DO 902 JV=1,3
  902 SOM=SOM+PS(JV,IV)*PS(JV,IV)
      DO 901 KV=1,3
  901 PS(KV,IV)=PS(KV,IV)/DSQRT(SOM)
      PXY=0.
      PYZ=0.
      PZX=0.
      DO 903 IV=1,3
      PXY=PXY+PS(IV,1)*PS(IV,2)
      PYZ=PYZ+PS(IV,2)*PS(IV,3)
  903 PZX=PZX+PS(IV,3)*PS(IV,1)
      CALL DMATIN(PS,PSM,3)
      DO 904  IV=1,NAT
      XS=X(IV)
      YS=Y(IV)
      ZS=Z(IV)
      X(IV)=PSM(1,1)*XS+PSM(1,2)*YS+PSM(1,3)*ZS
      Y(IV)=PSM(2,1)*XS+PSM(2,2)*YS+PSM(2,3)*ZS
  904 Z(IV)=PSM(3,1)*XS+PSM(3,2)*YS+PSM(3,3)*ZS
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE DIST(X,Y,Z,R,NAT)
      DIMENSION X(40),Y(40),Z(40),R(40,40)
      DOUBLE PRECISION X,Y,Z,R
      DOUBLE PRECISION PIGN
      R(1,1)=0.0
      DO 1 I=2,NAT
      R(I,I)=0.0
      IM=I-1
      DO 1 J=1,IM
      PIGN=DSQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2+(Z(I)-Z(J))**2)
      R(I,J)=PIGN
      R(J,I)=R(I,J)
    1 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE PERMUT(NSORT,ITRI,IAXE,X,Y,Z,R,ID,BOD,NE,NPRO,NAT,NATO)
      DIMENSION NSORT(40),ITRI(40),IAXE(40),X(40),Y(40),Z(40),R(40,40)
      DIMENSION JTRI(40),JAXE(40),X1(40),Y1(40),Z1(40),R1(40,40)
      DIMENSION ID(40,4),BOD(40),ID1(40,4),BOD1(40),NE(40),NE1(40)
      DOUBLE PRECISION X,Y,Z,X1,Y1,Z1,R,R1,BOD,BOD1
      DO 1 I=1,NAT
      K=ITRI(I)
      IF(K)2,3,2
    2 CONTINUE
      JTRI(I)=NSORT(K)
      GO TO 1
    3 JTRI(I)=0
    1 CONTINUE
      DO 4000 I=1,NAT
      K=NSORT(I)
      JAXE(K)=IAXE(I)
      X1(K)=X(I)
      Y1(K)=Y(I)
      Z1(K)=Z(I)
      DO 4000 J=1,NAT
      R1(K,J)=R(I,J)
 4000 CONTINUE
      DO 4001 I=1,NATO
      K=NSORT(I)
      NE1(K)=NE(I)
 4001 CONTINUE
      DO 5000 I=1,NAT
      X(I)=X1(I)
      Y(I)=Y1(I)
      Z(I)=Z1(I)
      ITRI(I)=JTRI(I)
      IAXE(I)=JAXE(I)
      K=NSORT(I)
      DO 5000 J=1,NAT
      R(J,K)=R1(J,I)
 5000 CONTINUE
      DO 5001 I=1,NATO
      NE(I)=NE1(I)
 5001 CONTINUE
      IF(NPRO)7000,7000,6000
 6000 CONTINUE
      ID(1,4)=1
      ID(2,4)=1
      ID(3,4)=1
      DO 8000 I=1,NAT
```

```
      K=NSORT(I)
      BOD1(K)=BOD(I)
      ID1(K,4)=ID(I,4)
8000  CONTINUE
      DO 9000 I=1,NAT
      ID(I,4)=ID1(I,4)
      BOD(I)=BOD1(I)
9000  CONTINUE
7000  CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE ECRICD(X,Y,Z,R,IDENT,NAT,IUNIT,IMP,IPERF,IPUNCH)
      DIMENSION X(40),Y(40),Z(40),XB(40),YB(40),ZB(40),IDENT(60)
      DIMENSION R(40,40)
      DOUBLE PRECISION X,Y,Z,XB,YB,ZB
      DOUBLE PRECISION R
  105 FORMAT('OMATRICE DES DISTANCES INTER-ATOMIQUES DANS LA NUMEROTATIO
     1N DEMANDEE')
  106 FORMAT(/2X,11(1X,F7.4))
  107 FORMAT(//2X,'XYZ EN ANGSTROMS',//1X,'NA',7X,'X',11X,'Y',11X,'Z')
  108 FORMAT(/1X,I2,3(2X,F11.8))
  109 FORMAT(//2X,'XYZ EN BOHRS',//1X,'NA',7X,'X',11X,'Y',11X,'Z')
 1030 FORMAT('COORDONNEES EN UNITE ANGSTROMS')
 1031 FORMAT('COORDONNEES EN UNITE DE BOHRS')
 1107 FORMAT(60A1)
 1108 FORMAT(I2,8X,3F11.8)
 6681 FORMAT(1H1)
C     ----------------------------------------------------------------
C       CONTROLE D'UNITE
C     ----------------------------------------------------------------
      IF(IUNIT-1)20,21,22
   20 WRITE(IMP,107)
      DO 14  NA=1,NAT
   14 WRITE(IMP,108)NA,X(NA),Y(NA),Z(NA)
      WRITE(IMP,6681)
      GO TO 18
   21 WRITE(IMP,109)
      DO 15  NA=1,NAT
      XB(NA)=1.8896*X(NA)
      YB(NA)=1.8896*Y(NA)
      ZB(NA)=1.8896*Z(NA)
   15 WRITE(IMP,108)NA,XB(NA),YB(NA),ZB(NA)
      WRITE(IMP,6681)
      GO TO 18
   22 WRITE(IMP,107)
      DO 16  NA=1,NAT
   16 WRITE(IMP,108)NA,X(NA),Y(NA),Z(NA)
      WRITE(IMP,109)
      DO 17  NA=1,NAT
      XB(NA)=1.8896*X(NA)
      YB(NA)=1.8896*Y(NA)
      ZB(NA)=1.8896*Z(NA)
   17 WRITE(IMP,108)NA,XB(NA),YB(NA),ZB(NA)
      WRITE(IMP,6681)
   18 IF(IPUNCH)1018,1018,30
   30 CONTINUE
```

```
C CONTROLE D UNITE
      IF(IUNIT-1)1020,1021,1022
 1020 WRITE(IPERF,1107)IDENT
      WRITE(IPERF,1030)
      DO 1014 NA=1,NAT
 1014 WRITE(IPERF ,1108)NA,X(NA),Y(NA),Z(NA)
      GO TO 1018
 1021 WRITE(IPERF ,1107)IDENT
      WRITE(IPERF ,1031)
      DO 1015 NA=1,NAT
      XB(NA)=1.8896*X(NA)
      YB(NA)=1.8896*Y(NA)
      ZB(NA)=1.8896*Z(NA)
 1015 WRITE(IPERF ,1108)XBNA,XB(NA),YB(NA),ZB(NA)
      GO TO 1018
 1022 WRITE(IPERF ,1107)IDENT
      WRITE(IPERF ,1030)
      DO 1016 NA=1,NAT
 1016 WRITE(IPERF ,1108)NA,X(NA),Y(NA),Z(NA)
      WRITE(IPERF ,1031)
      DO 1017 NA=1,NAT
      XB(NA)=1.8896*X(NA)
      YB(NA)=1.8896*Y(NA)
      ZB(NA)=1.8896*Z(NA)
 1017 WRITE(IPERF ,1108)NA,XB(NA),YB(NA),ZB(NA)
 1018 CONTINUE
      WRITE(IMP,105)
      DO 13 J=1,NAT
   13 WRITE(IMP,106)(R(I,J),I=1,J)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE PREC48
      DIMENSION NE(40),ZD(4,8),R(40,40)
      DOUBLE PRECISION ZD,R
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
 1001 FORMAT(8F10.6)
 1002 FORMAT(I5)
 1003 FORMAT(16I5)
 1004 FORMAT(I3,2I4)
 1005 FORMAT(I2,I4)
 1006 FORMAT(I2)
 1007 FORMAT(I2)
 1008 FORMAT(I4)
 1009 FORMAT('ONS= ',I5)
 1010 FORMAT(1H0,I3,2I4)
C     ------------------------------------------------------------------
C                  COMMENTAIRE
C                  -----------
C    LIM1  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-1S
C    LIM2  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-2S
C    LIM3  NOMBRE D'INTEGRALES DE RECOUVREMENTS  2S-2S
C    LIM4  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-SIGMA
C    LIM5  NOMBRE D'INTEGRALES DE RECOUVREMENTS  2S-SIGMA
C    LIM6  NOMBRE D'INTEGRALES DE RECOUVREMENTS  SIGMA-SIGMA
C    LIM7  NOMBRE D'INTEGRALES DE RECOUVREMENTS  PI-PI
C  NH  NOMBRE D'ATOMES D'HYDROGENE
C    NONC   NOMBRE D'ATOMES D'OXYGENE   AZOTE   CARBONE
C     ------------------------------------------------------------------
      NA=MA
      I1=100
      I2=200
      I3=211
      I4=210
      NH=0
      DO 3 I=1,NA
      IF(NE(I)-NTYPH)3,4,3
    4 NH=NH+OAVS
    3 CONTINUE
      NONC=NA-NH
      LIM1=NA*(NA-1)+NONC*(NONC-1)-(NH*(NH-1))/2
      LIM2=3*NONC*(NONC-1)+NA*(NA-1)-NH*(NH-1)
      LIM3=2*NONC*(NONC-1)
```

```
      LIM4=LIM2
      LIM5=2*LIM3
      LIM6=LIM3
      LIM7=LIM6
       NS=LIM1+LIM2+LIM3+LIM4+LIM5+LIM6+LIM7
      WRITE(IDISC)NS
      WRITE(IDISC)LIM1,I1,I1
      DO 10 LA=1,2
      DO 10 LB=1,2
   10 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM2,I1,I2
      DO 11 LA=1,2
      DO 11 LB=3,4
   11 CALL CLASS(LA,LB)
      DO 12 LB=1,2
      DO 12 LA=3,4
   12 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM3,I2,I2
      DO 13 LA=3,4
      DO 13 LB=3,4
   13 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM4,I1,I3
      DO 14 LA=1,2
      DO 14 LB=5,6
   14 CALL CLASS(LA,LB)
      DO 15 LB=1,2
      DO 15 LA=5,6
   15 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM5,I2,I3
      DO 16 LA=3,4
      DO 16 LB=5,6
   16 CALL CLASS(LA,LB)
      DO 17 LB=3,4
      DO 17 LA=5,6
   17 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM6,I3,I3
      DO 18 LA=5,6
      DO 18 LB=5,6
   18 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM7,I4,I4
      DO 19 LA=7,8
      DO 19 LB=7,8
   19 CALL CLASS(LA,LB)
      WRITE(IDISC)NA
      WRITE(IDISC)(NE(I),I=1,NA)
```

```
RETURN
END
```

```
// EXEC FORTRAN
      SUBROUTINE PREC46
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DOUBLE PRECISION ZD,R
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
 1001 FORMAT(6F10.6)
 1002 FORMAT(I5)
 1003 FORMAT(16I5)
 1005 FORMAT(I2,I4)
 1004 FORMAT(I3,2I4)
 1006 FORMAT(I2)
 1007 FORMAT(I2)
 1008 FORMAT(I4)
 1009 FORMAT('ONS= ',I5)
 1010 FORMAT(1HO,I3,2I4)
C                COMMENTAIRE
C                -----------
C    LIM1  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-1S
C    LIM2  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-2S
C    LIM3  NOMBRE D'INTEGRALES DE RECOUVREMENTS  2S-2S
C    LIM4  NOMBRE D'INTEGRALES DE RECOUVREMENTS  1S-SIGMA
C    LIM5  NOMBRE D'INTEGRALES DE RECOUVREMENTS  2S-SIGMA
C    LIM6  NOMBRE D'INTEGRALES DE RECOUVREMENTS  SIGMA-SIGMA
C    LIM7  NOMBRE D'INTEGRALES DE RECOUVREMENTS  PI-PI
C  NH  NOMBRE D'ATOMES D'HYDROGENE
C     NONC   NOMBRE D'ATOMES D'OXYGENE   AZOTE   CARBONE
C    ------------------------------------------------------------------------
      NA=MA
      I1=100
      I2=200
      I3=211
      I4=210
      NH=0
      DO 3 I=1,NA
      IF(NE(I)-NTYPH)3,4,3
    4 NH=NH+OAVS
    3 CONTINUE
      NONC=NA-NH
      LIM1=NA*(NA-1)/2
      LIM2=NH*NONC+(NONC*(NONC-1))
      LIM3=NONC*(NONC-1)/2
      LIM4=(NH*NONC+ NONC*(NONC-1))*2
```

```
      LIM5=2*NONC*(NONC-1)
      LIM6=LIM5
      LIM7=LIM6
       NS=LIM1+LIM2+LIM3+LIM4+LIM5+LIM6+LIM7
      WRITE(IDISC)NS
       KK=0
      DO 10  LA=1,2
      DO 10 LB=1,2
      IF(LA+LB-2)13,13,14
   13 WRITE(IDISC)LIM1,I1,I1
      GO TO 17
   14 IF(LA+LB-3)21,21,16
   21 KK=KK+1
      IF(KK-1)15,15,17
   15 WRITE(IDISC)LIM2,I1,I2
      GO TO 17
   16 WRITE(IDISC)LIM3,I2,I2
   17 CALL CLASS(LA,LB)
   10 CONTINUE
      WRITE(IDISC)LIM4,I1,I3
      LA=1
      DO 20  LB=3,4
   20 CALL CLASS(LA,LB)
      LB=1
      DO 30  LA=3,4
   30 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM5,I2,I3
      LA=2
      DO 40  LB=3,4
   40 CALL CLASS(LA,LB)
      LB=2
      DO 50  LA=3,4
   50 CALL CLASS(LA,LB)
      WRITE(IDISC)LIM6,I3,I3
      LA=3
      LB=3
      CALL CLASS(LA,LB)
      LA=4
      LB=4
      CALL CLASS(LA,LB)
      LA=3
      LB=4
      CALL CLASS(LA,LB)
      LA=4
      LB=3
```

```
CALL CLASS(LA,LB)
WRITE(IDISC)LIM7,I4,I4
LA=5
LB=5
CALL CLASS(LA,LB)
LA=6
LB=6
CALL CLASS(LA,LB)
LA=5
LB=6
CALL CLASS(LA,LB)
LA=6
LB=5
CALL CLASS(LA,LB)
WRITE(IDISC)NA
WRITE(IDISC)(NE(I),I=1,NA)
RETURN
END
```

```
// EXEC FORTRAN
      SUBROUTINE CLASS(LA,LB)
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DOUBLE PRECISION ZD,R,ER1,DZA,DZB
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      NA=MA
      NNA=NA-1
      DO 11 KA=1,NNA
      KKA=KA+1
      DO 11 KB=KKA,NA
      IF (LA-LB)1,1,2
    1 JJ=NE(KA)
      KK=NE(KB)
      DZA=ZD(JJ,LA)
      DZB=ZD(KK,LB)
      GO TO 3
    2 JJ=NE(KA)
      KK=NE(KB)
      DZA=ZD(KK,LB)
      DZB=ZD(JJ,LA)
    3 IF(DABS(DZA)-1.D-10)11,11,4
    4 IF(DABS(DZB)-1.D-10)11,11,5
    5 ER1=R(KA,KB)
      WRITE(IDISC)KA,KB,LA,LB,DZA,DZB,ER1
   11 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE VSYMET(X,Y,Z,NA,ISORT,IP1,IP2)
      DIMENSION X(40),Y(40),Z(40),MATSYM(3,3),ZI(3),ZJ(3),SJ(3)
      DIMENSION NE(40),ZD(4,8),R(40,40)
      DOUBLE PRECISION X,Y,Z,MATSYM,ZI,ZJ,SJ,SOM
      DOUBLE PRECISION ZD,R
      DOUBLE PRECISION EPS1,EPS2
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
 1001 FORMAT(3F10.0)
 1002 FORMAT(2I5)
 9999 FORMAT('OVERIFICATION DE LA SYMETRIE EFFECTUEE')
      EPS1=1.D-6
      EPS2=1.D-6
C  NPCTS  EST LE NOMBRE DE COUPLES DE POINTS QUI DOIVENT
C   VERIFIER LA SYMETRIE DEFINIE PAR LA MATRICE MATSYM
      READ(LEC,1002)NPCTS
C   LECTURE  DE LA MATRICE DE SYMETRIE MATSYM
C  ON NUMEROTE LES ELEMENTS DE LA MATRICE DE SYMETRIE COMME SUIT
C  1          2          3
C  4          5          6
C  7          8          9
      DO 6 I=1,3
      READ(LEC,1001)(MATSYM(I,J),J=1,3)
    6 CONTINUE
C   LECTURE DES NUMERO DES POINTS SE CORRESPONDANT PAR LA SYMETRIE
C   P(I2)=MATSYM*P(I1)
      DO 100 ICOU=1,NPCTS
      READ(LEC,1002)I1,J1
      ZI(1)=X(I1)
      ZI(2)=Y(I1)
      ZI(3)=Z(I1)
      ZJ(1)=X(J1)
      ZJ(2)=Y(J1)
      ZJ(3)=Z(J1)
      DO 1 I=1,3
      SOM=0.0
      DO 2 K=1,3
      SOM=SOM+MATSYM(I,K)*ZI(K)
    2 CONTINUE
      SJ(I)=SOM
      IF(SJ(I)*ZJ(I))11,11,3
   11 IF(DABS(SJ(I))-EPS1 )12,12,10
```

```
 12 IF(DABS(ZJ(I))-EPS1 )1,1,10
  3 IF(DABS(SJ(I)-ZJ(I))-EPS2 )1,1,10
  1 CONTINUE
100 CONTINUE
    ISORT=1
    GO TO 200
 10 ISORT=0
    IP1=I1
    IP2=J1
200 CONTINUE
    RETURN
    END
```

```
// EXEC FORTRAN
      SUBROUTINE DMATIN(A,X,N)
      DIMENSION A(3,3),X(3,3),B(3,3),D(3,3)
      DOUBLE PRECISION A,X,B,D
C               COMMENTAIRE
C               -----------
C A EST LA MATRICE A INVERSER
C  X EST LA MATRICE INVERSE CHERCHEE
C N EST LA DIMENSION DE LA MATRICE CARREE
C  LA MATRICE A EST SAUVEGARDEE
C   ------------------------------------------------------------------
      DO 3 L=1,N
      DO 3 M=1,N
      D(L,M)=A(L,M)
      B(L,M)=0.0
    3 B(L,L)=1.0
      NO=N-1
      DO 11 J=1,N
      DO 4 K=1,NO
      JO=K+1
      DO 4 L=JO,N
      B(L,J)=B(L,J)-(A(L,K)*B(K,J))/A(K,K)
      DO 4 M=JO,N
    4 A(L,M)=A(L,M)-(A(L,K)*A(K,M))/A(K,K)
      DO 10 K=1,N
      JO=N-K+1
      JOP=N-K+2
      X(JO,J)=B(JO,J)
      IF(JOP-N)7,7,9
    7 DO 8 I=JOP,N
    8 X(JO,J)=X(JO,J)-A(JO,I)*X(I,J)
    9 X(JO,J)=X(JO,J)/A(JO,JO)
   10 CONTINUE
      DO 11 L=1,N
      DO 11 M=1,N
   11 A(L,M)=D(L,M)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE CLASCO(X,Y,Z,JTRI,JAXE,NAT,RCOOR)
      DIMENSION XYZ(40,3),JTRI(40),JAXE(40),RCOOR(120,3)
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DIMENSION X(40),Y(40),Z(40)
      DOUBLE PRECISION X,Y,Z
      DOUBLE PRECISION ZD,R
      DOUBLE PRECISION XYZ,RCOOR
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      DO 1 I=1,120
      DO 1 J=1,3
    1 RCOOR(I,J)=0.0
      DO 5021 I=1,NAT
      XYZ(I,1)=X(I)
      XYZ(I,2)=Y(I)
      XYZ(I,3)=Z(I)
 5021 CONTINUE
      DO 2 I=1,NAT
      J=I-1
      J1=JTRI(I)-1
      J2=JAXE(I)+1
      DO 2 K=1,3
      IF(JTRI(I))4,3,4
    3 L=3*J+K
      RCOOR(L,1)=XYZ(I,K)
      GO TO 2
    4 L1=3*J1+K
      RCOOR(L1,J2)=XYZ(I,K)
    2 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE PERN1(MTAB,LTAB,NATOM)
      DIMENSION        MTAB(40),LTAB(40)
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DOUBLE PRECISION ZD,R
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
 9999 FORMAT('0  ENVER  EFFECTUE ')
99998 FORMAT('ERREUR  SUR LE TYPE DE L ATOME NUMERO',I2)
99999 FORMAT('OLE TYPE ',I2,' NE CORRESPOND PAS AUX TYPES DONNES ')
      DO 1 I=1,NATOM
      IF(NE(I)-NTYPH)3,2,3
    2 MTAB(I)=1
      LTAB(I)=1
      GO TO 1
    3 IF(NE(I)-NTYPC)5,4,5
    4 MTAB(I)=4
      LTAB(I)=6
      GO TO 1
    5 IF(NE(I)-NTYPN)6,7,6
    7 MTAB(I)=5
      LTAB(I)=7
      GO TO 1
    6 IF(NE(I)-NTYPO)9,8,9
    9 WRITE(IMP,99998)I
      WRITE(IMP,99999)NE(I)
      GO TO 1
    8 CONTINUE
      MTAB(I)=6
      LTAB(I)=8
    1 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE EREPNU(MTAB,NATOM,REP)
      DIMENSION MTAB(40)
      DIMENSION NE(40),R(40,40),ZD(4,8)
      DOUBLE PRECISION ZD,R
      DOUBLE PRECISION REP
      INTEGER OAVSP,OAVS
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,NDIMZD
      COMMON OAVSP,OAVS
      COMMON NE,ZD,R,MA
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      REP=0.0
      DO 1 I=2,NATOM
      IJ=I-1
      DO 1 J=1,IJ
      IF(R(I,J)-1.D-10)1,1,2
    2 CONTINUE
      REP=REP+MTAB(J)*MTAB(I)/R(I,J)
    1 CONTINUE
      REP=REP*0.529172*27.2098
      RETURN
      END
// EXEC LNKEDT
// ASSGN SYS001,X'192'
// ASSGN SYS002,X'192'
// ASSGN SYS003,X'191'
// ASSGN SYS004,X'192'
// ASSGN SYS005,X'192'
// ASSGN SYS006,X'192'
// DLBL IJSYS03,'COP1 DOUADY RECOUVREMENT',69/365
// EXTENT ,9,,,1960,20
// DLBL IJSYS04,,1
// EXTENT ,6,,,1100,200
// DLBL IJSYS05,,1
// EXTENT ,6,,1300,200
// DLBL IJSYS06,,1
// EXTENT ,6,,1500,200
// EXEC

