

# 分子軌道プログラムのTrue BASICへの移植

榊原 正明・高見 和邦・立花 良一・村畑 太郎

物質工学科

(1996年 8月28日受理)

Conversion of MO programs to True BASIC

by

Masaaki SAKAKIBARA, Kazukuni TAKAMI, Ryouichi TATIBANA  
and Taro MURAHATA

Department of Materials Science

(Received August 28, 1996)

Programs of overlap integrals and extended Huckel MO can be coded using external subroutines by True BASIC without using line numbers and go to statements. The use of MAT statements are found to be effective in expressing matrix operations.

Key words : True BASIC, Extended Huckel MO, Overlap integral

## 1. 初めに

われわれは前回、True BASICが主にNECのBASICとどのように違うのかについて報告した<sup>1)</sup>。今回はTrue BASICの適用例として、分子軌道のうち拡張ヒュッケル法による分子軌道計算プログラムと、重なり積分を表示するプログラムのTrue BASICへの変換について報告する。

## 2. 重なり積分の表示

2つの原子軌道(AO)の間の重なり積分は分子軌道法において重要である。ここではスレーター型の原子軌道を使用する。

スレーター型の原子軌道を表すには $\mu = Z^*/n$ の値が必要である。ただし $Z^*$ は有効核電荷を示す。

表示するのは、C, N, O, F原子の2pAOの間の重なり積分である。

2p $\sigma$ 結合と2p $\pi$ 結合では計算式が次のように異なる<sup>2)</sup>。

$$\sigma \text{ 結合} \quad (1/16) \alpha^{5/2} \beta^{5/2} (A_0 B_2 + A_2 (B_0 + B_4) - A_4 B_2)$$

$$\pi \text{ 結合} \quad (1/32) \alpha^{5/2} \beta^{5/2} (A_0 (B_2 - B_4) - A_2 (B_0 - B_4) + A_4 (B_0 - B_2))$$

2つの原子が同一種類であるかどうかで、 $B_0, B_2, B_4$ の値が異なる。

変数atom1, atom2には原子の種類を示す数字が読み込まれ、変数housikiには $\sigma$ 結合では1、 $\pi$ 結合では2の数字が読み込まれる。