    2    3    7    8    9    6    5    4

    4
    1    2    3    4

    1    3
```

```
7.555      13.310     13.310      13.310
13.310     7.935      13.245      13.245
13.310      13.245     7.935       13.245
13.310     13.245     13.245      7.935

6.985      12.125     12.125      12.125
12.125     6.855      11.635      11.635
12.125     11.635     6.855       11.635
12.125     11.635     11.635      6.855

5.865      10.185     10.185      10.185
10.185     5.755      9.900       9.900
10.185     9.900      5.755       9.900
10.185     9.900      9.900       5.755

6.425


-100.710   -84.096    -84.096     -84.096

-76.235    -61.808    -61.808     -61.808

-51.255    -41.830    -41.830     -41.830

-13.590

    4

7.7        2.24        1.57        3.35       1.57        3.35

6.7        1.92        1.35        2.65       1.35        2.65

5.7        1.61        1.10        2.10       1.10         2.10

1.2        0.0         0.0         0.0        0.0         0.0

-0.233898  1.026990    0.0         0.0        0.0         0.0
0.0        0.0         0.679941    0.396944   0.0         0.0
0.0        0.0         0.0         0.0        0.679941    0.396944


-0.228031  1.025669    0.0         0.0        0.0         0.0
0.0        0.0         0.629890    0.434702   0.0         0.0
0.0        0.0         0.0         0.0        0.629890    0.434702
```

```
-0.220166 1.023950  0.0       0.0       0.0       0.0
0.0       0.0       0.603696  0.456963  0.0       0.0
0.0       0.0       0.0       0.0       0.603696  0.456963


1.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0


 H-H-C-O  DERNIER PASSAGE  TOTAL

    8     4     4     1

-1.21                  0.0                  0.0
0.0                    0.0                  0.0
0.5768426              0.960027             0.0
0.5768426             -0.960027             0.0

-0.21                  0.0                  0.0
-1.21                  1.0                  0.0
1.0                    0.0                  0.0
0.0                    1.0                  0.0

    1     3     4     4

    1

    5

3.0       0.378174  -8.64

1.8       1.6       1.6       1.0       1.0       1.0       1.0       1.
1.0       1.0

   12     6     0

// OPTION LINK
```

PROGRAMME

INTEGRALES DE RECOUVREMENTS

```
// EXEC FORTRAN
C      DEBUT DU PROFRAMME
C DEBUT DE LA PREMIERE CHAINE
       DIMENSION COEF(4,3,8)
       DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
       DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
       DIMENSION FA(5), FB(5), FC(5), AA(5,4), AB(5,4), AC(5,4), FC1(10)
       DIMENSION B(40,4,4),X(40,3),XP1(40,3),XP2(40,3),NE(40)
       DIMENSION A(4,4),C(10,3,8),D(4,4),S1(8,8),S2(4,4),S3(4,4)
       DOUBLE PRECISION FA,FB,FC,AA,AB,AC,FC1
       DOUBLE PRECISION B,X,XP1,XP2
       DOUBLE PRECISION A,C,D,S1,S2,S3
       DOUBLE PRECISION COEF
       DOUBLE PRECISION  SDES,WDEW,ZDEZ
       DOUBLE PRECISION OZ,CZ,ZN
       DOUBLE PRECISION OW,CW,WN
       DOUBLE PRECISION           FACT,FACT1,DZA,DZB,ER1,VAL,ER,RES
       DOUBLE PRECISION F1,F2,F3,PS,S,C1,C2,C3,G1
       DOUBLE PRECISION TAU,CAP,RHO,POL,DZ,RO,COFC,COFD,SOM,D1,D2,D3
       DOUBLE PRECISION PMOINS,PPLUS
       DOUBLE PRECISION TKO,ESSEO,ZALPHA
       DOUBLE PRECISION HZ
       DOUBLE PRECISION HW
       DOUBLE PRECISION EPSILO
       INTEGER OAVSP,OAVS
       COMMON IB1
       COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
       COMMON NTYPO,NTYPN,NTYPC,NTYPH
       COMMON OAVSP,OAVS
       COMMON NE
       COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
       COMMON SDES,ZDEZ,WDEW
       COMMON NA,NESP,NDIMZD,NS
       COMMON X,XP1,XP2
        COMMON COEF
       KDISC=9
       CALL LECPER
       CALL ROOTHA
C    DEBUT DE LA DEUXIEME  CHAINE
       CALL BROT
C    DEBUT DE LA TROISIEME CHAINE
       CALL AROT
C    DEBUT DE LA QUATRIEME CHAINE
       CALL RESULT
       END
```

```
// EXEC FORTRAN
      SUBROUTINE LECPER
      DIMENSION COEF(4,3,8)
      DIMENSION X(40,3),XP1(40,3),XP2(40,3)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DIMENSION NE(40)
      DOUBLE PRECISION COEF
      DOUBLE PRECISION X,XP1,XP2
      DOUBLE PRECISION SDES,ZDEZ,WDEW
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
      COMMON X,XP1,XP2
       COMMON COEF
 5510 FORMAT(4F10.5)
 5511 FORMAT(F10.5)
C               COMMENTAIRE
C               ----------
C  NTYPO,NTYPN,NTYPC,NTYPH  SONT LES DIFFERENTS TYPES AFFECTES AUX ATOMES
C  OXYGENE  ,AZOTE, CARBONE HYDROGENE RESPECTIVEMENT
C   NTYPO=1
C  NTYPN=2
C  NTYPC=3
C  NTYPH=4
C  ON PEUT PRENDRE DES NOMBRES DIFFERENTS POUR CES TYPES AUX CONDITIONS
C   SUIVANTES
C  NTYPH  DOIT ETRE DIFFERENT DE 1
C   LE PRODUIT DE NTYPH PAR LUI-MEME DOIT ETRE DIFFERENT DU PRODUIT
C  DE DEUX  AUTRES TYPES (RESP O,N,C)
      READ(KDISC)LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      READ(KDISC)NESP
      READ(KDISC)NTYPO,NTYPN,NTYPC,NTYPH
      READ(KDISC)OAVSP,OAVS
      READ(JDISC)NDIMZD
      READ(JDISC)OZ,ZN,CZ,HZ
      READ(JDISC)OW,WN,CW,HW
      READ(JDISC)COEF
      RETURN
```

END

```
// EXEC FORTRAN
      SUBROUTINE ROOTHA
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DIMENSION X(40,3),XP1(40,3),XP2(40,3)
      DIMENSION COEF(4,3,8)
      DIMENSION NE(40)
      DIMENSION FA(5), FB(5), FC(5), AA(5,4), AB(5,4), AC(5,4), FC1(10)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DOUBLE PRECISION  SDES,WDEW,ZDEZ
      DOUBLE PRECISION X,XP1,XP2
      DOUBLE PRECISION FA,FB,FC,AA,AB,AC,FC1,PMOINS,PPLUS,FACT
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      DOUBLE PRECISION DZA,DZB,ER1,VAL,RES,ER,TAU,CAP,RHO,POL,DZ,RO
      DOUBLE PRECISION COFC,COFD,SOM,FACT1
      DOUBLE PRECISION COEF
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
      COMMON X,XP1,XP2
       COMMON COEF
   17 FORMAT(4I3,2F7.4,F9.6)
   23 FORMAT (//6H TYPE I4,I6//)
   67 FORMAT (I5,I5,F10.4,F10.4,F10.4,D18.8//)
  106 FORMAT (22X,38H INTEGRALES DE RECOUVREMENT. ROOTHAAN.//)
  150 FORMAT (I3,2I4)
  151 FORMAT (I4)
  500 FORMAT(I5)
  501 FORMAT('OLE TABLEAU DES DZETA EST DE DIMENSION  ',I2,'PAR ',I2)
 2002 FORMAT(2I4,5I1,2F4.1,F11.8)
 2003 FORMAT(2F5.1)
 2004 FORMAT(I1,F11.8)
 2007 FORMAT(F11.8)
      WRITE(IMP,501)NDIMZD
      WRITE(IMP,106)
      READ(IDISC)NS
      NOMB= 0
    1 READ(IDISC)LIM,NT1,NT2
C
 2001 DO 2008 I=1,5
      FA(I)=0.
```

```
      FB(I)=0.
      FC(I)=0.
      DO 2008 J=1,4
      AA(I,J)=0.
      AB(I,J)=0.
 2008 AC(I,J)=0.
      DO 2009 I=1,10
 2009 FC1(I)=0.0
      READ(LDISC,2002)NTA,NTB,MI,NI,IA,IB,IR,PMOINS,PPLUS,FACT
      MX=1
      DO 2010 I=1,NI
      READ(LDISC,2003)FA(I),FB(I)
      MX=MX+1
      IF(MX-MI)2006,2006,2005
 2005 MX=MI
 2006 DO 2010 J=1,MX
 2010 READ(LDISC,2003)AA(I,J),AB(I,J)
      READ(LDISC,2004)NI1,FACT1
      DO 2110 I=1,NI1
 2110 READ(LDISC,2007)FC1(I)
      WRITE(IMP,23)NTA,NTB
C
      DO 1500 NS1=1,LIM
   12 NOMB= NOMB + 1
      READ(IDISC)KA,KB,LA,LB,DZA,DZB,ER1
      IF(DABS(ER1))22,65,22
   65 VAL=0.
      GO TO 66
C
   22 ER= ER1/0.529172
      IF(DZA - DZB) 24, 118, 24
   24 RES= 0.
      TAU= (DZA - DZB)/(DZA + DZB)
      CAP= (TAU + 1./TAU)/2.
      K= 1
C
      RHO= DZA*ER
      DO 25 I=1, NI
      FC(I)= FA(I)
      DO 25 J= 1, MI
   25 AC(I,J)= AA(I,J)
C
   26 MX= 1
      DO 35 I= 1, NI
      MX= MX + 1
```

```
      IF(MX-MI)1031,1031,1030
 1030 MX=MI
 1031 DO 34 J=2,MX
   34 AC(I,1)= AC(I,1)*CAP + AC(I,J)
   35 AC(I,1)= AC(I,1)*FC(I)
      DO 40 I= 2, NI
   40 AC(1,1)= AC(1,1)*RHO + AC(I,1)
      POL=AC(1,1)*DEXP(-RHO)
C
      GO TO (45,55), K
   45 RES= RES - POL*(1. - CAP)**IA
      RHO= DZB*ER
      DO 50 I= 1, NI
      FC(I)= FB(I)
      DO 50 J= 1, MI
   50 AC(I,J)= AB(I,J)
      K= K + 1
      GO TO 26
C
   55 RES= RES + POL*(1. + CAP)**IB
      DZ= (DZA + DZB)/2.
      RO= (DZ*ER)**IR
      COFC= (1. - TAU)**PMOINS
      COFD= (1. + TAU)**PPLUS
      VAL= (FACT*RES)/(RO*TAU*COFC*COFD)
      GO TO 66
C
  118 RO= DZA*ER
      SOM= FC1(1)
      DO 120 I= 2, NI1
  120 SOM= SOM*RO + FC1(I)
      VAL=FACT1*SOM*DEXP(-RO)
C
   66 WRITE(IMP,67)KA,KB,DZA,DZB,ER1,VAL
      WRITE(NDISC)KA,KB,LA,LB,VAL
 1500 CONTINUE
      IF(NS-NOMB)2600,2600,1
 2600 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE BROT
      DIMENSION COEF(4,3,8)
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DIMENSION NE(40),B(40,4,4),X(40,3),XP1(40,3),XP2(40,3)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DOUBLE PRECISION  SDES,WDEW,ZDEZ
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      DOUBLE PRECISION B,X,XP1,XP2
      DOUBLE PRECISION COEF
      DOUBLE PRECISION D1,D2,D3,F1,F2,F3,PS,PPPS,EPSILO,G1
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
      COMMON X,XP1,XP2
       COMMON COEF
   89 FORMAT (10X,21H MATRICE B.ROTATIONS.//)
  501 FORMAT (I2,I4)
  502 FORMAT (I2)
  503 FORMAT(3F20.14)
  602 FORMAT (4F12.8)
 2505 FORMAT('OPRODUIT SCALAIRE DES AXES 1 ET 2 DU TRIEDRE LOCAL')
 2512 FORMAT(1H0,F13.8)
 2516 FORMAT(1H0,I2)
      EPSILO=1.D-5
      READ(IDISC)NA
      READ(IDISC)(NE(I),I=1,NA)
      DO 2 I=1,NA
               DO3 I1=1,4
      DO 3  I2= 1,4
    3 B(I,I1,I2)=0.
    2 B(I,1,1)=1.
      REWIND MDISC
   13 READ(IDISC)((X(I1,I2),XP1(I1,I2),XP2(I1,I2),I2=1,3),I1=1,NA)
  349 DO 350  ID=1,NA
      I=NE(ID)
      IF(I-NTYPH)61,62,61
   62 DO 63 I1=1,4
   63 B(ID,I1,I1)=1.
          GO TO 348
```

```
 61 CONTINUE
    D1=XP1(ID,1)-X(ID,1)
    D2=XP1(ID,2)-X(ID,2)
    D3=XP1(ID,3)-X(ID,3)
    F1=XP2(ID,1)-X(ID,1)
    F2=XP2(ID,2)-X(ID,2)
    F3=XP2(ID,3)-X(ID,3)
    B(ID,1,1)=1.
    B(ID,2,2)=D1
    B(ID,2,3)=D2
    B(ID,2,4)=D3
    B(ID,3,2)=F1
    B(ID,3,3)=F2
    B(ID,3,4) =F3
    PS=D1*F1 + D2*F2 + D3*F3
405 WRITE(IMP,2505)
    WRITE(IMP,2512)PS
 71 B(ID,4,2)= D2*F3 - F2*D3
    B(ID,4,3)= D3*F1 - F3*D1
    B(ID,4,4)= D1*F2 - F1*D2
    DO 79 I1=2,4
    G1=0.
    DO 78 I2=2,4
 78 G1=G1 + B(ID,I1,I2)**2
    DO 79 I2=2,4
 79 B(ID,I1,I2)=B(ID,I1,I2)/DSQRT(G1)
348 WRITE(IMP,2516)ID
    WRITE(IMP,89)
    WRITE(IMP,602)((B(ID,I1,I2),I2=1,4),I1=1,4)
350 CONTINUE
351 WRITE(MDISC) B
    REWIND MDISC
    RETURN
    END
```

```
// EXEC FORTRAN
      SUBROUTINE AROT
      DIMENSION B(40,4,4)
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DIMENSION           NE(40),S1(8,8),S2(4,4),A(4,4),D(4,4)
      DIMENSION S3(4,4)
      DIMENSION COEF(4,3,8)
      DIMENSION X(40,3),XP1(40,3),XP2(40,3)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DOUBLE PRECISION B
      DOUBLE PRECISION  SDES,WDEW,ZDEZ
      DOUBLE PRECISION COEF
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      DOUBLE PRECISION C,S1,S2,A,X,D,S3,XP1,XP2
      DOUBLE PRECISION S,C1,C2,C3
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
      COMMON X,XP1,XP2
       COMMON COEF
  499 FORMAT(20I2)
  500 FORMAT(8F10.6)
      IDES=1
      JDES=1
      IDEZ=1
      JDEZ=1
      IDEW=1
       DO 3040 I=1,32
       DO 3040 J=1,32
      SDES(I,J)=0.
 3040 ZDEZ(I,J)=0.
       DO 3041 I=1,32
      WDEW(I)=0.
 3041 SDES(I,I)=1.
      N1=NA-1
      READ(MDISC) B
      REWIND MDISC
      DO 1150 IA=1,NA
      N2=IA+1
      DO 1150 IB=IA,NA
```

```
      I=NE(IA)
      J=NE(IB)
      IF(IB-IA)3060,3000,3060
 3060 IF(IA-NA)3070,3000,3070
 3070 DO 15 I1=1,NDIMZD
      DO 15 I2=1,NDIMZD
   15 S1(I1,I2)=0.
      REWIND NDISC
      NP = 1
      DO 30 I3= 1, NS
      READ(NDISC) JA,JB,L,M,S
      I2=IABS(JA-IA)+IABS(JB-IB)
      IF(I2) 30,20,30
   20 S1(L,M)=S
      NP = 2
   30 CONTINUE
C     ATOME1 ESPECE ATOME2 ESPECE
      GO TO (1150,31),NP
   31 DO 1035 I1=1,4
      DO 1035 I2=1,4
 1035 S2(I1,I2)=0.
      DO 1050 I1=1,3
      DO 1050 I4=1,3
      S=0.
      DO 1040 I2=1,NDIMZD
      DO 1040 I3=1,NDIMZD
 1040 S=S+COEF(I,I1,I2)*S1(I2,I3)*COEF(J,I4,I3)
 1050 S2(I1,I4)=S
      S2(4,4)=S2(3,3)
C     RECOUVREMENTS ROOTHAN SIGMA PI
      DO 1055 I1=1,4
      DO 1055 I2=1,4
 1055 A(I1,I2)=0.
      A(1,1)=1.
      A(2,4)=1.
      A(3,2)=1.
      A(4,3)=1.
      C1 = X(IB,1)-X(IA,1)
      C2= X(IB,2)-X(IA,2)
      C3 = X(IB,3)-X(IA,3)
      IF(DABS(C1)+DABS(C2))56,57,56
   56 A(1,1)=1.
      A(2,2)=C1
      A(2,3)=C2
      A(2,4)=C3
```

```
      A(3,2)=-C2
      A(3,3)=C1
      A(4,2)=-C1*C3
      A(4,3)=-C2*C3
      A(4,4)= C1**2 + C2**2
   57 DO 70 I1 =2,4
      C1=0
      DO 60 I2=2,4
   60 C1=C1 +A(I1,I2)**2
      DO70 I2 =2,4
   70 A(I1,I2)=A(I1,I2)/DSQRT(C1)
      DO 80 I1 =1,4
      DO 80 I2 =1,4
   80 D(I1,I2)= A(I1,I2)
      DO 90 I1=1,4
   90 D(2,I1)= -A(2,I1)
      DO 110 I1=1,4
      DO 110 I6=1,4
      S=0.
      DO 100 I2=1,4
      DO 100 I3=1,4
      DO 100 I4= 1,4
      DO 100 I5= 1,4
  100 S=S+B(IA,I1,I2)*A(I3,I2)*S2(I3,I4)*D(I4,I5)*B(IB,I6,I5)
  110 S3(I1,I6)=S
 3000 CONTINUE
            CALL RANG(IDES,JDES,IDEZ,JDEZ,IDEW,I,J,IA,IB,S3)
      IF(IA-IB)3061,1150,3061
 3061 IF(IA-NA)3071,1150,3071
 3071 WRITE(MDISC) IA,I,IB,J,S1,S2,A,S3
 1150 CONTINUE
      REWIND MDISC
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE RANG(IDES,JDES,IDEZ,JDEZ,IDEW,I,J,IA,IB,S3)
      DIMENSION NE(40)
      DIMENSION S3(4,4)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DOUBLE PRECISION  SDES,WDEW,ZDEZ
      DOUBLE PRECISION S3
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
      NTYPHH=NTYPH*NTYPH
      IF(I*J-NTYPHH)3002,3001,3002
 3001 IF(IA-IB)3004,3003,3004
 3003 ZDEZ(IDEZ,IDEZ)=HZ
      WDEW(IDEZ)=HW
      IDEZ=IDEZ+OAVS
      IF(IB-NA)3006,3005,3005
 3005 JDES=JDES+OAVS
      IDES=JDES
      GO TO 5000
 3006 IDES=IDES+OAVS
      GO TO 5000
 3004 SDES(JDES,IDES)=S3(1,1)
      IF(IB-NA)3080,3081,3081
 3081 JDES=JDES+OAVS
      IDES=JDES
      GO TO 5000
 3080 IDES=IDES+OAVS
      GO TO 5000
 3002 IF(I-NTYPH)3008,3007,3008
 3007 DO 4000 IHI=1,OAVSP
      IIDES=IDES+IHI-1
 4000 SDES(JDES,IIDES)=S3(1,IHI)
      IF(IB-NA)3010,3009,3009
 3009 JDES=JDES+OAVS
      IDES=JDES
      GO TO 5000
 3010 IDES=IDES+OAVSP
```

```
      GO TO 5000
3008  IF(J-NTYPH)3012,3011,3012
3011  DO 4001 JHJ=1,OAVSP
      JJDES=JDES+JHJ-1
4001  SDES(JJDES,IDES)=S3(JHJ,1)
      IF(IB-NA)3014,3013,3014
3013  JDES=JDES+OAVSP
      IDES=JDES
      GO TO 5000
3014  IDES=IDES+OAVS
      GO TO 5000
3012  IF(IA-IB)3017,3018,3017
3018  IF(I-NTYPC)3031,4030,3031
4030  DO 3036 III=1,OAVSP
      IIDEZ=IDEZ+III-1
      DO 3036 JJJ=1,OAVSP
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=CW(III)
3036  ZDEZ(IIDEZ,JJDEZ)=CZ(III,JJJ)
      GO TO 3035
3031  IF(I-NTYPN)3033,3032,3033
3032  DO 3037 III=1,OAVSP
      IIDEZ=IDEZ+III-1
      DO 3037 JJJ=1,OAVSP
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=WN(III)
3037  ZDEZ(IIDEZ,JJDEZ)=ZN(III,JJJ)
      GO TO 3035
3033  IF(I-NTYPO)3035,3034,3035
3034  DO 3038 III=1,OAVSP
      IIDEZ=IDEZ+III-1
      DO 3038 JJJ=1,OAVSP
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=OW(III)
3038  ZDEZ(IIDEZ,JJDEZ)=OZ(III,JJJ)
3035  IDEZ=IDEZ+OAVSP
      IF(IB-NA)3019,3020,3020
3020  JDES=JDES+OAVSP
      IDES=JDES
      GO TO 5000
3019  IDES=IDES+OAVSP
      GO TO 5000
3017  DO 4002 JHJ=1,OAVSP
      JJDES=JDES+JHJ-1
      DO 4002 IHI=1,OAVSP
```

```
      IIDES=IDES+IHI-1
 4002 SDES(JJDES,IIDES)=S3(JHJ,IHI)
      IF(IB-NA)3016,3015,3015
 3016 IDES=IDES+OAVSP
      GO TO 5000
 3015 JDES=JDES+OAVSP
      IDES=JDES
 5000 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE RESULT
      DIMENSION NE(40)
      DIMENSION SDES(32,32),ZDEZ(32,32),WDEW(32)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DIMENSION S1(8,8),S2(4,4),S3(4,4),A(4,4)
      DOUBLE PRECISION S1,S2,S3,A
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      DOUBLE PRECISION  SDES,WDEW,ZDEZ
      INTEGER OAVSP,OAVS
      COMMON IB1
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON NTYPO,NTYPN,NTYPC,NTYPH
      COMMON OAVSP,OAVS
      COMMON NE
      COMMON HZ,CZ,ZN,OZ,HW,CW,WN,OW
      COMMON SDES,ZDEZ,WDEW
      COMMON NA,NESP,NDIMZD,NS
    9 FORMAT (//22X,28H INTEGRALES DE RECOUVREMENT.//)
   19 FORMAT (//22X,22H MATRICE S1. ROOTHAAN.//)
   29 FORMAT (//10X,24H MATRICE S2. S.SIGMA.PI.//)
   39 FORMAT (//10X,21H MATRICE A.ROTATIONS.//)
   49 FORMAT (//10X,40H INTEGRALES DE RECOUVREMENT.SYSTEME BMV.//)
   59 FORMAT (//17H AT1  E1 AT2  E2 )
  151 FORMAT (I4)
  600 FORMAT (4I4)
  601 FORMAT(8F10.8)
  602 FORMAT (4F12.8)
 2508 FORMAT(1H1)
 2512 FORMAT(1H0,F13.8)
 2513 FORMAT(4F20.14)
 2514 FORMAT(1H0,25F4.1)
 2515 FORMAT(6I5)
      NDIM=0
      DO 3022 I=1,NA
      IF(NE(I)-NTYPH)3023,3024,3023
 3024 NDIM=NDIM+1
      GO TO  3022
 3023 NDIM=NDIM+4
 3022 CONTINUE
      REWIND IDISC
      WRITE(IMP,2508)
      DO 3021  III=1,NDIM
      DO 3021 JJJ=1,III
 3021 SDES(III,JJJ)=SDES(JJJ,III)
      WRITE(IDISC)((SDES(III,JJJ),III=1,JJJ),JJJ=1,NDIM)
```

```
      WRITE(IDISC)((ZDEZ(III,JJJ),III=1,JJJ),JJJ=1,NDIM)
      WRITE(IDISC)((WDEW(III),III=1,NDIM))
      WRITE(IDISC)NA
      WRITE(IDISC)(NE(III),III=1,NA)
      WRITE(IMP,9)
      N1=NA-1
      DO 2150 J2=1,N1
      N2 =J2 +1
      DO 2150 J3=N2,NA
      READ(MDISC) IA,I,IB,J,S1,S2,A,S3
      WRITE(IMP,59)
      WRITE(IMP,600)IA,I,IB,J
      WRITE(IMP,19)
      DO 800 I1=1,NDIMZD
  800 WRITE(IMP,601)(S1(I1,I2),I2=1,NDIMZD)
      WRITE(IMP,29)
      WRITE(IMP,602)((S2(I1,I2),I2=1,4),I1=1,4)
      WRITE(IMP,39)
      WRITE(IMP,602)((A(I1,I2),I2=1,4),I1=1,4)
      WRITE(IMP,59)
      WRITE(IMP,600)IA,I,IB,J
      WRITE(IMP,49)
      WRITE(IMP,602)((S3(I1,I2),I2=1,4),I1=1,4)
 2150 CONTINUE
      RETURN
      END
// EXEC LNKEDT
// ASSGN SYS001,X'192'
// ASSGN SYS002,X'192'
// ASSGN SYS003,X'191'
// ASSGN SYS004,X'192'
// ASSGN SYS005,X'192'
// ASSGN SYS006,X'192'
// DLBL IJSYS03,'COP1 DOUADY RECOUVREMENT',69/365
// EXTENT ,9,,,1960,20
// DLBL IJSYS04,,1
// EXTENT ,6,,,1100,200
// DLBL IJSYS05,,1
// EXTENT ,6,,1300,200
// DLBL IJSYS06,,1
// EXTENT ,6,,1500,200
// EXEC
0100010022111-0.5-0.5+1.00000000
+01.0+01.0
+00.0+00.0
```

```
+01.0+01.0
+02.0+02.0
+01.0-01.0
+01.0+01.0
3+1.0000000
+0.33333333
+1.00000000
+1.00000000

0100020033111-0.5-0.5+0.57735027
+00.0+01.0
+00.0+00.0
+00.0+01.0
+01.0+04.0
+00.0+00.0
-02.0-01.0
+01.0+01.0
+02.0+02.0
-03.0+03.0
-01.0-05.0
+02.0+02.0
1+0.00000000
+0.00000000

0200020043111-0.5-0.5+0.33333333
+01.0+01.0
-02.0+02.0
+01.0+01.0
+04.0+04.0
-03.0-03.0
-01.0+01.0
+02.0+02.0
+02.0+02.0
-12.0+12.0
-12.0-12.0
+07.0-07.0
+07.0+07.0
5+1.00000000
+0.02222222
+0.11111111
+0.44444444
+1.00000000
+1.00000000

0100021134212+0.5-0.5+1.00000000
```

```
+00.0+01.0
+00.0+00.0
+00.0+01.0
+02.0+04.0
+00.0+00.0
+00.0-01.0
+01.0+01.0
+06.0+06.0
+00.0+01.0
+01.0-02.0
+01.0+01.0
+06.0+06.0
+00.0+01.0
+01.0-02.0
+01.0+01.0
1+0.00000000
+0.00000000

0200021144212+0.5-0.5+0.57735027
+02.0+01.0
+00.0+02.0
+01.0+01.0
+02.0+04.0
+00.0-03.0
+06.0+01.0
+05.0+02.0
+06.0+06.0
+00.0+04.0
+04.0-05.0
+07.0-02.0
+03.0+03.0
+06.0+06.0
+00.0+04.0
+04.0-05.0
+07.0-02.0
+03.0+03.0
5+0.28867514
+0.13333333
+0.46666666
+1.00000000
+1.00000000
+0.00000000

0211021135223+0.5+0.5+1.00000000
+02.0+02.0
```

```
+00.0+00.0
+01.0+01.0
+02.0+02.0
+00.0+00.0
+06.0-06.0
+05.0+05.0
+48.0+48.0
+00.5+00.5
+01.0-01.0
+00.5+00.5
+48.0+48.0
+01.0+01.0
+02.0-02.0
+01.0+01.0
+48.0+48.0
+01.0+01.0
+02.0-02.0
+01.0+01.0
5+1.00000000
+0.06666666
+0.13333333
-0.20000000
-1.00000000
-1.00000000

0210021034223+0.5+0.5+1.00000000
+02.0+02.0
+00.0+00.0
+01.0+01.0
+12.0+12.0
+00.0+00.0
+01.0-01.0
+01.0+01.0
+24.0+24.0
+01.0+01.0
+02.0-02.0
+01.0+01.0
+24.0+24.0
+01.0+01.0
+02.0-02.0
+01.0+01.0
4+1.00000000
+0.06666666
+0.40000000
+1.00000000
```

```
+1.00000000
// OPTION LINK
```

PROGRAMME

FONCTION D'ONDE

```
// EXEC FORTRAN
      DIMENSION S(32,32),Z(32,32),W(32),Q(32),RNS(32,32),H(32,32)
      DIMENSION ABO(32),AB1(32),AB2(32),AB(32)
      DIMENSION VPS(32,32),VS(32),ENERG(32),SPIN(32)
      DIMENSION G(32,32),NTYP(32),MTAB(32)
      DOUBLE PRECISION S,Z,W,Q,RNS,H,ABO,AB1,AB2,AB,C,VPS,VS,G
      DOUBLE PRECISION SPIN,ENERG,TKO,ALFAO,SO,SOMEN
      DOUBLE PRECISION SA,ST,TM,TN,SOM1
      DOUBLE PRECISION TSPIN,TCHARG
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON M,NIT,NELEC,N,NSO,NDO,NA,NTYP,MTAB
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,KSACK
      COMMON S,Z,W,Q,TKO,SO,ALFAO,SA,ST,TM
      COMMON H,RNS,G
    6 FORMAT(1H0,20F5.1)
   88 FORMAT(1H0,2F20.8)
  343  FORMAT('0 ENERGIE EN E.V. ET VECTEURS PROPRES ASSOCIES')
  393 FORMAT(29H0 NOMBRE D ITERATIONS DEPASSE)
  395 FORMAT (13H0 FIN CALCULS)
  444 FORMAT(F13.8)
  522 FORMAT(10H0K SO ALFA)
 1113 FORMAT('0VALEURS PROPRES DE LA MATRICE S ')
 1114 FORMAT(1H0,F15.8)
 2003 FORMAT('0DENSITE DE SPIN   VECTEUR CHARGE')
 2005 FORMAT(1H0,'NOMBRE D ITERATIONS')
 2022 FORMAT('0MATRICE H SYMETRIQUE')
 2225 FORMAT('0SOMME DES ELEMENTS DU VECTEUR CHARGE  ',F10.6)
 2226 FORMAT(' SOMME DES ELEMENTS DU VECTEUR DENSITE DE SPIN',F10.6)
 3502 FORMAT(5X,F4.1)
 3503 FORMAT(F20.14)
 3505 FORMAT(1H0,10F8.4)
 3506 FORMAT(4F20.14)
 5001 FORMAT(31H0NOMBRE D ITERATIONS EFFECTUEES)
 5002 FORMAT(I3)
 7777 FORMAT(1H1)
 8801 FORMAT(10X,F10.5,40X,F15.8)
 8802 FORMAT(15X,F15.8,45X,F15.8)
 8804 FORMAT(1H0)
 8805 FORMAT(17H0ITERATION NUMERO,I2)
 8806 FORMAT('0UNE DES VALEURS PROPRES DE S EST NEGATIVE IL Y A UNE
     1ERREUR')
 8809 FORMAT('0KSACK =',I4)
 8810 FORMAT('0VECTEUR CHARGE')
 8811 FORMAT(1H0,F20.8)
 8812 FORMAT('0LA SOMME DES ENERGIES VAUT',F15.8)
      KDISC=9
```

```
      CALL LECTUR
      DO 9  I=1,M
      TN=N
      AB1(I)=0.0
      AB2(I)=0.0
      DO 9  J=1,M
    9 VPS(I,J)=S(I,J)
C
      CALL DIAGO(VPS,VS,M)
       WRITE(IMP,1113)
      DO 1115 I=1,M
 1115 WRITE(IMP,1114)VS(I)
C LES VECTEURS PROPRES SONT LES LIGNES DE VPS
      DO 11  I=1,M
      IF(VS(I))10,11,11
   10 WRITE(IMP,8806)
      STOP
   11 CONTINUE
C   RNS   EST LA MATRICE S-1/2
            CALL SDEMI(VPS,VS,RNS,M)
C LES CALCULS SUIVANTS NOUS FONT PASSER D UN VECTEUR CHARGE Q
C           A UN VECTEUR CHARGE AB
            CALL CREVS(VS,NSO,NDO,M,L)
C NCOMP COMPTE TOUTES LES 3 ITERATIONS AFIN DE FAIRE L ITERATION DE SACK
C NITA COMPTE LES ITERATIONS  QUAND NITA VAUT NIT   SORTIE
      NCOMP=0
      NITA=0
      KPUT=1
  391 NITA=NITA+1
      WRITE(IMP,7777)
      WRITE(IMP,8805) NITA
            CALL HAMILT
            CALL ENERGI(ENERG)
      IF(KPUT)2400,2401,2400
 2401 CONTINUE
      SOMEN=0.0
      WRITE(IMP,2022)
      DO 3109 I=1,M
 3109 WRITE(IMP,6)(H(I,J),J=1,I)
      WRITE(IMP,343)
      DO 8800 I=1,M
      II=M+1-I
      SOMEN=SOMEN+ENERG(I)*VS(II)
      WRITE(IDISC)ENERG(I)
      WRITE(IMP,8801)ENERG(I)
```

```
      DO 8803 J=1,M
      WRITE(IDISC)G(J,II)
      WRITE(IMP,8802)G(J,II)
8803  CONTINUE
8800  CONTINUE
      SOMEN=SOMEN/2.0
      WRITE(IMP,8812)SOMEN
2400  CONTINUE
      IF(NSO)8011,8011,8010
8010  CONTINUE
      CALL DSPIN(S,G,SPIN,M,L)
8011  CONTINUE
            CALL CHARGE(S,G,VS,AB,M)
      TSPIN=0.0
      TCHARG=0.0
      DO 2222 I=1,M
      IF(NSO)2222,2222,9003
9003  CONTINUE
      TSPIN=TSPIN+SPIN(I)
2222  TCHARG=TCHARG+AB(I)
      DO 2224 I=1,M
      IF(NSO)9005,9005,9004
9004  CONTINUE
      SPIN(I)=SPIN(I)/TSPIN
9005  CONTINUE
      AB(I)=TN*(AB(I)/TCHARG)
2224  CONTINUE
      IF(NSO)4002,4003,4002
4002  CONTINUE
      WRITE(IMP,2225)TCHARG
      WRITE(IMP,2226)TSPIN
      WRITE(IMP,2003)
      DO 3111 I=1,M
3111  WRITE(IMP,88)SPIN(I),AB(I)
      GO TO 4004
4003  CONTINUE
      WRITE(IMP,2225)TCHARG
      WRITE(IMP,8810)
      DO 4005 I=1,M
4005  WRITE(IMP,8811)AB(I)
4004  LAS=0
      DO 380 J=1,M
      IF(DABS(AB1(J)-AB(J))-0.001)379,379,380
 379  LAS=LAS+1
 380  CONTINUE
```

```
      DO 384 J=1,M
      AB1(J)=AB(J)
  378 AB(J)=(AB(J)+Q(J))/2.
      ABO(J)=AB2(J)
      AB2(J)=Q(J)
  384 Q(J)=AB(J)
      NCOMP=NCOMP+KSACK
      IF(NCOMP-2)2450,38 ,385
C ITERATION DE SACK
  385 NCOMP=0
            CALL PSACK(ABO,AB2,AB,Q,M)
 2450 CONTINUE
      IF(LAS-M)381,394,394
  381 IF(NITA-NIT)391,392,3500
C   SORTIE    DIFFERENTES
C   SOIT LE NOMBRE D ITERATIONS EST DEPASSE
C              SOIT ON A OBTENU UNE BONNE VALEUR
  392 WRITE(IMP,393)
      GO TO 396
  394 CONTINUE
      IF(KPUT)2430,3500,2430
 2430 CONTINUE
      WRITE(IMP,5001)
      WRITE(IMP,5002)NITA
  396 CONTINUE
      KPUT=0
      GO TO 391
 3500 CONTINUE
      WRITE(IMP,395)
      STOP
      END
```

```
// EXEC FORTRAN
      SUBROUTINE LECTUR
      DIMENSION S(32,32),Z(32,32),W(32),Q(32),VS(32)
      DIMENSION NTYP(32),MTAB(32)
      DOUBLE PRECISION S,Z,W,Q,VS,TKO,SO,SA,ST,TM,ALFAO
      INTEGER OAVSP,OAVS
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON M,NIT,NELEC,N,NSO,NDO,NA,NTYP,MTAB
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,KSACK
      COMMON S,Z,W,Q,TKO,SO,ALFAO,SA,ST,TM
    1 FORMAT (I4)
    2 FORMAT(4F20.14)
    4 FORMAT (6I5)
    5 FORMAT(3HOM=,I4)
    6 FORMAT(1HO,20F5.1)
    7 FORMAT (1H ,I4)
   50 FORMAT('OMATRICE S SYMETRIQUE')
   51 FORMAT('OMATRICE Z SYMETRIQUE')
   52 FORMAT(10HOVECTEUR W)
   53 FORMAT('OVECTEUR CHARGE INITIAL')
   54 FORMAT(1H ,4I10)
  522 FORMAT(10HOK SO ALFA)
 1111 FORMAT('O   SA           ST          TM')
 2005 FORMAT(1HO,'NOMBRE D ITERATIONS')
 1112 FORMAT(1HO,3F10.5)
 2006 FORMAT(16HONOMBRE D ATOMES)
 2007 FORMAT(16HOTYPE DES ATOMES)
 2011 FORMAT('ONELEC  N.S.O. N.D.O.')
 8809 FORMAT('OKSACK =',I4)
C                COMMENTAIRE
C                ___________
C
C M = DIMENSION DE LA MATRICE S
C   ------------------------------------------------------------------
C KSACK=1 SI ON VEUT UTILISER L ITERATION DE SACK
C KSACK=0 SI ON NE VEUT PAS UTILISER L ITERATION DE SACK
C   ------------------------------------------------------------------
C    NA = NOMBRE D ATOME
C   ------------------------------------------------------------------
C DEFINITION DE LA STRUCTURE ATOMIQUE DE LA MOLECULE
C  NTYPE (I)   TABLEAU CONTENANT  LE TYPE DES ATOMES COMPOSANT LA
C  MOLECULE  AVEC  LA CONVENTION SUIVANTE
C  ATOME NUMERO (I) =OXYGENE      NTYPE(I)=1
C  ATOME NUMERO (I) =AZOTE        NTYPE(I)=2
C  ATOME NUMERO (I) =CARBONE      NTYPE(I)=3
C  ATOME NUMERO (I) =HYDROGENE    NTYPE(I)=4
```

```
C   ------------------------------------------------------------------
C  NELEC = NOMBRE D ELECTRONS A REPARTIR
C   ------------------------------------------------------------------
C  NIT   =  NOMBRE D ITERATIONS DEMANDEES
C   ------------------------------------------------------------------
C     NDO   = NOMBRE D ORBITALES DOUBLEMENT OCCUPEES
C      NSO   = NOMBRE D ORBITALES SIMPLEMENT OCCUPEES
C
C   ------------------------------------------------------------------
C   ------------------------------------------------------------------
C
C MTAB  TABLEAU INDIQUANT POUR CHAQUE ATOME LE NOMBRE D ORBITALE LUI
CORRESPONDANT
C   ------------------------------------------------------------------
      READ(KDISC)LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      READ(KDISC)NESP
      READ(KDISC)NTYPO,NTYPN,NTYPC,NTYPH
      READ(KDISC)OAVSP,OAVS
      READ(KDISC)NDIM
      READ(KDISC)KSACK
      READ(KDISC)NIT
      READ(KDISC)TKO,SO,ALFAO
C   LECTURE DES CHARGES DE DEPART
      READ(KDISC)(Q(I),I=1,NDIM)
      READ(KDISC)NELEC,NDO,NSO
      M=NDIM
      WRITE(IMP,8809)KSACK
      WRITE(IMP,5)M
      READ(IDISC)((S(I,J),I=1,J),J=1,M)
      READ(IDISC)((Z(I,J),I=1,J),J=1,M)
      READ(IDISC)(W(I),I=1,M)
      READ(IDISC)NA
      READ(IDISC)(NTYP(I),I=1,NA)
      DO 8  I=1,M
      DO 8  J=1,I
      Z(I,J)= Z(J,I)
    8 S(I,J)=S(J,I)
      WRITE(IMP,50)
      DO 3100 I=1,M
 3100 WRITE(IMP,6)(S(I,J),J=1,I)
      WRITE(IMP,51)
      DO 3101 I=1,M
 3101 WRITE(IMP,6)(Z(I,J),J=1,I)
      WRITE(IMP,52)
      DO 3102 I=1,M
```

```
 3102 WRITE(IMP,6)W(I)
      WRITE(IMP,522)
      WRITE(IMP,6)TKO,SO,ALFAO
      WRITE(IMP,2005)
      WRITE(IMP,7)NIT
      WRITE(IMP,2006)
      WRITE(IMP,7) NA
      WRITE(IMP,2007)
      WRITE(IMP,54) (NTYP(I),I=1,NA)
      NANA=NA-1
C
      WRITE(IMP,53)
      DO 3104 I=1,M
 3104 WRITE(IMP,6)Q(I)
      WRITE(IMP,2011)
      WRITE(IMP,4)NELEC,NSO,NDO
C FIN LECTURE DONNEE
C
C
      N=NELEC
  390 SA=SO*ALFAO
      ST=(TKO-1.)/SO
      TM=TKO*0.50
      WRITE(IMP,1111)
      WRITE(IMP,1112)SA,ST,TM
C
      DO 103  I=1,NA
      IF(NTYP(I)-NTYPH)102,101,102
  101 MTAB(I)=1
      GO TO 103
  102 MTAB(I)=4
  103 CONTINUE
      REWIND IDISC
      WRITE(IDISC)LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      WRITE(IDISC)M
      WRITE(IDISC,1)M
      WRITE(IDISC)((S(I,J),I=1,J),J=1,M)
      WRITE(IDISC)NA
      WRITE(IDISC)NELEC,NSO,NDO
      WRITE(IDISC)(MTAB(I),I=1,NA)
C FIN ECRITURES DONNEES
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE SDEMI(VPS,VS,RNS,M)
      DIMENSION VPS(32,32),VS(32),RNS(32,32)
      DOUBLE PRECISION VPS,VS,RNS,SOM1
      DO 13  J=1,M
      DO 13  I=1,J
      SOM1=0.0
      DO 12 K=1,M
   12 SOM1=SOM1+VPS(K,I)*VPS(K,J)/DSQRT(VS(K))
   13 RNS(I,J)=SOM1
      DO 18 I=1,M
      DO 18 J=1,I
      RNS(I,J)=RNS(J,I)
   18 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE HAMILT
      DIMENSION S(32,32),RNS(32,32),G(32,32),NTYP(32),MTAB(32)
      DIMENSION W(32),Z(32,32),Q(32),H(32,32)
      DOUBLE PRECISION S,RNS,G,TKO
      DOUBLE PRECISION W,Z,Q,H,ALFAO,SA,ST,TM,SOM1,SO
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON M,NIT,NELEC,N,NSO,NDO,NA,NTYP,MTAB
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,KSACK
      COMMON S,Z,W,Q,TKO,SO,ALFAO,SA,ST,TM
      COMMON H,RNS,G
      DO 150 I=1,M
      SOM1=0.0
      DO 15 J=1,M
   15 SOM1=SOM1+Z(I,J)*Q(J)
  150 H(I,I)=W(I)+SOM1
C    MODIFICATION DES TERMES DIAGONAUX    TRANSLATION DU ZERO  D ENERGIE
      DO 16 I=1,M
   16 H(I,I)=(H(I,I)-ALFAO)/SA
C    CALCUL DES TERMES NON DIAGONAUX DE L HAMILTONIEN H
      DO 17  J=2,M
      JJ=J-1
      DO 17 I=1,JJ
      H(I,J)=((H(I,I)+H(J,J))*TM+ST)*S(I,J)
   17 CONTINUE
      DO 18  I=1,M
      DO 18  J=1,I
   18 H(I,J)=H(J,I)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE ENERGI(ENERG)
      DIMENSION LIGNE(32)
      DIMENSION S(32,32),Z(32,32),W(32),Q(32)
      DIMENSION H(32,32),RNS(32,32),VPC(32,32),VC(32),ENERG(32)
      DIMENSION G(32,32),IND(32),NTYP(32),MTAB(32)
      DOUBLE PRECISION S,Z,W,Q,SO,ST,TM
      DOUBLE PRECISION ALFAO,SA,TKO
      DOUBLE PRECISION H,RNS,VPC,VC,ENERG,G,SOM1
      DOUBLE PRECISION LIGNE
      COMMON LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      COMMON M,NIT,NELEC,N,NSO,NDO,NA,NTYP,MTAB
      COMMON NTYPO,NTYPN,NTYPC,NTYPH,NESP,KSACK
      COMMON S,Z,W,Q,TKO,SO,ALFAO,SA,ST,TM
      COMMON H,RNS,G
C  CHANGEMENT DE VARIABLE
C       ON POSE  V= S-1/2*C
C    ON CALCULE  UN HAMILTONIEN HP=S-1/2*H*S-1/2
      DO 19 I=1,M
      DO 20 L=1,M
      SOM1=0.0
      DO 21 K=1,M
   21 SOM1=SOM1+RNS(I,K)*H(K,L)
   20 LIGNE(L)=SOM1
      DO 22 J=1,M
      SOM1=0.0
      DO 23 K=1,M
   23 SOM1=SOM1+LIGNE(K)*RNS(K,J)
   22 VPC(I,J)=SOM1
   19 CONTINUE
C     DIAGONALISATION DE L HAMILTONIEN HP
C    OBTENTION DES VALEURS PROPRES ET DES VECTEURS PROPRES DE HP
      CALL DIAGO(VPC,VC,M)
C CLASSEMENT DES VALEURS PROPRES DEC PAR ORDRE CROISSANT
      IPLACE=1
      IND(IPLACE)=1
   24 IPLACE=IPLACE+1
      IND(IPLACE)=IPLACE
      I=IPLACE
   25 I1=IND(I)
      I2=IND(I-1)
      IF(VC(I1)-VC(I2)) 26,28,28
   26 K=IND(I)
      IND(I)=IND(I-1)
      IND(I-1)=K
      IF(I-2) 27,28,27
```

```
   27 I=I-1
      GO TO 25
   28 IF(IPLACE-M) 24,29,24
C CLASSEMENT CORRESPONDANT DES VECTEURS PROPRES  CREATION DE LA MATRICE ORDI
C VPC LES VECTEURS PROPRES SONT LES LIGNES
   29 DO 33 I=1,M
      DO 33 J=1,M
   33 G(I,J)=VPC(J,I)
      DO 34 IPLACE=1,M
      J=IND(IPLACE)
       II=M+1-IPLACE
      ENERG(II)=ALFAO+SA*VC(J)
      DO 34 I=1,M
   34 VPC(I,IPLACE)=G(I,J)
C     RETOUR A LA VARIABLE INITIALE  V=S-1/2*C
      DO 334 I=1,M
      DO 334 J=1,M
      SOM1=0.0
      DO 3340  K=1,M
 3340 SOM1=SOM1+RNS(I,K)*VPC(K,J)
  334 G(I,J)=SOM1
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE CHARGE(S,G,VS,AB,M)
      DIMENSION G(32,32),AB(32),VS(32)
      DIMENSION S(32,32),LIGNE(32)
      DOUBLE PRECISION S
      DOUBLE PRECISION G,AB,VS,SOM1,LIGNE
C   OBTENTION DU NOUVEAU VECTEUR CHARGE  AB
C   CALCUL  DE N*G*GBARRE*S
      DO 6662 I=1,M
      DO 6652 L=1,M
      SOM1=0.0
      DO 6663 K=1,M
6663  SOM1=SOM1+VS(K)*G(I,K)*G(L,K)
6652  LIGNE(L)=SOM1
      SOM1=0.0
      DO 6665 K=1,M
6665  SOM1=SOM1+LIGNE(K)*S(K,I)
6662  AB(I)=SOM1
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE DSPIN(S,G,SPIN,M,L)
      DIMENSION S(32,32),G(32,32),SPIN(32)
      DOUBLE PRECISION S,G,SPIN,SOM1
      DO 5553 I=1,M
      SOM1=0.0
      DO 5554 K=1,M
      SOM1=SOM1+G(I,L)*G(K,L)*S(K,I)
 5554 CONTINUE
 5553 SPIN(I)=SOM1
      RETURN
      END
```

```
EXEC FORTRAN
      SUBROUTINE PSACK(ABO,AB2,AB,Q,M)
      DIMENSION ABO(32),AB2(32),AB(32),Q(32)
      DOUBLE PRECISION ABO,AB2,AB,Q,ENORM,ALF,BET,GAM
      DOUBLE PRECISION TABO,TAB2,TAB,TQ
      TABO=0.0
      TAB2=0.0
      TAB=0.0
      TQ=0.0
       ALF=0.
       BET=0.
       GAM=0.
      DO 386 J=1,M
      TABO=TABO+ABO(J)
      TAB2=TAB2+AB2(J)
      TAB=TAB+AB(J)
      ALF=ALF+(AB(J)-AB2(J))**2
      BET=BET+(AB(J)-AB2(J))*(AB2(J)-ABO(J))
      GAM=GAM+(AB2(J)-ABO(J))**2
  386 CONTINUE
      ENORM = 1./ ( ALF-2.*BET+GAM)
      WRITE(3,9998)
 9998 FORMAT('0ABO',18X,'AB2',18X,'AB',18X,'Q')
      DO 387 J=1,M
      Q(J)=ENORM*(ALF*ABO(J)-2.*BET*AB2(J)+GAM*AB(J))
      TQ=TQ+Q(J)
      WRITE(3,9999)ABO(J),AB2(J),AB(J),Q(J)
  387 CONTINUE
      WRITE(3,9997)
 9997 FORMAT(1H0,80X,'   TOTAL')
      WRITE(3,9999)TABO,TAB2,TAB,TQ
 9999 FORMAT(1H0,4(F10.6,10X))
      RETURN
      END
```

```
' EXEC FORTRAN
      SUBROUTINE CREVS(VS,NSO,NDO,M,L)
      DIMENSION VS(32)
      DOUBLE PRECISION VS
      DO 2 I=1,M
      VS(I)=0.0
    2 CONTINUE
      IF(NSO)10,11,10
   10 CONTINUE
      DO 1 INSO=1,NSO
      L=M-NDO+1-INSO
      VS(L)=1.0
    1 CONTINUE
   11 CONTINUE
      L=M-NDO+1
      DO 4 I=L,M
      VS(L)=2.0
    4 CONTINUE
      L=M-NDO
      RETURN
      END
```

```
EXEC FORTRAN
   SUBROUTINE  DIAGO(VECP,VALP,N)
   DIMENSION VECP(32,32),VALP(32),A(528),R(1024)
   DOUBLE PRECISION A,R
   DOUBLE PRECISION VECP,VALP
   K=1
   DO 1 I=1,N
   DO 1 J=1,I
   A(K)=VECP(I,J)
 1 K=K+1
   CALL DEJAC(A,R,N,0)
   DO 2 I=1,N
   ID=I*(I-1)/2
   IL=ID+I
   VALP(I)=A(IL)
   DO 2 J=1,N
   JD=(I-1)*N
   JL=JD+J
 2 VECP(I,J)=R(JL)
   RETURN
   END
```

```
  EXEC FORTRAN
 DEJAC   VERSION 01 FORTRAN 30/06/67 MATH E002
      SUBROUTINE DEJAC(A,R,N,MV)
C        SUBROUTINE EIGEN                                              EIGENC
C                                                                      EIGENC
C        PURPOSE                                                       EIGENC
C           COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC    EIGENC
C           MATRIX                                                     EIGENC
C                                                                      EIGENC
C        USAGE                                                         EIGENC
C           CALL EIGEN(A,R,N,MV)                                       EIGENC
C                                                                      EIGENC
C        DESCRIPTION OF PARAMETERS                                     EIGENC
C           A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION. EIGENC
C               RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF     EIGENC
C               MATRIX A IN DESCENDING ORDER.                          EIGENC
C           R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,    EIGENC
C               IN SAME SEQUENCE AS EIGENVALUES)                       EIGENC
C           N - ORDER OF MATRICES A AND R                              EIGENC
C           MV- INPUT CODE                                             EIGENC
C                     0   COMPUTE EIGENVALUES AND EIGENVECTORS        EIGENC
C                     1   COMPUTE EIGENVALUES ONLY (R NEED NOT BE     EIGENC
C                         DIMENSIONED BUT MUST STILL APPEAR IN CALLING EIGENC
C                         SEQUENCE)                                    EIGENC
C                                                                      EIGENC
C        REMARKS                                                       EIGENC
C           ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)  EIGENC
C           MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R        EIGENC
C                                                                      EIGENC
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 EIGENC
C           NONE                                                       EIGENC
C                                                                      EIGENC
C        METHOD                                                        EIGENC
C           DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED    EIGENC
C           BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICALEIGENC
C           METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND   EIGENC
C           H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7  EIGENC
C                                                                      EIGENC
C     .....EIGENC
C                                                                      EIGENC
      DIMENSION A(1),R(1)
      DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,Z,V,SINX,SINX2,COSX,
     *                 COSX2,SINCS,YZ,YU,ZU,UW
C                                                                      EIGENC
C        .....EIGENC
C                                                                      EIGENC
```

```
C     IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE   EIGENO
C     C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION       EIGENO
C     STATEMENT WHICH FOLLOWS.                                        EIGENO
C                                                                     EIGENO
C     DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,       EIGENO
C    1                 COSX2,SINCS                                    EIGENO
C                                                                     EIGENO
C     THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS     EIGENO
C     APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS       EIGENO
C     ROUTINE.                                                        EIGENO
C                                                                     EIGENO
C     THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO       EIGENO
C     CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT IN STATEMENTSEIGENO
C     40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT.  ABS IN STATEMENT EIGENO
C     62 MUST BE CHANGED TO DABS.                                     EIGENO
C                                                                     EIGENO
C     ................................................................EIGENO
C                                                                     EIGENO
C     GENERATE IDENTITY MATRIX                                        EIGENO
C                                                                     EIGENO
      IF(MV-1) 10,25,10                                             EIGENO
   10 IQ=-N                                                         EIGENO
      DO 20 J=1,N                                                   EIGENO
      IQ=IQ+N                                                       EIGENO
      DO 20 I=1,N                                                   EIGENO
      IJ=IQ+I                                                       EIGENO
      R(IJ)=0.0                                                     EIGENO
      IF(I-J) 20,15,20                                              EIGENO
   15 R(IJ)=1.0                                                     EIGENO
   20 CONTINUE                                                      EIGENO
C
C              NORMALISATION DU PLUS GRAND TERME EXTRADIAGONAL
C
   25 Z=DABS(A(1))
      MVK=0
      DO 200 I=1,N
      K=(I*I+I)/2
      V=DABS(A(K))
      IF (Z-V) 1200,1200,200
 1200 Z=V
  200 CONTINUE
      ZU=DABS(A(2))
      DO 220 I=1,N
      DO 220 J=I,N
      IF(I-J) 215,220,215
```

```
  215 K=I+(J*J-J)/2
      V=DABS(A(K))
      IF (ZU-V) 1220,1220,220
 1220 ZU=V
  220 CONTINUE
      IF (ZU-0.D0) 1165,165,1165
 1165 Y=1.D30
      NN=(N*(N-1))/2
      NNL=NN+N
      YZ=DFLOAT(NN)
      YU=1.D37/DSQRT(YZ)
      UW=1.D-37/DSQRT(YZ)
      IF(Z-1.D-44)133,134,134
  133 IF(ZU-UW)141,142,142
  142 Z=ZU/YU
      GO TO 400
  141 MVK=MVK+1
      DO 147 K=1,NNL
  147 A(K)=A(K)*1.D50
      Z=Z*1.D50
       ZU=ZU*1.D50
      GO TO 137
  134 IF(ZU-UW)401,137,137
  401 Z=Z/Y
      GO TO 400
  137 Z=Z/Y
      ZU=ZU/YU
      IF(Z-ZU)132,400,400
  132 Z=ZU
  400 MVK=MVK+1
      DO 205 K=1,NNL
      A(K)=A(K)/Z
  205 CONTINUE
C                                                                       E
C        COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)             E
C                                                                       E
      ANORM=0.D0
      DO 35 I=1,N                                                       E
      DO 35 J=I,N                                                       E
      IF(I-J) 30,35,30                                                  E
   30 IA=I+(J*J-J)/2                                                    E
      ANORM=ANORM+A(IA)*A(IA)                                           E
   35 CONTINUE                                                          E
      IF(ANORM) 165,165,40                                              E
   40 ANORM=1.414*SQRT(ANORM)
```

```
      ANRMX=ANORM*1.0D-15/DFLOAT(N)
C                                                                       EIG
C        INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR               EIG
C                                                                       EIG
      IND=0                                                             EIG
      THR=ANORM                                                         EIG
   45 THR=THR/DFLOAT(N)
   50 LQ=0
      L=1
   55 M=L+1                                                             EIG
C                                                                       EIG
C        COMPUTE SIN AND COS                                            EIG
C                                                                       EIG
      LQ=LQ+L-1
   60 MQ=(M*M-M)/2                                                      EIG
      LM=L+MQ                                                           EIG
   62 IF(DABS(A(LM))-THR) 130,65,65
   65 IND=1                                                             EIG
      LL=L+LQ                                                           EIG
      MM=M+MQ                                                           EIG
      X=0.5*(A(LL)-A(MM))                                               EIG
   68 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75                                                    EIG
   70 Y=-Y                                                              EIG
   75 SINX=Y/DSQRT(2.0D0*(1.0D0+(DSQRT(1.0D0-Y*Y))))
      SINX2=SINX*SINX
   78 COSX=DSQRT(1.0D0-SINX2)
      COSX2=COSX*COSX
      SINCS=SINX*COSX
C        ROTATE L AND M COLUMNS                                         EIG
C                                                                       EIG
      ILQ=N*(L-1)                                                       EIG
      IMQ=N*(M-1)                                                       EIG
      DO 125 I=1,N                                                      EIG
      IQ=(I*I-I)/2                                                      EIG
      IF(I-L) 80,115,80                                                 EIG
   80 IF(I-M) 85,115,90                                                 EIG
   85 IM=I+MQ                                                           EIG
      GO TO 95                                                          EIG
   90 IM=M+IQ                                                           EIG
   95 IF(I-L) 100,105,105                                               EIG
  100 IL=I+LQ                                                           EIG
      GO TO 110                                                         EI
  105 IL=L+IQ                                                           EI
  110 X=A(IL)*COSX-A(IM)*SINX                                           EI
```

```
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
  115 IF(MV-1) 120,125,120
  120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
  125 CONTINUE
      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C
C        TESTS FOR COMPLETION
C
C        TEST FOR M = LAST COLUMN
C
  130 IF(M-N) 135,140,135
  135 M=M+1
      GO TO 60
C
C        TEST FOR L = SECOND FROM LAST COLUMN
C
  140 IF(L-(N-1)) 145,150,145
  145 L=L+1
      GO TO 55
  150 IF(IND-1) 160,155,160
  155 IND=0
      GO TO 50
C
C        COMPARE THRESHOLD WITH FINAL NORM
C
  160 IF(THR-ANRMX) 165,165,45
C
C        SORT EIGENVALUES AND EIGENVECTORS
C
  165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=(I*I+I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
```

```
      JQ=JQ+N
      MM=(J*J+J)/2
      IF(A(LL)-A(MM)) 170,185,185
  170 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
      IF(MV-1) 175,185,175
  175 DO 180 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
      IF (MVK) 1450,450,1450
 1450 IF (MVK-1) 1460,460,1460
 1460 DO 360 K=1,NNL
  360 A(K)=A(K)*1.D-50
  460 DO 350 K=1,NNL
      DO 350 K=1,NNL
  350 A(K)=A(K)*Z
  450 RETURN
      END
// EXEC LNKEDT
// ASSGN SYS001,X'192'
// ASSGN SYS002,X'192'
// ASSGN SYS003,X'191'
// ASSGN SYS004,X'192'
// ASSGN SYS005,X'192'
// ASSGN SYS006,X'192'
// DLBL IJSYS03,'COP1 DOUADY RECOUVREMENT',69/365
// EXTENT ,9,,,1960,20
// DLBL IJSYS04,,1
// EXTENT ,6,,,1100,200
// DLBL IJSYS05,,1
// EXTENT ,6,,1300,200
// DLBL IJSYS06,,1
// EXTENT ,6,,1500,200
// EXEC
// OPTION LINK
```

PROGRAMME

POPULATION DE RECOUVREMENT

```
// EXEC FORTRAN
      DIMENSION S(32,32),RAN(16,16),VS(32),C(32),CC(32,32)
      DIMENSION CCS(32,32),VALP(32),NOA(16),ZM(32,32)
      DIMENSION SO(256)
      DOUBLE PRECISION S,RAN,VS,C,CC,CCS,VALP,ZM
      DOUBLE PRECISION SO
      DOUBLE PRECISION SOM,SOM1
      DOUBLE PRECISION SOM2,SOM3
      DOUBLE PRECISION ZNOR,VNOR,CNOR
 1001 FORMAT(I4)
 1002 FORMAT('0LA DIMENSION DE LA MATRICE S EST',I4)
 1003 FORMAT('0LE NOMBRE D ATOMES EST',I4)
 1004 FORMAT(6I5)
 1005 FORMAT('0NOMBRE D ELECTRONS',I5,10X,'DOUBLEMENT OCCUPE'I5,10X,
     1'SIMPLEMENT OCCUPES',I5)
 1006 FORMAT(4F20.14)
 1007 FORMAT(F20.14)
 1008 FORMAT(4F20.14)
 1009 FORMAT('0ORBITALE NUMERO',I3,20X,'OVERLAP POPULATION PARTIELLE PAR
     1 O.M.')
 1010 FORMAT('0ATOME',I3,5X,'ATOME',I3,20X,F20.10)
 1011 FORMAT('0OVERLAP POPULATION TOTALE POUR L O.M.NUMERO',I3,5X,F20.10
     1)
 1012 FORMAT('0OVERLAP POPULATION TOTALE PAR COUPLE')
 1013 FORMAT('0POUR LE COUPLE',I3,I3,10X,F20.10)
 1014 FORMAT('0OVERLAP POPULATION TOTALE SOMMEE PAR LIGNE',F20.10)
 1015 FORMAT(1H0,20F5.1)
 1016 FORMAT('0MATRICE S SYMETRIQUE')
 1017 FORMAT('0VECTEUR MULTIPLICAREUR')
 1018 FORMAT(1H0,F5.1)
 1019 FORMAT('0ORBITALE NUMERO',I3)
 1020 FORMAT('0 ATOME ',I3,10X,'ATOME',I3,5X,'ELEMENT CALCULE',F15.8)
 1021 FORMAT(1H0)
 1022 FORMAT(1H1)
 1023 FORMAT('0ENERGIE ASSOCIEE A L ORBITALE',F15.8)
 1024 FORMAT('0VECTEUR PROPRE          VECTEUR PROPRE NORME')
 1025 FORMAT(F15.8,10X,F15.8)
 1027 FORMAT(1H0,10F10.4)
 1029 FORMAT('0OVERLAP POPULATION TOTALE SOMMEE PAR COLONNE',F20.10)
      CALL PREDIS(IDISC)
      REWIND IDISC
      READ(IDISC)LEC,IPERF,IMP,IDISC,JDISC,KDISC,LDISC,MDISC,NDISC
      SOM4=0.0
      K=0
      READ(IDISC,1001)M
      WRITE(IMP,1002)M
```

```
      READ(IDISC,1006)((S(I,J),I=1,J),J=1,M)
      DO 301 J=1,M
      DO 301 I=1,J
  301 S(J,I)=S(I,J)
      READ(IDISC,1001)NAT
      WRITE(IMP,1003)NAT
      READ(IDISC,1004)NELEC,NSO,NDO
      WRITE(IMP,1005)NELEC,NDO,NSO
      READ(IDISC,1004)(NOA(I),I=1,NAT)
      NOR=NDO+NSO
C   CONSTRUCTION DE MULTIPLICATEUT VS
      DO 1 I=1,M
    1 VS(I)=0.0
      DO 2 I=1,NDO
    2 VS(I)=2.0
      IF(NSO-1)3,4,4
    4 DO 5 I=1,NSO
      K1=NDO+I
    5 VS(K1)=1.0
    3 CONTINUE
      WRITE(IMP,1017)
      DO 303 I=1,M
  303 WRITE(IMP,1018)VS(I)
      WRITE(IMP,1022)
      SOM3=0.0
      NNA=NAT-1
      DO 304 I=1,225
  304 SO(I)=0.0
      DO 200 IOR=1,M
      WRITE(IMP,1019)IOR
      K=K+1
      READ(IDISC,1007)VALP(IOR)
      DO 9785 I=1,M
      READ(IDISC,1008)C(I)
 9785 CONTINUE
      ZNOR=0.0
      DO 210 I=1,M
      DO 210 J=1,M
      CC(I,J)=C(I)*C(J)
      ZM(I,J)=CC(I,J)*S(I,J)
      ZNOR=ZNOR+ZM(I,J)
  210 CONTINUE
      VNOR=DSQRT(ZNOR)
      WRITE(IMP,1023)VALP(IOR)
      WRITE(IMP,1024)
```

```
      DO 211 I=1,M
      CNOR=C(I)/VNOR
  211 WRITE(IMP,1025)C(I),CNOR
      WRITE(IMP,1021)
      WRITE(IMP,1021)
      IF(IOR-NOR)3500,3500,3501
 3500 CONTINUE
      DO 10 I=1,NNA
      N1=NOA(I)
      IF(I-1)11,11,12
   11 IV=1
      GO TO 13
   12 IV=IV+NOA(I-1)
   13 JV=IV
      KK=I+1
      DO 10 J=KK,NAT
      N2=NOA(J)
      JV=JV+NOA(J-1)
      IF(N1*N2-16)14,15,14
   15 SOM=0.0
      DO 100 II=1,4
      IZM=IV+II-1
      DO 100 JJ=1,4
      JZM=JV+JJ-1
  100 SOM=SOM+ZM(IZM,JZM)
      GO TO 305
   14 IF(N1*N2-4)17,18,17
   18 SOM=0.0
      IF(N1-1)20,19,20
   20 DO 101 II=1,4
      IZM=IV+II-1
  101 SOM=SOM+ZM(IZM,JV)
      GO TO 305
   19 DO 102 JJ=1,4
      JZM=JV+JJ-1
  102 SOM=SOM+ZM(IV,JZM)
      GO TO 305
   17 SOM=ZM(IV,JV)
  305 RAN(I,J)=SOM
   10 CONTINUE
      L=0
      DO 300 I=1,NNA
      JJ=I+1
      DO 300 J=JJ,NAT
      L=L+1
```

```
      RAN(I,J)=RAN(I,J)*2.0*VS(K)
  300 SO(L)=SO(L)+RAN(I,J)
      SOM2=0.0
      WRITE(IMP,1009) IOR
      DO 400 I=1,NNA
      JJ=I+1
      DO 400 J=JJ,NAT
      SOM2=SOM2+RAN(I,J)
  400 WRITE(IMP,1010)I,J,RAN(I,J)
      WRITE(IMP,1021)
      SOM3=SOM3+SOM2
      WRITE(IMP,1011)IOR,SOM2
      WRITE(IMP,1021)
      WRITE(IMP,1021)
 3501 CONTINUE
      WRITE(IMP,1022)
  200 CONTINUE
      L=0
      WRITE(IMP,1012)
      DO 500 I=1,NNA
      JJ=I+1
      DO 500 J=JJ,NAT
      L=L+1
      SOM4=SOM4+SO(L)
  500 WRITE(IMP,1013)I,J,SO(L)
      WRITE(IMP,1021)
      WRITE(IMP,1014)SOM3
      WRITE(IMP,1021)
      WRITE(IMP,1029)SOM4
      STOP
      END
```

PROGRAMME

INTEGRALES DE RECOUVREMENT

INTEGRALES DE MOMENT DIPOLAIRE

```
// EXEC FORTRAN
      DEFINE FILE 5(100,70,U,N2)
      DEFINE FILE 6(2100,70,U,N3)
      DEFINE FILE 7(2100,70,U,N4)
      DEFINE FILE 8(1100,70,U,N5)
      DEFINE FILE 9(150,70,U,N6)
      DEFINE FILE 10(100,70,U,N7)
      DEFINE FILE 11(100,70,U,N8)
      DEFINE FILE 12(150,70,U,N9)
      DEFINE FILE 13(100,70,U,N1)
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
      COMMON TYPE
      COMMON NAT,BASE
      COMMON NDIM
      NDIM=32
      LEC=1
      IMP=3
      IB4=13
      IB5=5
      IB6=6
      IB7=7
      IB8=8
      IB9=9
      IB10=10
      IB11=11
      IB12=12
      IB13=13
      CALL LECTUR
      CALL PART1
      CALL PART2
      END
```

```
// EXEC FORTRAN
      SUBROUTINE LECTUR
      INTEGER TYPE
      INTEGER TYPATO,TYPATN,TYPATC,TYPATH
      DIMENSION MATO(4,5),MATN(4,5),MATC(4,5),MATH(4,5)
      DIMENSION COEFO(2),COEFN(2),COEFC(2),COEFH(2)
      DIMENSION C(8,8)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DOUBLE PRECISION HZ,CZ,ZN,OZ,HW,CW,WN,OW
      DOUBLE PRECISION MATO,MATN,MATC,MATH
      DOUBLE PRECISION COEFO,COEFN,COEFC,COEFH
      DOUBLE PRECISION C,FC,FC1,FC2
      DOUBLE PRECISION S
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
      COMMON TYPE
      COMMON NAT,BASE
      COMMON NDIM
 1001 FORMAT(I5)
 1002 FORMAT('ONOMBRE DE TYPES ',I5)
 1003 FORMAT(4I5,F15.8)
 1004 FORMAT(1HO,5I5,F4.1)
 1005 FORMAT(5I5,5X,3F10.5)
 1006 FORMAT(1HO,5I5,5X,3F15.8)
 1007 FORMAT(8F10.1)
 1008 FORMAT(1HO,8F10.1)
 1010 FORMAT(1HO,I2,5I5)
 1011 FORMAT(1HO,30X,3F10.5)
 1101 FORMAT(1HO,8X,'IZ',3X,'JZ',3X,'IS',3X,'JS',1X,'SIGNE')
 1102 FORMAT(  1HO,'NTYP',1X,'IMAX',3X,'IA',3X,'IB',3X,'IC',7X,'FC',
     19X,'FC1',9X,'FC2')
 1103 FORMAT('OMATRICE DES COEFFICIENTS DES A(I) ET DES B(I) ')
 5510 FORMAT(4F10.5)
 5511 FORMAT(F10.5)
 5512 FORMAT('ODONNEES DE L OXYGENE  TYPE = ',I2)
 5513 FORMAT(1HO,15X,'MATRICE Z ',25X,'VECTEUR W ')
 5514 FORMAT('ODONNEES DE L AZOTE    TYPE = ',I2)
 5515 FORMAT('ODONNEES DU   CARBONE  TYPE = ',I2)
 5516 FORMAT('ODONNEES DE L HYDROGENE  TYPE= ',I2)
 5517 FORMAT(1HO,4F10.3,10X,F10.3)
 5518 FORMAT(1HO,F10.3,50X,F10.3)
 5550 FORMAT('OC1S= ',F15.8,'   C2S = ',F15.8)
C  IZ ET JZ  SERVENT A REPERER LA PLACE DES DZETAS
C  IS ET JS SONT LES TYPES D'OA RESPECTIVEMENT DE ',ATOME A ET DE L'ATOME
C  S EST UN SIGNE AFFECTE AU RECOUVREMENT
C  NTYP EST = O SI L'INTEGRALE DE RECOUVREMENT EST NULLE
```

```
C  NTYP EST = 1 SINON
C IMAX EST LA DIMENSION DE LA MATRICE DES COEFFICIENTS NECESSAIRES POUR
C  CALCULER LES  AI ET LES BI
C   IA,IB,IC  FC, SERVENT A CALCULER L'EXPRESSION
C  S=((1.0+TO)**(IA/2))*((1.0-TO)**(IB/2))*(RO)**IC)*(1.0/FC)
      N1=1
      IB1=1
      READ(LEC,1001)TYPE
      WRITE(IMP,1002)TYPE
      DO 100 ITYPE=1,TYPE
      READ(LEC,1003)IZ,JZ,IS,JS,S
      WRITE(IB4'N1)IZ,JZ,IS,JS,S
  100 CONTINUE
      DO 200 ITYPE =1,TYPE
      READ(LEC,1005) NTYP,IMAX,IA,IB,IC,FC,FC1,FC2
      WRITE(IB4'N1)NTYP,IMAX,IA,IB,IC,FC,FC1,FC2
      DO 300 I=1,IMAX
       READ(LEC,1007)(C(I,J),J=1,IMAX)
  300 CONTINUE
      DO 600 KC=1,IMAX
  600 WRITE(IB4'N1)(C(KC,JC),JC=1,IMAX)
  200 CONTINUE
      READ(LEC,1001)TYPATO
      READ(LEC,1007)COEFO
      READ(LEC,1001)TYPATN
      READ(LEC,1007)COEFN
      READ(LEC,1001)TYPATC
      READ(LEC,1007)COEFC
      READ(LEC,1001)TYPATH
      READ(LEC,1007)COEFH
      DO 7000 I=1,4
      READ(LEC,5510)(OZ(I,J),J=1,4)
 7000 CONTINUE
      DO 7001 I=1,4
      READ(LEC,5510)(ZN(I,J),J=1,4)
 7001 CONTINUE
      DO 7002 I=1,4
      READ(LEC,5510)(CZ(I,J),J=1,4)
 7002 CONTINUE
      READ(LEC,5511)HZ
      READ(LEC,5510)OW
      READ(LEC,5510)WN
      READ(LEC,5510)CW
      READ(LEC,5511)HW
      CALL GENMAT(COEFO,MATO)
```

```
      CALL GENMAT(COEFN,MATN)
      CALL GENMAT(COEFC,MATC)
      CALL GENMAT(COEFH,MATH)
      N8=1
      WRITE(IB11'N8)OZ
      WRITE(IB11'N8)ZN
      WRITE(IB11'N8)CZ
      WRITE(IB11'N8)HZ
      WRITE(IB11'N8)OW
      WRITE(IB11'N8)WN
      WRITE(IB11'N8)CW
      WRITE(IB11'N8)HW
      N7=1
      WRITE(IB10'N7)TYPATO,TYPATN,TYPATC,TYPATH
      WRITE(IB10'N7)MATO
      WRITE(IB10'N7)MATN
      WRITE(IB10'N7)MATC
      WRITE(IB10'N7)MATH
      N8=1
      READ(IB11'N8)OZ
      READ(IB11'N8)ZN
      READ(IB11'N8)CZ
      READ(IB11'N8)HZ
      READ(IB11'N8)OW
      READ(IB11'N8)WN
      READ(IB11'N8)CW
      READ(IB11'N8)HW
      N7=1
      READ(IB10'N7)TYPATO,TYPATN,TYPATC,TYPATH
      READ(IB10'N7)MATO
      READ(IB10'N7)MATN
      READ(IB10'N7)MATC
      READ(IB10'N7)MATH
      WRITE(IMP,5512)TYPATO
      WRITE(IMP,5550)COEFO
      WRITE(IMP,5513)
      DO 6000 I=1,4
6000  WRITE(IMP,5517)(OZ(I,J),J=1,4),OW(I)
      WRITE(IMP,5514)TYPATN
      WRITE(IMP,5550)COEFN
      WRITE(IMP,5513)
      DO 6001 I=1,4
6001  WRITE(IMP,5517)(ZN(I,J),J=1,4),WN(I)
      WRITE(IMP,5515)TYPATC
      WRITE(IMP,5550)COEFC
```

```
      WRITE(IMP,5513)
      DO 6002 I=1,4
6002  WRITE(IMP,5517)(CZ(I,J),J=1,4),CW(I)
      WRITE(IMP,5516)TYPATH
      WRITE(IMP,5550)COEFH
      WRITE(IMP,5513)
6003  WRITE(IMP,5518)HZ,HW
      N1=1
      WRITE(IMP,1101)
      DO 400 ITYPE=1,TYPE
      READ(IB4'N1)IZ,JZ,IS,JS,S
      WRITE(IMP,1004)ITYPE,IZ,JZ,IS,JS,S
 400  CONTINUE
      WRITE(IMP,1102)
      DO 500 ITYPE=1,TYPE
      READ(IB4'N1)NTYPE,IMAX,IA,IB,IC,FC,FC1,FC2
      DO 700 KC=1,IMAX
 700   READ(IB4'N1)(C(KC,JC),JC=1,IMAX)
      WRITE(IMP,1010)ITYPE,NTYPE,IMAX,IA,IB,IC
      WRITE(IMP,1011)FC,FC1,FC2
      WRITE(IMP,1103)
      DO 501 I=1,IMAX
      WRITE(IMP,1008)(C(I,J),J=1,IMAX)
 501  CONTINUE
 500  CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE PART1
      INTEGER BASE ,BI,ESP,TYPE,CALC
      INTEGER SAVE,SAVE1
      INTEGER TYPATO,TYPATN,TYPATC,TYPATH
      DIMENSION X(20),Y(20),Z(20),ZET(3,20),C(8,8),A(8),B(8)
      DIMENSION ESP(20)
      DOUBLE PRECISION C
      DOUBLE PRECISION X,Y,Z,ZET,A,B
      DOUBLE PRECISION S,R,ZETA,ZETB,ZETC,FC,FC1,FC2,P,PT,PT2,VAL
      DOUBLE PRECISION VAL1,VAL2,D1,D2,D3
      DOUBLE PRECISION T,EI1,E
      DOUBLE PRECISION DENOM
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
      COMMON TYPE
      COMMON NAT,BASE
      COMMON NDIM
 1001 FORMAT(2I5)
 1002 FORMAT(3F10.7,2I5,3F10.0)
 1003 FORMAT('ONOMBRE D ATOMES ',I3,10X,'BASE ',I5)
 1005 FORMAT(I5,3F15.8)
 1006 FORMAT(2I5,6F15.8)
 1008 FORMAT(4I5,F15.8)
 1009 FORMAT(2I5,F15.8)
 1010 FORMAT(2I5,3F15.8)
 1011 FORMAT(I5)
 1012 FORMAT(2I5,F15.8,I5)
 1013 FORMAT(4I5,F10.1)
 1014 FORMAT(2I5,3F15.8)
 1015 FORMAT(2I5,2F15.8)
 2004 FORMAT(1HO)
 2005 FORMAT(5I5,3F15.8)
 2007 FORMAT(8F10.1)
      DO 2 I=1,20
      DO 2 J=1,3
    2 ZET(J,I)=0.0
      N7=1
      READ(IB10'N7)TYPATO,TYPATN,TYPATC,TYPATH
      READ(LEC,1001) NAT,BASE
      READ(LEC,1002)(X(I),Y(I),Z(I),ILE,ESP(I),(ZET(J,I),J=1,3),I=1,NAT)
      WRITE(IMP,1003)NAT,BASE
      WRITE(IMP,2004)
      N2=1
      N9=1
      DO 15 I=1,NAT
```

```
      BI=1
      IF(ESP(I)-TYPATH)7,9,7
    7 BI=5
    9 CONTINUE
      WRITE(IB12'N9)ESP(I)
      WRITE(IB5'N2)ESP(I),BI,X(I),Y(I),Z(I)
      JM=3
      IF(ESP(I)-TYPATH)15,10,15
   10 JM=1
   15 CONTINUE
      WRITE(IMP,2004)
      WRITE(IMP,2004)
      N1=1
      N3=1
      DO 600 ITYPE=1,TYPE
      READ(IB4'N1)IZ,JZ,IS,JS,S
      LIM=0
      WRITE(IB6'N3)LIM,IS,JS,S
      SAVE=N3-1
      DO 103 I=1,NAT
      DO 103 J=1,I
      R=DSQRT((X(I)-X(J))**2 +(Y(I)-Y(J))**2 +(Z(I)-Z(J))**2)
      R=R/0.529167
      ZETA=ZET(IZ,I)
      ZETB=ZET(JZ,J)
      IF(ZETA*ZETB-1.D-6)30,30,20
   20 WRITE(IB6'N3)I,J,ZETA,ZETB,R
      LIM=LIM+1
   30 IF(IS-JS)33,103,33
   33 IF(I-J)37,103,37
   37 ZETA=ZET(JZ,I)
      ZETB=ZET(IZ,J)
      IF(ZETA*ZETB-1.D-6)103,103,43
   43 WRITE(IB6'N3)J,I,ZETB,ZETA,R
      LIM=LIM+1
  103 CONTINUE
      SAVE1=N3
      N3=SAVE
      WRITE(IB6'N3)LIM,IS,JS,S
      N3=SAVE1
  600 CONTINUE
      N3=1
      N4=1
      DO 500  ITYPE=1,TYPE
      READ(IB4'N1)NTYPE,IMAX,IA,IB,IC,FC,FC1,FC2
```

```
      DO 800 KC=1,IMAX
  800 READ(IB4'N1)(C(KC,JC),JC=1,IMAX)
      IMAX1=IMAX+1
      READ(IB6'N3)LIM,IS,JS,S
      WRITE(IB7'N4)LIM,IS,JS,NTYPE,S
      IF(LIM)500,500,3
    3 DO 450 CALC=1,LIM
      READ(IB6'N3)JA,JB,ZETA,ZETB,R
      VAL1=0.0
      VAL2=0.0
      IF(R-1.D-6)110,110,5
    5 P=(ZETA+ZETB)*0.5*R
      T=(ZETA-ZETB)/(ZETA+ZETB)
      PT=P*T
      PT2=PT*PT
      S=DSQRT((1.0+T)**IA*(1.0-T)**IB)*P**IC/FC
      A(1)=DEXP(-P)/P
      DO 35 I=2,IMAX1
      I1=I-1
      EI1=DFLOAT(I1)
      A(I)=(DEXP(-P)+EI1*A(I-1))/P
   35 CONTINUE
      DO 70 I=1,IMAX1,2
      E=DFLOAT(I)
      VAL=1.0/E
      D1=1.0
      D3=0.0
   40 D3=D3+2.0
      D1=D1*PT2/(D3*(D3-1.0))
      D2=D1/(E+D3)
      IF(D2/VAL-1.D-8)50,50,45
   45 VAL=VAL+D2
      GO TO 40
   50 B(I)=2.0*VAL
      IF(I-IMAX1)55,73,73
   55 E=DFLOAT(I)+1.0
      VAL=-PT/(E+1.0)
      D1=-PT
      D3=1.0
   60 D3=D3+2.0
      D1=D1*PT2/(D3*(D3-1.0))
      D2=D1/(E+D3)
      IF(DABS(T)-1.D-8)7201,7201,7200
 7200 CONTINUE
      IF(D2/VAL-1.D-8)70,70,65
```

```
   65 VAL=VAL+D2
      GO TO 60
 7201 CONTINUE
   70 B(I+1)=2.0*VAL
   73 DO 80 I=1,IMAX
      VAL=0.0
      DO 75 J=1,IMAX
   75 VAL=VAL+C(I,J)*B(J)
   80 VAL1=VAL1+VAL*A(I)
      IF(NTYPE)85,85,90
   85 VAL2=VAL1
      VAL1=0.0
      GO TO 105
   90 DO 100 I=1,IMAX
      I1=I+1
      VAL=0.0
      DO 95 J=1,IMAX
      J1=J+1
   95 VAL=VAL+C(I,J)*B(J1)
  100 VAL2=VAL2+VAL*A(I1)
      VAL1=VAL1*S
  105 VAL2=VAL2*R*S*0.5
      GO TO 200
  110 ZETC=(ZETA+ZETB)*0.5
      S=DSQRT(ZETA**IA*ZETB**IB)/ZETC**IC
      IF(FC1-1.D-6)150,150,115
  115 VAL1=FC1*S
  150 IF(FC2-1.D-6)200,200,155
  155 VAL2=FC2*S/ZETC
  200 WRITE(IB7'N4)JA,JB,VAL1,VAL2
  450 CONTINUE
  500 CONTINUE
      RETURN
      END
```

```
C FORTRAN
      SUBROUTINE PART2
      DIMENSION DNOU(4,4,4)
      DIMENSION MAT1(4,5),MAT2(4,5),MATTRA(5,4)
      DIMENSION S(5,5),X(5,5),Y(5,5),Z(5,5),U(5,5),A(5,5),D(5,5,4)
      DIMENSION B(32,32)
      DOUBLE PRECISION DNOU
      DOUBLE PRECISION MAT1,MAT2,MATTRA
      DOUBLE PRECISION S,X,Y,Z,U,A,D,B
      DOUBLE PRECISION XI,YI,ZI,XJ,YJ,ZJ,SIGNE,VAL,VAL1,VAL2
      DOUBLE PRECISION XIJ,YIJ,ZIJ,X1,Y1,Z1,CONV
      EQUIVALENCE (D(1),X(1)),(D(26),Y(1)),(D(51),Z(1)),(D(76),S(1))
      EQUIVALENCE (B(1),A(1)),(B(26),U(1))
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
      COMMON TYPE
      COMMON NAT,BASE
      COMMON NDIM
      INTEGER BASRED
      INTEGER ESPI,ESPJ
      INTEGER BIB,BJB
      INTEGER TYPATO,TYPATN,TYPATC,TYPATH
      INTEGER CALC,TYPE,BI,BJ,BASE
      INTEGER SAVE
 1001 FORMAT(I5,3F15.8)
 2000 FORMAT('0MATRICE X')
 2001 FORMAT('0MATRICE Y')
 2002 FORMAT('0MATRICE Z')
 2003 FORMAT('0 MATRICE S')
 2004 FORMAT(1H0)
 1010 FORMAT(1H0,7F15.8)
 2010 FORMAT(2I5,2F15.8)
 2013 FORMAT(4I5,F10.1)
       N7=1
      READ(IB10'N7)TYPATO,TYPATN,TYPATC,TYPATH
      CONV=0.944881294
      SAVE=1
      N5=1
      DO 140 I=1,NAT
      N2=SAVE
      READ(IB5'N2)ESPI,BI,XI,YI,ZI
      SAVE=N2
      WRITE(IB8'N5)ESPI,BI
      N2=1
      DO 130 J=1,I
      READ(IB5'N2)ESPJ,BJ,XJ,YJ,ZJ
```

```
      DO 5 K=1,5
      DO 5 L=1,5
      U(K,L)=0.0
      S(K,L)=0.0
      X(K,L)=0.0
      Y(K,L)=0.0
      Z(K,L)=0.0
    5 CONTINUE
      U(1,1)=1.0
      U(2,2)=1.0
      N4=1
      DO 55 ITYPE=1,TYPE
      READ(IB7'N4)LIM,I1,J1,NTYPE,SIGNE
      IF(LIM)10,55,10
   10 DO  50 CALC=1,LIM
      READ(IB7'N4)I2,J2,VAL1,VAL2
      IF(IABS(I-I2)+ IABS(J-J2))15,15,30
   15 IF(NTYPE)20,25,20
   20 S(I1,J1)=VAL1
      Z(I1,J1)=VAL2
      GO TO 50
   25 X(I1,J1)=VAL2
      GO TO 50
   30 IF(IABS(J-I2)+IABS(I-J2))35,35,50
   35 IF(NTYPE)40,45,40
   40 S(J1,I1)=VAL1*SIGNE
      Z(J1,I1)=VAL2*(-SIGNE)
      GO TO 50
   45 X(J1,I1)=VAL2
   50 CONTINUE
   55 CONTINUE
      Y(2,5)=X(2,4)
      Y(1,5)=X(1,4)
      Y(3,5)=X(3,4)
      Y(5,1)=X(4,1)
      Y(5,2)=X(4,2)
      Y(5,3)=X(4,3)
      Z(5,5)=Z(4,4)
      S(5,5)=S(4,4)
      IF(I-J)70,60,70
   70 XIJ=XJ-XI
      YIJ=YJ-YI
      ZIJ=ZJ-ZI
      IF(DABS(XIJ)+DABS(YIJ)-1.D-6)72,72,75
   72 IF(ZIJ)73,74,74
```

```
C    CAS  MONOCENTRIQUE SYMETRISATION DE X,Y,Z,S
   60 DO 65 K=1,4
      DO 65 L=1,5
      DO 64 M=1,L
   64 D(L,M,K)=D(M,L,K)
   65 S(L,L)=1.0
      GO TO 74
C  CAS OU LA LIGNE DES ATOMES EST PARALLELE A OZ
C ET ZJINFERIEUR OU EGAL A ZI
   73 U(1,1)=1.0
      U(2,2)=1.0
      U(5,3)=-1.0
      U(4,5)=-1.0
      U(3,4)=1.0
      GO TO 87
C  CAS OU LA LIGNE DES ATOMES EST PARALLELE A OZ
C    ZJ SUPERIEUR A ZI
   74 U(1,1)=1.0
      U(2,2)=1.0
      U(5,3)=1.0
      U(4,5)=1.0
      U(3,4)=1.0
      GO TO 87
C    CAS GENERAL
   75 U(3,3)=XIJ
      U(4,3)=YIJ
      U(5,3)=ZIJ
      U(3,4)=-YIJ
      U(4,4)=XIJ
      U(5,4)=0.0
      U(3,5)=-XIJ*ZIJ
      U(4,5)=-YIJ*ZIJ
      U(5,5)=XIJ*XIJ+YIJ*YIJ
      DO 85 K=3,5
      VAL=0.0
      DO 80 L=3,5
   80 VAL=VAL+U(L,K)**2
      VAL=1.0/(DSQRT(VAL))
      DO 85 L=3,5
   85 U(L,K)=U(L,K)*VAL
   87 DO 90 K=1,5
      DO 90 L=1,5
      X1=X(K,L)
      Y1=Y(K,L)
      Z1=Z(K,L)
```

```
      X(K,L)=X1*U(3,4)+Y1*U(3,5)+Z1*U(3,3)
      Y(K,L)=X1*U(4,4)+Y1*U(4,5)+Z1*U(4,3)
   90 Z(K,L)=X1*U(5,4)+Y1*U(5,5)+Z1*U(5,3)
      DO 110 K=1,4
      DO 100 L=1,5
      DO 100 M=1,5
      VAL=0.0
      DO 95 N=1,5
   95 VAL=VAL+D(L,N,K)*U(M,N)
  100 A(L,M)=VAL
      DO 110 L=1,5
      DO 110 M=1,5
      VAL=0.0
      DO 105 N=1,5
  105 VAL=VAL+U(L,N)*A(N,M)
  110 D(L,M,K)=VAL
  115 XIJ=(XI+XJ)*CONV
      YIJ=(YI+YJ)*CONV
      ZIJ=(ZI+ZJ)*CONV
      DO 120 K=1,5
      DO 120 L=1,5
      VAL=S(K,L)
      X(K,L)=VAL*XIJ+X(K,L)
      Y(K,L)=VAL*YIJ+Y(K,L)
  120 Z(K,L)=VAL*ZIJ+Z(K,L)
      WRITE(IB8'N5)ESPJ,BJ
      DO 700 ID=1,5
  700 WRITE(IB8'N5)((D(ID,JD,KD),JD=1,5),KD=1,4)
  130 CONTINUE
  140 CONTINUE
      CALL ARANG(NAT,BASRED)
      N6=1
      DO 540 K=1,4
      N5=1
      LMAX=0
      DO 520 I=1,NAT
      READ(IB8'N5)ESPI,BI
      IF(BI-1)6000,6001,6002
 6001 BIB=BI
      GO TO 6000
 6002 BIB=BI-1
 6000 CONTINUE
      LMIN=LMAX+1
      LMAX=LMAX+BIB
      MAX=0
```

```
      DO 520 J=1,I
      READ(IB8'N5)ESPJ,BJ
      CALL OBMAT(ESPI,MAT1)
      CALL OBMAT(ESPJ,MAT2)
      CALL TRANSP(MAT2,MATTRA)
      IF(BJ-1)6003,6004,6005
6004  BJB=BJ
      GO TO 6003
6005  BJB=BJ-1
6003  CONTINUE
      DO 800 ID=1,5
 800   READ(IB8'N5)((D(ID,JD,KD),JD=1,5),KD=1,4)
      CALL REDUC(MAT1,D,MATTRA,DNOU)
      MIN=MAX+1
      MAX=MAX+BJB
      L=LMIN-1
      DO 520 L1=1,BIB
      L=L+1
      M=MIN-1
      MAX1=BJB
      IF(LMAX-MAX)500,500,510
 500  MAX1=BIB
 510  DO 520 M1=1,MAX1
      M=M+1
      VAL=DNOU(L1,M1,K)
      B(L,M)=VAL
 520  B(M,L)=VAL
      IF(K-1)200,200,201
 200  WRITE(IMP,2000)
      GO TO 206
 201  IF(K-2)202,202,203
 202  WRITE(IMP,2001)
      GO TO 206
 203  IF(K-3)204,204,205
 204  WRITE(IMP,2002)
      GO TO 206
 205  WRITE(IMP,2003)
 206  CONTINUE
      DO 535 L=1,BASRED
 535  WRITE(IMP,1010)(B(L,M),M=1,L)
      DO 536 L=1,NDIM
 536  WRITE(IB9'N6)(B(L,M),M=1,NDIM)
 540  CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE REPART(ZDEZ,WDEW,IDEZ,JDEZ,IDEW,I,TYPATO,TYPATN,TYPATC,
     1 TYPATH,OZ,ZN,CZ,HZ,OW,WN,CW,HW)
      DIMENSION ZDEZ(32,32),WDEW(32)
      DIMENSION OW(4),CW(4),WN(4),OZ(4,4),CZ(4,4),ZN(4,4)
      DOUBLE PRECISION ZDEZ,WDEW,HZ,CZ,ZN,OZ,HW,CW,WN,OW
      INTEGER TYPATO,TYPATN,TYPATC,TYPATH
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
 4000 FORMAT('OLE TYPE I= ',I2,' NE CORRESPOND PAS A LA NUMEROTATION ')
 9999 FORMAT('0 Z ET W  CREES  POUR L ATOME DE TYPE ',I2)
      IF(I-TYPATH)3018,3003,3018
 3003 ZDEZ(IDEZ,IDEZ)=HZ
      WDEW(IDEZ)=HW
      IDEZ=IDEZ+1
      WRITE(IMP,9999)I
 3004 GO TO 5000
 3018 IF(I-TYPATC)3031,3030,3031
 3030 DO 3036 III=1,4
      IIDEZ=IDEZ+III-1
      DO 3036 JJJ=1,4
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=CW(III)
 3036 ZDEZ(IIDEZ,JJDEZ)=CZ(III,JJJ)
      GO TO 3035
 3031 IF(I-TYPATN)3033,3032,3033
 3032 DO 3037 III=1,4
      IIDEZ=IDEZ+III-1
      DO 3037 JJJ=1,4
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=WN(III)
 3037 ZDEZ(IIDEZ,JJDEZ)=ZN(III,JJJ)
      GO TO 3035
 3033 IF(I-TYPATO)3002,3034,3002
 3002 WRITE(IMP,4000)I
      GO TO 5000
 3034 DO 3038 III=1,4
      IIDEZ=IDEZ+III-1
      DO 3038 JJJ=1,4
      JJDEZ=IDEZ+JJJ-1
      WDEW(IIDEZ)=OW(III)
 3038 ZDEZ(IIDEZ,JJDEZ)=OZ(III,JJJ)
 3035 IDEZ=IDEZ+4
      WRITE(IMP,9999)I
 5000 CONTINUE
      RETURN
```

END

```
// EXEC FORTRAN
      SUBROUTINE OBMAT(ESP,MAT)
      DIMENSION MAT(4,5),MATO(4,5),MATN(4,5),MATC(4,5),MATH(4,5)
      DOUBLE PRECISION MAT,MATO,MATC,MATN,MATH
      INTEGER ESP
      INTEGER TYPATO,TYPATN,TYPATC,TYPATH
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
 1001 FORMAT('OLE TYPE I= ',I2,' NE CORRESPOND PAS A LA NUMEROTATION ')
       N7=1
      READ(IB10'N7)TYPATO,TYPATN,TYPATC,TYPATH
      STOP
      END
```

```
// EXEC FORTRAN
      SUBROUTINE TRANSP(MAT,MAT1)
      DIMENSION MAT(4,5),MAT1(5,4)
      DOUBLE PRECISION MAT,MAT1
      DO 1 I=1,4
      DO 1 J=1,5
      MAT1(J,I)=MAT(I,J)
    1 CONTINUE
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE REDUC(MAT1,D,MATTRA,DD)
      DIMENSION MAT1(4,5),D(5,5,4),MATTRA(5,4),DD(4,4,4)
      DIMENSION DIN(4,5,4)
      DOUBLE PRECISION SOM
      DOUBLE PRECISION DIN
      DOUBLE PRECISION MAT1,D,MATTRA,DD
      COMMON LEC,IMP,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9
      DO 1 K=1,4
      DO 10 I=1,4
      DO 10 J=1,5
      SOM=0.0
      DO 20 L=1,5
      SOM=SOM+MAT1(I,L)*D(L,J,K)
   20 CONTINUE
      DIN(I,J,K)=SOM
   10 CONTINUE
      DO 30 I=1,4
      DO 30 J=1,4
      SOM=0.0
      DO 40 L=1,5
      SOM=SOM+DIN(I,L,K)*MATTRA(L,J)
   40 CONTINUE
      DD(I,J,K)=SOM
   30 CONTINUE
    1 CONTINUE
      RETURN
      END
```

PROGRAMME

FONCTION D'ONDE

```
// EXEC FORTRAN
      DEFINE FILE 9(150,70,U,N6)
      DEFINE FILE 10(100,70,U,N7)
      DEFINE FILE 12(150,70,U,N9)
      COMMON ABO(32),AB(32),AB1(32),AB2(32),VPV(32)
      COMMON Z(32,32),W(32),Q(32),RNS(32,32),H(32,32)
      DIMENSION MULT(32,32),SPIN(32),ENERG(32)
      DIMENSION G(32,32)
      DIMENSION G1(32,32)
      DIMENSION S(32,32),VS(32),VPS(32,32),C(32,32),VC(32),VPC(32,32)
      DIMENSION IND(32),NTYP(32),MTAB(32)
      DOUBLE PRECISION VPS,VS,VPC,VC
      DOUBLE PRECISION VPV,AB1,AB2,S,Z,W,Q,RNS,H,C,G,ABO,AB,MULT,RES
      DOUBLE PRECISION SPIN,GB,ENERG,TKO,SO,ALFAO,A,TN,SA,ST,TM,TU,TUTU
      DOUBLE PRECISION TO,TOTO,ALF,BET,GAM,ENORM,TOTAL
      DOUBLE PRECISION TD1,TD2,TD3,TD,DTM,DST,DALF,DBET,DGAM,DENOR
      DOUBLE PRECISION TD4,DTOTAL,DRAP
      DOUBLE PRECISION SOM1
      DOUBLE PRECISION SOM25
      DOUBLE PRECISION G1
      EQUIVALENCE (VPS(1,1),G(1,1))
      EQUIVALENCE(VPS(1,1),C(1,1))
      EQUIVALENCE(VPC(1,1),MULT(1,1))
    1 FORMAT (I4)
    2 FORMAT(4F20.14)
    3 FORMAT(3F20.14)
    4 FORMAT(16I5)
    5 FORMAT(3HOM=,I4)
    6 FORMAT(1HO,20F5.1)
    7 FORMAT (1H ,I4)
   50 FORMAT('OMATRICE S SYMETRIQUE')
   51 FORMAT('OMATRICE Z SYMETRIQUE')
   52 FORMAT(10HOVECTEUR W)
   53 FORMAT('OVECTEUR CHARGE INITIAL')
   54 FORMAT(1H ,4I10)
   88 FORMAT(1HO,2F20.8)
  343 FORMAT('OVALEUR PROPRE DE C ET VECTEUR PROPRE ASSOCIE     ENERGIE
     1EN E.V. ET VECTEUR ASSOCIE AUX O.M.')
  393 FORMAT(29HO NOMBRE D ITERATIONS DEPASSE)
  395 FORMAT (13HO FIN CALCULS)
  444 FORMAT(F13.8)
 2003 FORMAT('ODENSITE DE SPIN   VECTEUR CHARGE')
 2005 FORMAT(1HO,'NOMBRE D ITERATIONS')
 2006 FORMAT(16HONOMBRE D ATOMES)
 2007 FORMAT(16HOTYPE DES ATOMES)
 2008 FORMAT(8F10.4)
```

```
 2009 FORMAT(28HOCOEFFICIENT DE MODIFICATION)
 2011 FORMAT(14HON    NSO  NDO)
 2022 FORMAT('OMATRICE H SYMETRIQUE')
 2223 FORMAT(' SOMME DES CHARGES AVANT NORMALISATION',F10.6)
 2225 FORMAT(' SOMME DES CHARGES APRES NORMALITION',F10.6)
 2226 FORMAT(' SOMME DES ELEMENTS DU VECTEUR DENSITE DE SPIN',F10.6)
 2227 FORMAT(' SOMME DES ELEMENT DU VECTEUR SPIN AVANT NORM',F10.6)
 3502 FORMAT(5X,F4.1)
 3503 FORMAT(F20.14)
 3505 FORMAT(1H0,10F8.4)
 3506 FORMAT(4F20.14)
 5001 FORMAT(31HONOMBRE D ITERATIONS EFFECTUEES)
 5002 FORMAT(I3)
 7777 FORMAT(1H1)
 8801 FORMAT(10X,F10.5,40X,F15.8)
 8802 FORMAT(15X,F15.8,45X,F15.8)
 8804 FORMAT(1H0)
 8805 FORMAT(17HOITERATION NUMERO,I2)
 8806 FORMAT('OUNE DES VALEURS PROPRES DE S EST NEGATIVE IL Y A UNE
     1ERREUR')
 8809 FORMAT('OKKK=',I4)
 8810 FORMAT('OVECTEUR CHARGE')
 8811 FORMAT(1H0,F20.8)
 8812 FORMAT('OLA SOMME DES ENERGIES VAUT',F15.8)
      LEC=1
      IPERF=2
      IMP=3
C    KKK=1 SI ON VEUT UTILISER L ITERATION DE SACKS
C    KKK=0 SI ON NE VEUT PAS UTILISER L ITERATION DE SACKS
      IB9=9
      IB10=10
      IB12=12
      NDIM=32
      N9=1
      READ(LEC,1)KKK
      WRITE(IMP,8809)KKK
      READ(IB12'N9)M
      WRITE(3,5)M
      N6=3*NDIM+1
      DO 1111 I=1,NDIM
 1111 READ(IB9'N6)(S(I,J),J=1,NDIM)
      DO 522 I=1,M
  522 READ(IB12'N9)(Z(I,J),J=1,I)
      DO 8  I=1,M
      DO 8  J=1,I
```

```
      Z(J,I)=Z(I,J)
    8 CONTINUE
      WRITE(IMP,50)
      DO 3100 I=1,M
 3100 WRITE(IMP,6)(S(I,J),J=1,I)
      WRITE(IMP,51)
      DO 3101 I=1,M
 3101 WRITE(IMP,6)(Z(I,J),J=1,I)
      READ(IB12'N9)(W(I),I=1,M)
      WRITE(IMP,52)
      DO 3102 I=1,M
 3102 WRITE(IMP,6)W(I)
      READ(LEC,3)TKO,SO,ALFAO
      WRITE(IMP,3)TKO,SO,ALFAO
      READ(LEC,1)NIT
      WRITE(IMP,2005)
      WRITE(IMP,7)NIT
C
C NIT  NOMBRE D ITERATION
C
C DEFINITION DE LA STRUCTURE ATOMIQUE DE LA MOLECULE
C    NA = NOMBRE D ATOME
C       NTYP DEFINIT LE TYPE D ATOME
C LECTURE DES RAYONS R JOIGNANT LES ATOMES
C  LE TABLEAU R MESURE LES DISTANCES INTER ATOMIQUES EN ANGSTROMS
      READ(LEC,1)NA
      WRITE(IMP,2006)
      WRITE(IMP,7)NA
      READ(LEC,4)(NTYP(I),I=1,NA)
      WRITE(IMP,2007)
      WRITE(IMP,54)(NTYP(I),I=1,NA)
      NANA=NA-1
C
      A=14.400
C
C  A CONSTANTE INTERVENANT DANS LE CALCUL DE H(I,I)
C     A=14.400 EN E.V.
C
C  LECTURE DES CHARGES EN 8F10.4
      WRITE(IMP,53)
      READ(LEC,2008)(Q(I),I=1,M)
      DO 3104 I=1,M
      WRITE(IMP,6)Q(I)
 3104 CONTINUE
C   LECTURE DU NOMBRE D ELECTRONS  N  POUR ITERER
```

```
C      NSO  = NOMBRE D ORBITALES SIMPLEMENT OCCUPEES
C     NDO  = NOMBRE D ORBITALES DOUBLEMENT OCCUPEES
       READ(LEC,4)N,NSO,NDO
       WRITE(IMP,2011)
       WRITE(IMP,4)N,NSO,NDO
C FIN LECTURE DONNEE
C
C   CETTE SEQUENCE EST A MODIFIER
C
       TN=N
       IF(MOD(N,2)) 366,322,366
  322  N=N/2
       NSO=0
       GO TO 390
  366  N=(N-1)/2
       NSO=1
  390  SA=S0*ALFA0
       ST=(TK0-1.)/S0
       TM=TK0*0.50
C FIN ECRITURES DONNEES
C
C NCOMP COMPTE TOUTES LES 3 ITERATIONS AFIN DE FAIRE SACKS
C NITA COMPTE LES ITERATIONS  QUAND NITA VAUT NIT   SORTIE
C
       NCOMP=0
       NITA=0
C
C MTAB  TABLEAU INDIQUANT POUR CHAQUE ATOME LE NOMBRE D ORBITALE LUI
CORRESPONDANT
C
       DO 103  I=1,NA
       IF(NTYP(I)-4) 102,101,102
  101  MTAB(I)=1
       GO TO 103
  102  MTAB(I)=4
  103  CONTINUE
       DO 9  I=1,M
       DO 9  J=1,M
    9  VPS(I,J)=S(I,J)
       CALL UDEJAC(VPS,VS,M)
C LES VECTEURS PROPERS SONT LES LIGNES DE VPS
       DO 11  I=1,M
       IF(VS(I))10,11,11
   10  WRITE(IMP,8806)
       STOP
```

```
   11 CONTINUE
      DO 13  J=1,M
      AB1(J)=0.
      AB2(J)=0.
      DO 13  I=1,J
      RNS(I,J)=0.
      SOM1=0.0
      DO 12 K=1,M
      TD1=VPS(K,I)
      TD2=VPS(K,J)
      TD4=VS(K)
      TD3=DSQRT(TD4)
   12 SOM1=SOM1+TD1*TD2/TD3
   13 RNS(I,J)=SOM1
C A CE STADE VS ET VPS NE SERVENT PLUS A RIEN
C MATRICE H
C LES CALCULS SUIVANTS NOUS FONT PASSER D UN VECTEUR CHARGE Q
C            A UN VECTEUR CHARGE AB
  391 NITA=NITA+1
      SOM25=0.0
      WRITE(IMP,7777)
      WRITE(IMP,8805)NITA
      DO 15 I=1,M
      SOM1=W(I)
      TD=0.0
      DO 150 J=1,M
      TD1=Z(I,J)
      TD2=Q(J)
  150 TD=TD+TD1*TD2
   15 H(I,I)=SOM1+TD
C MODIF DE BERTHIER DES TERMES DIAGONAUX
      DO 16 I=1,M
      TD1=ALFA0
      TD2=SA
      TD3=H(I,I)
      TD=(TD3-TD1)/TD2
   16 H(I,I)=TD
      DO 17  J=2,M
      JJ=J-1
      DO 17 I=1,JJ
 1055 TD1=H(I,I)
      TD2=H(J,J)
      DTM=TM
      DST=ST
      TD3=S(I,J)
```

```
      H(I,J)=((TD1+TD2)*DTM+DST)*TD3
   17 CONTINUE
      DO 18  I=1,M
      DO 18  J=1,I
      RNS(I,J)=RNS(J,I)
   18 H(I,J)=H(J,I)
      WRITE(3,2022)
      DO 3109 I=1,M
 3109 WRITE(3,6)(H(I,J),J=1,I)
      DO 20  I=1,M
      DO 20  J=1,M
      VPC(I,J)=0.
      SOM1=0.0
      DO 19 K=1,M
      TD2=H(I,K)
      TD3=RNS(K,J)
   19 SOM1=SOM1+TD2*TD3
   20 VPC(I,J)=SOM1
      DO 22  J=1,M
      DO 22 I=1,J
      C(I,J)=0.
      SOM1=0.0
      DO 21 K=1,M
      TD2=RNS(I,K)
      TD3=VPC(K,J)
   21 SOM1=SOM1+TD2*TD3
   22 C(I,J)=SOM1
      DO 23  I=1,M
      DO 23  J=1,I
      VPC(J,I)=C(J,I)
   23 VPC(I,J)=VPC(J,I)
      CALL UDEJAC(VPC,VC,M)
C CLASSEMENT DES VALEURS PROPRES DEC PAR ORDRE CROISSANT
      IPLACE=1
      IND(IPLACE)=1
   24 IPLACE=IPLACE+1
      IND(IPLACE)=IPLACE
      I=IPLACE
   25 I1=IND(I)
      I2=IND(I-1)
      IF(VC(I1)-VC(I2)) 26,28,28
   26 K=IND(I)
      IND(I)=IND(I-1)
      IND(I-1)=K
      IF(I-2) 27,28,27
```

```
   27 I=I-1
      GO TO 25
   28 IF(IPLACE-M) 24,29,24
C CLASSEMENT CORRESPONDANT DES VECTEURS PROPRES  CREATION DE LA MATRICE ORDO
C VPC LES VECTEURS PROPRES SONT LES LIGNES
   29 DO 33 I=1,M
      DO 33 J=1,M
   33 G(I,J)=VPC(J,I)
      DO 34 IPLACE=1,M
      J=IND(IPLACE)
       II=M+1-IPLACE
      VPV(II)=VC(J)
      TD1=ALFAO
      TD2=SA
      TD3=VC(J)
      TD=TD1+TD2*TD3
      ENERG(II)=TD
      DO 34 I=1,M
   34 VPC(I,IPLACE)=G(I,J)
      DO 334 I=1,M
      DO 334 J=1,M
      G(I,J)=0.
      SOM1=0.0
      DO 3340  K=1,M
      TD2=RNS(I,K)
      TD3=VPC(K,J)
 3340 SOM1=SOM1+TD2*TD3
  334 G(I,J)=SOM1
 8011 CONTINUE
      WRITE(3,343)
      DO 8800 I=1,M
      II=M+1-I
      SOM25=SOM25+ENERG(I)*VS(II)
      DO 8803 J=1,M
      G1(J,I)=G(J,II)
 8803 CONTINUE
 8800 CONTINUE
 8010 CONTINUE
      DO 333 I=1,M
  333 VS(I)=0.
      IF (NSO-1) 344,389,344
  389 L=M-N
      VS(L)=1.
  344 L=M-N+1
      DO 355 I=L,M
```

```
355 VS(I)=2.
    DO 5551 I=1,M
    L=M-N
    DO 5551 J=1,M
   CALCUL  DE  V*VBARRE*S
    TD1=G(I,L)
    TD2=G(J,L)
    TD=TD1*TD2
5551 MULT(I,J)=TD
    DO 5553 I=1,M
    DO 5553 J=1,M
    SOM1=0.0
    DO 5554 K=1,M
    TD2=MULT(I,K)
    TD3=S(K,J)
5554 SOM1=SOM1+TD2*TD3
    IF(I-J)5553,5552,5553
5552 SPIN(I)=SOM1
5553 CONTINUE
  CALCUL DE  GBARRE*S
    DO 6662 I=1,M
    DO 6662 J=1,M
    MULT(I,J)=0.
    SOM1=0.0
    DO 6652 K=1,M
    TD2=G(K,I)
    TD3=S(K,J)
6652 SOM1=SOM1+TD2*TD3
6662 MULT(I,J)=SOM1
   CALCUL DE N*G
    DO 6663 J=1,M
    DO 6663 I=1,M
    TD1=VS(J)
    TD2=G(I,J)
    TD=TD1*TD2
6663 G(I,J)=TD
   CALCUL DE N*G*GBARRE*S
    DO 6664 I=1,M
    DO 6664 J=1,M
    SOM1=0.0
    DO 6665 K=1,M
    TD2=G(I,K)
    TD3=MULT(K,J)
6665 SOM1=SOM1+TD2*TD3
    IF(I-J)6664,6666,6664
```

```
6666 AB(I)=SOM1
6664 CONTINUE
     TU=0.
     TUTU=0.
     TO=0.
     TOTO=0.
     DO 2222 I=1,M
     TU=TU+SPIN(I)
2222 TO=TO+AB(I)
     IF(NSO)4000,4001,4000
4000 WRITE(IMP,2227)TU
4001 WRITE(IMP,2223)TO
     DO 2224 I=1,M
     TD1=TU
     TD2=TN
     TD3=TO
     TD4=SPIN(I)
     TD=TD4/TD1
     SPIN(I)=TD
     TD4=AB(I)
     TD=(TD4/TD3)*TD2
     AB(I)=TD
     TUTU=TUTU+SPIN(I)
2224 TOTO=TOTO+AB(I)
     IF(NSO)4002,4003,4002
4002 WRITE(IMP,2003)
     DO 3111 I=1,M
3111 WRITE(IMP,88)SPIN(I),AB(I)
     WRITE(IMP,2226)TUTU
     WRITE(IMP,2225)TOTO
     GO TO 4004
4003 WRITE(IMP,8810)
     DO 4005 I=1,M
4005 WRITE(IMP,8811)AB(I)
     WRITE(IMP,2225)TOTO
4004 LAS=0
     DO 380 J=1,M
     IF(ABS(AB1(J)-AB(J))-0.001)379,379,380
 379 LAS=LAS+1
 380 CONTINUE
     IF ( LAS-M) 381,394,394
 381 DO 378 J=1,M
     AB1(J)=AB(J)
 378 AB(J)=(AB(J)+Q(J))/2.
     IF(NITA-NIT)382,382,392
```

```
  382 NCOMP=NCOMP+KKK
      DO 384 J=1,M
      ABO(J)=AB2(J)
      AB2(J)=Q(J)
  384 Q(J)=AB(J)
      SOM25=SOM25/2.0
      WRITE(IMP,8812)SOM25
      IF(NCOMP-2)391,385,385
C ITERATION DE SACKS
  385 NCOMP=0
       ALF=0.
       BET=0.
       GAM=0.
      DO 386 J=1,M
      TD1=AB(J)
      TD2=AB2(J)
      TD3=ABO(J)
      DALF=ALF
      DBET=BET
      DGAM=GAM
      ALF=DALF+(TD1-TD2)**2
      BET=DBET+(TD1-TD2)*(TD2-TD3)
  386 GAM=DGAM+(TD2-TD3)**2
      ENORM = 1./ ( ALF-2.*BET+GAM)
      TOTAL=0.
      DO 387 J=1,M
      DENOR=ENORM
      DALF=ALF
      DBET=BET
      DGAM=GAM
      TD1=AB(J)
      TD2=AB2(J)
      TD3=ABO(J)
      Q(J)=DENOR*(DALF*TD3-2.*DBET*TD2+DGAM*TD1)
  387 CONTINUE
      GO TO 391
C   SORTIE    DIFFERENTES
C   SOIT LE NOMBRE D ITERATIONS EST DEPASSE
C              SOIT ON A OBTENU UNE BONNE VALEUR
  392 WRITE(IMP,393)
      GO TO 396
  394 WRITE(IMP,5001)
      WRITE(IMP,5002)NITA
      WRITE(IMP,395)
  396 CONTINUE
```

```
      N9=1
      WRITE(IB12'N9)(ENERG(I),I=1,M)
      N7=1
      WRITE(IMP,343)
      DO 3500 I=1,M
      WRITE(IMP,8811)ENERG(I)
      WRITE(IB10'N7)(G1(J,I),J=1,M)
      WRITE(IMP,8811)(G1(J,I),J=1,M)
3500  CONTINUE
```

```
// EXEC FORTRAN
      SUBROUTINE UDEJAC(VECP,VALP,N)
      DIMENSION VECP(32,32),VALP(32),A(528),R(1024)
      DOUBLE PRECISION A,R
      DOUBLE PRECISION VECP,VALP
      K=1
      DO 1 I=1,N
      DO 1 J=1,I
      A(K)=VECP(I,J)
    1 K=K+1
      CALL DEJAC(A,R,N,0)
      DO 2 I=1,N
      ID=I*(I-1)/2
      IL=ID+I
      VALP(I)=A(IL)
      DO 2 J=1,N
      JD=(I-1)*N
      JL=JD+J
    2 VECP(I,J)=R(JL)
      RETURN
      END
```

PROGRAMME

LOCALISATION DES ORBITALES MOLECULAIRES

```
// EXEC FORTRAN
      DEFINE FILE 5(100,70,U,N2)
      DEFINE FILE 9(150,70,U,N6)
      DIMENSION C(32,32),T(32,32),IP(32),VALP(32)
      DIMENSION A(528),RR(1024)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C,T
      DOUBLE PRECISION X1,C,T
      DOUBLE PRECISION A,RR
      DOUBLE PRECISION VALP
 1000 FORMAT(1H0,8F10.6)
 9996 FORMAT('0MATRICE C  DE DEJAC')
 9997 FORMAT(1H0,8F10.5)
 9998 FORMAT('0MATRICE T  DE DEJAC')
 9999 FORMAT('0DIAGO EFFECTUE')
      IB3=3
      IB5=5
      IB9=9
      NA=10
      NDIM=32
      N=NA
      N6=3*NDIM+1
      DO 10 I=1,NDIM
   10 READ(IB9'N6)(C(I,J),J=1,NDIM)
      DO 1001 I=1,NA
 1001 WRITE(IB3,1000)(C(I,J),J=1,I)
      K=1
      DO 1 I=1,N
      DO 1 J=1,I
      A(K)=C(I,J)
    1 K=K+1
      CALL DEJAC(A,RR,N,0)
      DO 2 I=1,N
      ID=I*(I-1)/2
      IL=ID+I
      VALP(I)=A(IL)
      DO 2 J=1,N
      JD=(I-1)*N
      JL=JD+J
    2 T(J,I)=RR(JL)
      DO 3 I=1,N
      DO 3 J=1,N
```

```
      C(I,J)=0.0
      IF(I-J)3,4,3
    4 C(I,J)=VALP(I)
    3 CONTINUE
      N2=1
      WRITE(IB5'N2)(C(IC,IC),IC=1,NA)
      DO 300 IT=1,NA
      WRITE(IB5'N2)(T(IT,JT),JT=1,NA)
  300 CONTINUE
      WRITE(IB3,9999)
      END
```

```
 EXEC FORTRAN
DEJAC   VERSION 01 FORTRAN 30/06/67 MATH E002
    SUBROUTINE DEJAC(A,R,N,MV)
       SUBROUTINE EIGEN                                                 EIGE
C                                                                       EIGE
C      PURPOSE                                                          EIGE
C         COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC      EIGE
C         MATRIX                                                        EIGE
C                                                                       EIGE
C      USAGE                                                            EIGE
C         CALL EIGEN(A,R,N,MV)                                          EIGE
C                                                                       EIGE
C      DESCRIPTION OF PARAMETERS                                        EIGE
C         A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.    EIGE
C             RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF        EIGE
C             MATRIX A IN DESCENDING ORDER.                             EIGE
C         R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,       EIGE
C             IN SAME SEQUENCE AS EIGENVALUES)                          EIGE
C         N - ORDER OF MATRICES A AND R                                 EIGE
C         MV- INPUT CODE                                                EIGE
C                   0   COMPUTE EIGENVALUES AND EIGENVECTORS            EIGE
C                   1   COMPUTE EIGENVALUES ONLY (R NEED NOT BE         EIGE
C                       DIMENSIONED BUT MUST STILL APPEAR IN CALLING    EIGE
C                       SEQUENCE)                                       EIGE
C                                                                       EIGE
C      REMARKS                                                          EIGE
C         ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)     EIGE
C         MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R           EIGE
C                                                                       EIGE
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    EIGE
C         NONE                                                          EIGE
C                                                                       EIGE
C      METHOD                                                           EIGE
C         DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED       EIGE
C         BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICALEIGE
C         METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND      EIGE
C         H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7     EIGE
C                                                                       EIGE
C   ....................................................................EIGE
C                                                                       EIGE
      DIMENSION A(1),R(1)
      DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,Z,V,SINX,SINX2,COSX,
     *                 COSX2,SINCS,YZ,YU,ZU,UW
C                                                                       EIGE
C      ....................................................................EIGE
C                                                                       EIGE
```

```
      IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  EIGE
      C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION      EIGE
      STATEMENT WHICH FOLLOWS.                                       EIGE
                                                                     EIGE
     DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,       EIGE
     1                 COSX2,SINCS                                   EIGE
                                                                     EIGE
      THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS    EIGE
      APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS      EIGE
      ROUTINE.                                                       EIGE
                                                                     EIGE
      THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO      EIGE
      CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SQRT IN STATEMENTSEIGE
      40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT.  ABS IN STATEMENT EIGE
      62 MUST BE CHANGED TO DABS.                                    EIGE
                                                                     EIGE
      ..............................................................EIGE
                                                                     EIGE
      GENERATE IDENTITY MATRIX                                       EIGE
                                                                     EIGE
      IF(MV-1) 10,25,10                                              EIGE
   10 IQ=-N                                                          EIGE
      DO 20 J=1,N                                                    EIGE
      IQ=IQ+N                                                        EIGE
      DO 20 I=1,N                                                    EIGE
      IJ=IQ+I                                                        EIGE
      R(IJ)=0.0                                                      EIGE
      IF(I-J) 20,15,20                                               EIGE
   15 R(IJ)=1.0                                                      EIGE
   20 CONTINUE                                                       EIGE

            NORMALISATION DU PLUS GRAND TERME EXTRADIAGONAL

   25 Z=DABS(A(1))
      MVK=0
      DO 200 I=1,N
      K=(I*I+I)/2
      V=DABS(A(K))
      IF (Z-V) 1200,1200,200
 1200 Z=V
  200 CONTINUE
      ZU=DABS(A(2))
      DO 220 I=1,N
      DO 220 J=I,N
      IF(I-J) 215,220,215
```

```
  215 K=I+(J*J-J)/2
      V=DABS(A(K))
      IF (ZU-V) 1220,1220,220
 1220 ZU=V
  220 CONTINUE
      IF (ZU-0.D0) 1165,165,1165
 1165 Y=1.D30
      NN=(N*(N-1))/2
      NNL=NN+N
      YZ=DFLOAT(NN)
      YU=1.D37/DSQRT(YZ)
      UW=1.D-37/DSQRT(YZ)
      IF(Z-1.D-44)133,134,134
  133 IF(ZU-UW)141,142,142
  142 Z=ZU/YU
      GO TO 400
  141 MVK=MVK+1
      DO 147 K=1,NNL
  147 A(K)=A(K)*1.D50
      Z=Z*1.D50
       ZU=ZU*1.D50
      GO TO 137
  134 IF(ZU-UW)401,137,137
  401 Z=Z/Y
      GO TO 400
  137 Z=Z/Y
      ZU=ZU/YU
      IF(Z-ZU)132,400,400
  132 Z=ZU
  400 MVK=MVK+1
      DO 205 K=1,NNL
      A(K)=A(K)/Z
  205 CONTINUE
C                                                                       EIG
C         COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)            EIG
C                                                                       EIG
      ANORM=0.D0
      DO 35 I=1,N                                                       EIG
      DO 35 J=I,N                                                       EIG
      IF(I-J) 30,35,30                                                  EIG
   30 IA=I+(J*J-J)/2                                                    EIG
      ANORM=ANORM+A(IA)*A(IA)                                           EIG
   35 CONTINUE                                                          EIG
      IF(ANORM) 165,165,40                                              EIG
   40 ANORM=1.414*SQRT(ANORM)
```

```
      ANRMX=ANORM*1.0D-15/DFLOAT(N)
C                                                                       EIG
C        INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR               EIG
C                                                                       EIG
      IND=0                                                             EIG
      THR=ANORM                                                         EIG
   45 THR=THR/DFLOAT(N)
   50 LQ=0
      L=1
   55 M=L+1                                                             EIG
C                                                                       EIG
C        COMPUTE SIN AND COS                                            EIG
C                                                                       EIG
      LQ=LQ+L-1
   60 MQ=(M*M-M)/2                                                      EIG
      LM=L+MQ                                                           EIG
   62 IF(DABS(A(LM))-THR) 130,65,65
   65 IND=1                                                             EIG
      LL=L+LQ                                                           EIG
      MM=M+MQ                                                           EIG
      X=0.5*(A(LL)-A(MM))                                               EIG
   68 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75                                                    EIG
   70 Y=-Y                                                              EIG
   75 SINX=Y/DSQRT(2.0D0*(1.0D0+(DSQRT(1.0D0-Y*Y))))
      SINX2=SINX*SINX
   78 COSX=DSQRT(1.0D0-SINX2)
      COSX2=COSX*COSX
      SINCS=SINX*COSX
C        ROTATE L AND M COLUMNS                                         EIG
C                                                                       EIG
      ILQ=N*(L-1)                                                       EIG
      IMQ=N*(M-1)                                                       EIG
      DO 125 I=1,N                                                      EIG
      IQ=(I*I-I)/2                                                      EIG
      IF(I-L) 80,115,80                                                 EIG
   80 IF(I-M) 85,115,90                                                 EIG
   85 IM=I+MQ                                                           EIG
      GO TO 95                                                          EIG
   90 IM=M+IQ                                                           EIG
   95 IF(I-L) 100,105,105                                               EIG
  100 IL=I+LQ                                                           EIG
      GO TO 110                                                         EIG
  105 IL=L+IQ                                                           EIG
  110 X=A(IL)*COSX-A(IM)*SINX                                           EIG
```

```
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
  115 IF(MV-1) 120,125,120
  120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
  125 CONTINUE
      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C
C        TESTS FOR COMPLETION
C
C        TEST FOR M = LAST COLUMN
C
  130 IF(M-N) 135,140,135
  135 M=M+1
      GO TO 60
C
C        TEST FOR L = SECOND FROM LAST COLUMN
C
  140 IF(L-(N-1)) 145,150,145
  145 L=L+1
      GO TO 55
  150 IF(IND-1) 160,155,160
  155 IND=0
      GO TO 50
C
C        COMPARE THRESHOLD WITH FINAL NORM
C
  160 IF(THR-ANRMX) 165,165,45
C
C        SORT EIGENVALUES AND EIGENVECTORS
C
  165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=(I*I+I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
```

```
      JQ=JQ+N
      MM=(J*J+J)/2
      IF(A(LL)-A(MM)) 170,185,185
  170 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
      IF(MV-1) 175,185,175
  175 DO 180 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
      IF (MVK) 1450,450,1450
 1450 IF (MVK-1) 1460,460,1460
 1460 DO 360 K=1,NNL
  360 A(K)=A(K)*1.D-50
  460 DO 350 K=1,NNL
      DO 350 K=1,NNL
  350 A(K)=A(K)*Z
  450 RETURN
      END
// EXEC LNKEDT
// ASSGN SYS001,X'192'
// ASSGN SYS002,X'192'
// ASSGN SYS003,X'191'
// ASSGN SYS004,X'192'
// ASSGN SYS005,X'192'
// ASSGN SYS006,X'192'
// DLBL IJSYS03,'COP1 DOUADY RECOUVREMENT',69/365
// EXTENT ,9,,,1960,20
// DLBL IJSYS04,,1
// EXTENT ,6,,,1100,200
// DLBL IJSYS05,,1
// EXTENT ,6,,1300,200
// DLBL IJSYS06,,1
// EXTENT ,6,,1500,200
// EXEC
// OPTION LINK
```

```
// EXEC FORTRAN
      DEFINE FILE 5(100,70,U,N2)
      DEFINE FILE 9(150,70,U,N6)
      DEFINE FILE 10(100,70,U,N7)
      DEFINE FILE 12(150,70,U,N9)
      INTEGER SAVE
      DIMENSION S(32,32),T(32,32),SPLUS(32,32),EPS(32),IP(32)
      DIMENSION SMIN(32,32),C(32,32),X(32,32)
      DIMENSION EPS1(32)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      COMMON T
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION T
      DOUBLE PRECISION SMIN,S
      DOUBLE PRECISION AUX1,AUX2,AUX3,SPLUS,X,EPS
      DOUBLE PRECISION EPS1
      EQUIVALENCE(SMIN(1,1),S(1,1)),(S(1,1),C(1,1))
      EQUIVALENCE(SPLUS(1,1),X(1,1))
 1000 FORMAT('0K=',I2)
 1002 FORMAT(1H0,8F10.6)
 1003 FORMAT(16I5)
 9999 FORMAT('0LOWD EFFECTUE')
      IB1=1
      IB3=3
      IB5=5
      IB9=9
      IB10=10
      IB12=12
      NDIM=32
      READ(IB1,1003)NA
      NT=NA
      READ(IB1,1003)(IP(I),I=1,NT)
      N2=1
      READ(IB5'N2)(C(IC,IC),IC=1,NA)
      DO 601 IT=1,NA
      READ(IB5'N2)(T(IT,JT),JT=1,NA)
  601 CONTINUE
      DO 5 I=1,NA
    5 EPS(I)=DSQRT(C(I,I))
      DO 15 I=1,NA
      DO 14 J=1,I
```

```
      AUX1=0.0
      AUX2=0.0
      DO 10 K=1,NA
      AUX3=T(I,K)*T(J,K)
      AUX1=AUX1+AUX3*EPS(K)
   10 AUX2=AUX2+AUX3/EPS(K)
       SMIN(I,J)=AUX2
      SMIN(J,I)=AUX2
      SPLUS(I,J)=AUX1
      SPLUS(J,I)=AUX1
   14 CONTINUE
   15 CONTINUE
      N7=NDIM+1
      DO 200 IS=1,NA
  200 WRITE(IB10'N7)(SMIN(IS,JS),JS=1,NA)
      N7=1
      DO 300 I=1,NT
      READ(IB10'N7)(EPS(J),J=1,NA)
      KNS=IP(I)
      DO 3050 J=1,NA
      C(KNS,J)=EPS(J)
 3050 CONTINUE
  300 CONTINUE
      N9=1
      READ(IB12'N9)(EPS(J),J=1,NT)
      DO 303 I=1,NT
      KNS=IP(I)
      EPS1(KNS)=EPS(I)
  303 CONTINUE
      N9=1
      WRITE(IB12'N9)(EPS1(I),I=1,NT)
      DO 25 I=1,NA
      DO 24 J=1,NA
      AUX1=0.0
      DO 20 K=1,NA
   20 AUX1=AUX1+C(I,K)*SPLUS(K,J)
      T(I,J)=AUX1
   24 CONTINUE
   25 CONTINUE
      N7=1
      N9=2
      DO 301 IT=1,NA
      WRITE(IB10'N7)(T(IT,JT),JT=1,NA)
  301  WRITE(IB12'N9)(T(IT,JT),JT=1,NA)
      N7=NDIM+1
```

```
      DO 302 IS=1,NA
  302 READ(IB10'N7)(SMIN(IS,JS),JS=1,NA)
      N6=1
      DO 50 N=1,3
      SAVE=N6
      DO 500 IC=1,NDIM
  500 READ(IB9'N6)(X(IC,JC),JC=1,NDIM)
      WRITE(IB3,1000)N
      DO 1001 IC=1,NA
 1001 WRITE(IB3,1002)(X(IC,JC),JC=1,IC)
      DO 40 I=1,NA
      DO 39 J=1,I
      AUX1=0.0
      DO 35 K=1,NA
      AUX2=SMIN(I,K)
      AUX3=0.0
      DO 30 L=1,NA
   30 AUX3=AUX3+SMIN(J,L)*X(K,L)
   35 AUX1=AUX1+AUX2*AUX3
      T(I,J)=AUX1
      T(J,I)=AUX1
   39 CONTINUE
   40 CONTINUE
      N6=SAVE
      DO 400 IC=1,NDIM
  400 WRITE(IB9'N6)(T(IC,JC),JC=1,NDIM)
   50 CONTINUE
      WRITE(IB3,9999)
      END
// EXEC LNKEDT
// ASSGN SYS002,X'192'
// ASSGN SYS006,X'192'
// ASSGN SYS007,X'192'
// ASSGN SYS009,X'192'
// DLBL IJSYS02,'O NE PAS PROTEGER 2',68/001
// EXTENT SYS002,6,,,1110,10
// DLBL IJSYS06,'Y NE PAS PROTEGER 6',68/001
// EXTENT SYS006,6,,,1620,20
// DLBL IJSYS07,'J NE PAS PROTEGER 7',68/001
// EXTENT SYS007,6,,,1640,10
// DLBL IJSYS09,'R NE PAS PROTEGER 9',68/001
// EXTENT SYS009,6,,,1660,20
// EXEC
  10
    1    2    3    4    5    6    7    8    9   10   11   12
```

```
/&
// JOB DOUADY   0125582100101072,COP1   INTEGRALES DIP. 4PAR 4
// OPTION LINK
    PHASE ZUTZUT,ROOT
```

```
// EXEC FORTRAN
      DEFINE FILE 5(100,70,U,N2)
      DEFINE FILE 6(2100,70,U,N3)
      DEFINE FILE 7(2100,70,U,N4)
      DEFINE FILE 8(1100,70,U,N5)
      DEFINE FILE 9(150,70,U,N6)
      DEFINE FILE 10(100,70,U,N7)
      DEFINE FILE 11(100,70,U,N8)
      DEFINE FILE 12(150,70,U,N9)
      DEFINE FILE 13(100,70,U,N1)
      INTEGER CALC,CAMX
      DIMENSION C(32,32),T(32,32),IP(32)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      COMMON T
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION T
 1000 FORMAT('OROSS =',E13.6)
 1002 FORMAT(1H1)
 1010 FORMAT('ONEWT = ',E13.6)
      IB1=1
      IB2=2
      IB3=3
      IB4=13
      IB5=5
      IB6=6
      IB7=7
      IB8=8
      IB9=9
      IB10=10
      IB11=11
      IB12=12
      IB13=13
      IB14=14
      NDIM=32
      DO 60 ITERA=1,2
      CALL OVELAY(1)
      CALL LECTUR
      N7=1
      DO 600 IC=1,NA
  600 READ(IB10'N7)(C(IC,JC),JC=1,NA)
      N5=1
```

```
      READ(IB8'N5)CAMX
      CALL OVELAY(2)
      DO 50 CALC=1,CAMX
      READ(IB8'N5)IMIN,IMAX
    5 IF(IP(IMIN))50,15,10
   10 CONTINUE
C     CALL ROSS1
      GO TO 20
   15 CALL ROSSI
   20 WRITE(IB3,1000)X1
      IF(X1-0.03)25,5,5
   25 IF(IP(IMIN))50,35,30
   30 CONTINUE
C     CALL NEWT1
      GO TO 40
   35 CALL NEWTO
   40 WRITE(IB3,1010)X1
      IF(X1-1.D-5)50,50,45
   45 CALL ORTHO
      GO TO 25
   50 CONTINUE
      CALL OVELAY(3)
      CALL RESUL
      WRITE(IB3,1002)
   60 CONTINUE
      END
   PHASE DOUADY1,*
```

```
// EXEC FORTRAN
      SUBROUTINE LECTUR
      INTEGER CAMX
      DIMENSION IP(32),E(32),C(32,32),NS(32),EE(32)
      DIMENSION JMIN(4),JMAX(4)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      DOUBLE PRECISION CC,EE
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION E
    5 FORMAT(3I3)
    7 FORMAT(8F10.6)
   10 FORMAT(8F10.0)
   15 FORMAT(1H0/1H0,'NOMBRE O.A.',I3 /1H0/'NOMBRE O.M. OCC.',I3/1H0/'
     1NOMBRE O.M.VIRT.',I3)
   16 FORMAT(1H0,20X,F15.8)
   20 FORMAT(1H0/1H0'ORBITALES INITIALES')
   25 FORMAT(1H0/1H0'ENERGIE ORBITALE',I4)
   40 FORMAT(40I2)
   50 FORMAT(1H0,F15.8)
  300 FORMAT(16I5)
 9999 FORMAT('0LECTURE EFFECTUEE')
      LEC=1
      IMP=3
      READ(LEC,5)NA,NO,NV
      NT=NO+NV
      READ(LEC,40)CAMX,(JMIN(I),JMAX(I),I=1,CAMX)
      READ(LEC,40)(IP(I),I=1,NA)
      N4=1
      N7=1
      DO 305 I=1,NT
      READ(IB10'N7)(C(I,JA),JA=1,NA)
      WRITE(IB7'N4)(C(I,JA),JA=1,NA)
  305 CONTINUE
      N9=1
      READ(IB12'N9)(E(JA),JA=1,NA)
      WRITE(IB7'N4)(E(JA),JA=1,NA)
      WRITE(IMP,15)NA,NO,NV
      WRITE(IMP,20)
      DO 30  I=1,NT
      WRITE(IMP,25)I
      WRITE(IMP,50)E(I)
```

```
      DO 31 J=1,NA
      WRITE(IMP,16)C(I,J)
   31 CONTINUE
   30 CONTINUE
      N5=1
      WRITE(IB8'N5)CAMX
      DO 200 II=1,CAMX
  200 WRITE(IB8'N5)(JMIN(II),JMAX(II))
C    TRANSFERT
      N6=1
      DO 41 IZU=1,4
      DO 42 JZU=1,NDIM
      READ(IB9'N6)(EE(KZU),KZU=1,NDIM)
      WRITE(IB7'N4)(EE(KZU),KZU=1,NDIM)
   42 CONTINUE
   41 CONTINUE
      N6=1
      WRITE(IB3,9999)
      RETURN
      END
   PHASE DOUADY2,DOUADY1
```

```
// EXEC FORTRAN
      SUBROUTINE ROSSI
      DIMENSION C(32,32),DIP(32,32,3),X(32,32),IP(32)
      DIMENSION Y(32,32),Z(32,32)
      DIMENSION AA(3),AB(3),BB(3)
      EQUIVALENCE (DIP(1),X(1)),(DIP(1025),Y(1)),(DIP(2049),Z(1))
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION AUX1,AUX2,AUX3,VAL,A,B,X0,CO,DIP,X,Y,Z,AA,AB,BB
      DOUBLE PRECISION SI
 1000 FORMAT('0A= ',E15.8,10X,'B= ',E15.8)
 9999 FORMAT('0ROSSI EFFECTUE')
      N6=1
      DO 200 IX=1,NDIM
  200 READ(IB9'N6)(X(IX,JX),JX=1,NDIM)
      DO 300 IY=1,NDIM
  300 READ(IB9'N6)(Y(IY,JY),JY=1,NDIM)
      DO 400 IZ=1,NDIM
  400 READ(IB9'N6)(Z(IZ,JZ),JZ=1,NDIM)
      X1=0.0
      IMIN1=IMIN+1
      DO 130 I=IMIN1,IMAX
      MAX=I-1
      DO 120 J=IMIN,MAX
      DO 50 K=1,3
      AA(K)=0.0
      AB(K)=0.0
   50 BB(K)=0.0
      DO 80 L=1,NA
      DO 70 M=1,NA
      AUX2=C(I,L)*C(J,M)
      AUX1=C(J,L)*C(J,M)
      AUX3=C(I,L)*C(I,M)
      DO 60 K=1,3
      VAL=DIP(L,M,K)
      AA(K)=AA(K)+AUX1*VAL
      AB(K)=AB(K)+AUX2*VAL
   60 BB(K)=BB(K)+AUX3*VAL
   70 CONTINUE
   80 CONTINUE
      A=0.0
```

```
      B=0.0
      DO 255 K=1,3
      AUX1=AA(K)-BB(K)
      AUX2=AB(K)
      A=A+AUX1*AUX2
  255 B=B+(AUX1**2)*0.25-AUX2**2
      IF(DABS(A/B)-1.D+60)132,133,133
  133 IF(A/B)134,134,135
  134 XO=-0.7853981634/2.0
      GO TO 136
  135 XO= 0.7853981634/2.0
      GO TO 136
  132 XO=DATAN(A/B)*0.25
  136 CONTINUE
      IF(B)137,137,138
  137 XO=XO+0.7853981634
  138 CONTINUE
   90 IF(DABS(XO)-X1)100,100,95
   95 X1=DABS(XO)
C ..........ROTATION.........
  100 CO=DCOS(XO)
      SI=DSIN(XO)
      DO 110 L=1,NA
      AUX1=C(J,L)
      C(J,L)=CO*C(J,L)+SI*C(I,L)
      C(I,L)=-SI*AUX1+CO*C(I,L)
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      WRITE(IB3,9999)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE NEWTO
      DIMENSION C(32,32),R(32,32),DX(32,32),E1(32,3),EX(32),EY(32)
      DIMENSION EZ(32),A(3),B(3),DIP(32,32,3),IP(32)
      DIMENSION X(32,32),Y(32,32),Z(32,32)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      COMMON R
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION R
      DOUBLE PRECISION AUX1,AUX2,AUX3,X0,AX,BX,AY,BY,AZ,BZ,Z,Y,X,E1,EX,
     1EY,EZ,A,B,DIP
      DOUBLE PRECISION XON,X0D
      DOUBLE PRECISION DX
      EQUIVALENCE (DIP(1),X(1)),(DIP(1025),Y(1)),(DIP(2049),Z(1))
      EQUIVALENCE (AX,A(1)),(AY,A(2)),(AZ,A(3))
      EQUIVALENCE (BX,B(1)),(BY,B(2)),(BZ,B(3))
      EQUIVALENCE (E1(1),EX(1)),(E1(33),EY(1)),(E1(65),EZ(1))
 9999 FORMAT('ONEWTON EFFECTUE')
      N6=1
      N8=1
      DO 36 N=1,3
      DO 200 IX=1,NDIM
  200 READ(IB9'N6)(X(IX,JX),JX=1,NDIM)
      DO 35 I=IMIN,IMAX
      AUX3=0.0
      DO 30 L=1,NA
      AUX2=0.0
      DO 25 M=1,NA
   25 AUX2=AUX2+C(I,M)*X(L,M)
      DX(I,L)=AUX2
   30 AUX3=AUX3+DX(I,L)*C(I,L)
   35 E1(I,N)=AUX3
      DO 300 ID=1,NA
  300 WRITE(IB11'N8)(DX(ID,JD),JD=1,NA)
   36 CONTINUE
      N8=1
      DO 400 IX=1,NA
  400 READ(IB11'N8)(X(IX,JX),JX=1,NA)
      DO 500 IY=1,NA
  500 READ(IB11'N8)(Y(IY,JY),JY=1,NA)
      DO 600 IZ=1,NA
```

```
  600 READ(IB11'N8)(Z(IZ,JZ),JZ=1,NA)
      X1=0.0
      IMIN1=IMIN+1
      DO 100 I=IMIN1,IMAX
      MAX=I-1
      DO 90 J=IMIN,MAX
      DO 40 K=1,3
      A(K)=E1(I,K)-E1(J,K)
   40 B(K)=0.0
      DO 45 L=1,NA
      AUX1=C(J,L)
      DO 45 K=1,3
   45 B(K)=B(K)+DIP(I,L,K)*AUX1
      XON=BX*AX+BY*AY+BZ*AZ
      XOD=4.0*(BX**2+BY**2+BZ**2)-AX**2-AY**2-AZ**2
      IF(DABS(XON)-1.D-70)103,103,106
  106 CONTINUE
      IF(DABS(XON)-1.D-40)101,101,102
  101 IF(DABS(XOD)-1.D+30)104,103,103
  103 XO=0.0
      GO TO 105
  102 IF(DABS(XOD)-1.D+30)104,103,103
  104 XO=XON/XOD
  105 CONTINUE
C.......MATRICE ROTATION.............
      R(I,J)=XO
      R(J,I)=-XO
      IF(DABS(XO)-X1)90,90,50
   50 X1=DABS(XO)
   90 CONTINUE
  100 CONTINUE
      WRITE(IB3,9999)
      RETURN
      END
```

```
// EXEC FORTRAN
      SUBROUTINE ORTHO
      DIMENSION C(32,32),F(32,32),R(32,32),IP(32)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      COMMON R
      DOUBLE PRECISION X1,C
      DOUBLE PRECISION R
      DOUBLE PRECISION VAL,XO,F
      DOUBLE PRECISION XVAL,YVAL,PVAL
 9998 FORMAT('0O. DE C=',I2,2X,'O. DE F=',I2,2X,'O. DE C*F=',I2,2X,
     1'O. DE VAL=',I2)
 9999 FORMAT('0ORTHO EFFECTUE')
C.........ROTATION.........
      DO 115 I=IMIN,IMAX
      R(I,I)=1.0
      DO 110 L=1,NA
      VAL=0.0
      DO 105 J=IMIN,IMAX
      VAL=VAL+R(J,I)*C(J,L)
  105 CONTINUE
      F(I,L)=VAL
  110 CONTINUE
  115 CONTINUE
      DO 130 I=IMIN,IMAX
      DO 130 L=1,NA
  130 C(I,L)=F(I,L)
    1 DO 15 L=1,NA
      DO 10 M=1,NA
      F(L,M)=0.0
      DO 5 I=IMIN,IMAX
      F(L,M)=F(L,M)+C(I,L)*C(I,M)
    5 CONTINUE
   10 CONTINUE
   15 CONTINUE
      X1=0.0
      DO 50 I=IMIN,IMAX
      DO 45 L=1,NA
      VAL=0.0
      DO 20 M=1,NA
      XVAL=C(I,M)
      CALL OVERFL(JXVAL)
```

```
      YVAL=F(M,L)
      CALL OVERFL(JYVAL)
      PVAL=XVAL*YVAL
      CALL OVERFL(JPVAL)
      VAL=VAL+PVAL
      CALL OVERFL(IVAL)
      JXYPV=JXVAL*JYVAL*JPVAL*IVAL
      IF(JXYPV-16)200,20,200
  200 WRITE(IB3,9998)JXVAL,JYVAL,JPVAL,IVAL
   20 CONTINUE
      X0=DABS(C(I,L)-VAL)
      IF(X0-X1)30,30,25
   25 X1=X0
   30 R(I,L)=1.5*C(I,L)-0.5*VAL
   45 CONTINUE
   50 CONTINUE
      DO 55 I=IMIN,IMAX
      DO 55 L=1,NA
   55 C(I,L)=R(I,L)
      IF(X1-5.0D-8)60,60,1
   60 CONTINUE
      WRITE(IB3,9999)
      RETURN
      END
  PHASE DOUADY3,DOUADY2
```

```
// EXEC FORTRAN
      SUBROUTINE RESUL
      INTEGER P
      DIMENSION C(32,32),Q(32,32),TMIN(32,32),E(32),F(32),IP(32)
      COMMON IMIN,IMAX,IP,NA,NO,NV
      COMMON N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      COMMON IB1,IB2,IB3,IB4,IB5,IB6,IB7,IB8,IB9,IB10,IB11,IB12,IB13
      COMMON IB14
      COMMON NDIM
      COMMON X1,C
      DOUBLE PRECISION C,Q,TMIN,X1
      DOUBLE PRECISION AUX1,AUX2,E,F
    5 FORMAT(1HO/1HO,'ORBITALES MOLECULAIRES LOCALISEES')
   10 FORMAT(1HO/1HO,' ENERGIE ORBITALE' ,I4)
   15 FORMAT(1HO,F15.8)
   16 FORMAT(1HO,20X,F15.8)
 1000 FORMAT(5F9.6)
 1010 FORMAT(80X)
 1020 FORMAT(F10.6)
 9999 FORMAT('ORESULTAT EFFECTUE')
      NT=NO+NV
      N9=1
      READ(IB12'N9)(E(IE),IE=1,NT)
      DO 200 IT=1,NA
  200 READ(IB12'N9)(Q(IT,JT),JT=1,NA)
      DO 80 I=1,NT
      AUX2=0.0
      DO 70 J=1,NA
      AUX1=0.0
      DO 60 P=1,NA
   60 AUX1=AUX1+C(I,P)*Q(J,P)
   70 AUX2=AUX2+E(J)*AUX1*AUX1
   80 F(I)=AUX2
      N7=NDIM+1
      DO 300 IS=1,NA
  300 READ(IB10'N7)(TMIN(IS,JS),JS=1,NA)
      DO 30 I=1,NA
      DO 30 J=1,NA
      AUX1=0.0
      DO 25 K=1,NA
   25 AUX1=AUX1+C(I,K)*TMIN(K,J)
   30 Q(I,J)=AUX1
      WRITE(IB3,5)
      DO 20 I=1,NT
      WRITE(IB3,10)I
      WRITE(IB3,15)F(I)
```

```
      DO 21 L=1,NA
      WRITE(IB3,16)Q(I,L)
   21 CONTINUE
   20 CONTINUE
      N4=1
      N7=1
      DO 40 I=1,NT
      READ(IB7'N4)(Q(I,J),J=1,NA)
      WRITE(IB10'N7)(Q(I,J),J=1,NA)
   40 CONTINUE
      N9=1
      READ(IB7'N4)(E(I),I=1,NT)
      WRITE(IB12'N9)(E(I),I=1,NT)
      N6=1
      DO 42 K=1,4
      DO 41 I=1,NDIM
      READ(IB7'N4)(Q(I,J),J=1,NDIM)
      WRITE(IB9'N6)(Q(I,J),J=1,NDIM)
   41 CONTINUE
   42 CONTINUE
      WRITE(IB3,9999)
      RETURN
      END
// EXEC LNKEDT
// ASSGN SYS002,X'192'
// ASSGN SYS003,X'192'
// ASSGN SYS004,X'192'
// ASSGN SYS005,X'192'
// ASSGN SYS006,X'192'
// ASSGN SYS007,X'192'
// ASSGN SYS008,X'192'
// ASSGN SYS009,X'192'
// ASSGN SYS010,X'192'
// DLBL UOUT,'M NE PAS PROTEGER U',68/001
// EXTENT SYS002,6,,,1120,200
// EXEC CLRDSK
// UCL B=(K=0,D=280),X'00',OY
// END
// DLBL IJSYS10,'D NE PAS PROTEGER 1',68/001
// EXTENT SYS010,6,,,1100,10
// DLBL IJSYS02,'O NE PAS PROTEGER 2',68/001
// EXTENT SYS002,6,,,1110,10
// DLBL IJSYS03,'U NE PAS PROTEGER 3',68/001
// EXTENT SYS003,6,,,1120,200
// DLBL IJSYS04,'A NE PAS PROTEGER 4',68/001
```

```
// EXTENT SYS004,6,,,1320,200
// DLBL IJSYS05,'D NE PAS PROTEGER 5',68/001
// EXTENT SYS005,6,,,1520,100
// DLBL IJSYS06,'Y NE PAS PROTEGER 6',68/001
// EXTENT SYS006,6,,,1620,20
// DLBL IJSYS07,'J NE PAS PROTEGER 7',68/001
// EXTENT SYS007,6,,,1640,10
// DLBL IJSYS08,'E NE PAS PROTEGER 8',68/001
// EXTENT SYS008,6,,,1650,10
// DLBL IJSYS09,'R NE PAS PROTEGER 9',68/001
// EXTENT SYS009,6,,,1660,20
// EXEC
 10  6  4
 2 1 6 710
 0 0 0 0 0 0 0 0 0 0 0 0
/&
// JOB DOUADY   0125582100101072,COP1   DIAGO  DE S POUR BOYS
// OPTION LINK
```