グラフィックについてはSET WINDOWで縦方向の正の向きを下から上に設定している。なおcolor 8では色が見にくいいためcolor 13に変更した。

原子間距離の変化に対する重なり積分の値を計算して配列sに代入しておき、重なり積分の値を画面に表示するようにした。

True BASICはPC98上のwindow95で実行することはできるが、完全には対応していないので、画面上の図を他の画面に貼り付けることはできない。

なお2sと、H原子の1sとの重なり積分についても、さらに計算式<sup>2)</sup>を追加すれば同じように表すことができる。

## 重なり積分表示プログラム

```
! 2P原子軌道どうしの重なり積分
OPTION ROLET
SET MODE "TEXT"
DIM s(12,151),vmyu(5),dai$(12),num(12)
bohrr=.529772 ! ボーア半径
FOR i=1 TO 5
  READ vmyu(i)
NEXT i
DATA 1,3,1,625,1,95,2,275,2,60 ! H,C,N,O,F
!
DATA "2P sigma (C)-2P sigma (C)----- 1" , 2,2,1
DATA "2P sigma (C)-2P sigma (N)----- 2" , 2,3,1
DATA "2P sigma (C)-2P sigma (O)----- 3" , 2,4,1
DATA "2P pi (C)-2P pi (C)----- 4" , 2,2,2
DATA "2P pi (C)-2P pi (N)----- 5" , 2,3,2
DATA "2P pi (C)-2P pi (O)----- 6" , 2,4,2
DATA "2P sigma (N)-2P sigma (N)----- 7" , 3,3,1
DATA "2P sigma (N)-2P sigma (O)----- 8" , 3,4,1
DATA "2P sigma (O)-2P sigma (O)----- 9" , 4,4,1
DATA "2P pi (N)-2P pi (N)-----10" , 3,3,2
DATA "2P pi (N)-2P pi (O)-----11" , 3,4,2
DATA "2P pi (O)-2P pi (O)-----12" , 4,4,2
!
FOR patti=1 TO 12
  count=0
  READ dai$(patti),atom1,atom2,housiki
  FOR r=1 TO 2.5 STEP 0.01
    count=count+1
    rrr=r/bohrr
    a=vmyu(atom1)*rrr
    b=vmyu(atom2)*rrr
    p=.5*(a+b)
    q=.5*(a-b)
    a0=exp(-p)/p
    a1=a0+a0/p
    a2=a0+2*a1/p
    a3=a0+3*a2/p
    a4=a0+4*a3/p
    IF atom1<>atom2 THEN
      g1=-exp(-q)/q
      g2=exp(q)/q
      b0=g1+g2
      b1=g1-g2*b0/q
      b2=g1+g2+2*b1/q
      b3=g1-g2+3*b2/q
      b4=g1+g2+4*b3/q
    ELSE ! 同じ種類の原子間の場合
      b0=2
      b1=0
      b2=2/3
      b3=0
      b4=.4
    END IF
    SELECT CASE housiki
    CASE 1 ! sigma の場合
      s(patti,count)=(1/16)*sqr((a+b)^5)+(-a4*b2+a2*(b0+b4)-a0*b2)
    CASE 2 ! pi の場合
      s(patti,count)=(1/32)*sqr((a+b)^5)*(a4*(b0-b2)+a2*(-b0+b4)+a0
      *(b2-b4))
    END SELECT
  NEXT r
NEXT patti
!
```

```

PRINT "どの組み合わせですか"
FOR i=1 to 12
  PRINT dai$(i)
NEXT i
INPUT PROMPT "HOW MANY PATTERNS ?":many
FOR i=1 TO many
  PRINT "( :1: )=":
  INPUT num(i)
NEXT i
!
CLEAR
SET MODE "GRAPHICS"
SET COLOR 15
SET WINDOW 0, 9, 2, 57, -0, 07, 0, 55
PLOT 1, 0, 55; 1, -0, 1
PLOT 0, 90, 0; 2, 55, 0
FOR y=0 TO 0, 55 STEP 0, 02
  PLOT 1, y; 1, 005, y
NEXT y
FOR x=1 TO 2, 55 STEP 0, 1
  PLOT x, 0; x, 0, 005
NEXT x
FOR ym=0 TO 0, 5 STEP 0, 1
  READ Y$
  PLOT TEXT, AT 0, 92, ym+0, 01: Y$
  PLOT 1, ym; 1, 015, ym
NEXT ym
DATA 0, 0, 0, 1, 0, 2, 0, 3, 0, 4, 0, 5
FOR xm=1 TO 2, 5 STEP 0, 5
  READ x$
  PLOT TEXT, AT xm, -0, 03: x$
  PLOT xm, 0; xm, 0, 012
NEXT xm
DATA 1, 0, 1, 5, 2, 0, 2, 5
FOR i=1 TO many
  j=num(i)
  IF j=8 THEN jc=13 ELSE jc=j
  SET COLOR jc
  SET CURSOR i+1, 50
  PRINT dai$(j)
  count=0
  FOR r=1 TO 2, 5 STEP 0, 01
    count=count+1
    PLOT r, s(j, count)
  NEXT r
NEXT i
END

```

### 3. 拡張ヒュッケル法 (E H M O 法) の概要

拡張ヒュッケル法は、 $\pi$ 電子系について成功を取めたヒュッケルの方法を $\sigma$ 電子を含む系に拡張して適用したもので、すべての原子価電子 ( $\sigma$ 電子、 $\pi$ 電子、および $n$ 電子)の状態を統一的に取り扱える。これは、R. Hoffmannによって広く有機化合物や金属錯体などの問題に 응용されて成功を取め、軌道対称性の保存則を導くための有力な理論的武器となったことで知られている。

計算機の高速化により、Ab-initio法を含めたSCF-

MO法が手軽に利用できるようになった現在では、拡張ヒュッケル法が用いられる機会も少なくなってきている。しかし、原子間の積分の評価に必要とされるのはそれぞれの原子パラメータだけであるため、対象原子種を増やしてもパラメータの数が膨大にならず、パラメータの値を決定しやすいという点から、金属錯体の計算などに利用されている。

拡張ヒュッケル法は、単純ヒュッケル法とは異なり、非隣接軌道間の相互作用を考慮にいれた結果、電子間相互作用をあらわに考慮しないため、分子構造の安定性の議論に際しては注意が必要であるが、異性体などの分子の立体的な相違に基づくエネルギー差を求めることができる。また、この方法は重なり積分を無視しないので、より正しい波動関数の分布を与えるという特色がある。

以下簡単にそのあらましを説明する。通常のLCAO-MO法と同様に、基底関数としては、外殻電子の原子軌道の一次結合をとり、分子軌道 $\psi_i$ を原子軌道 $\phi_r$ の一次結合で表わす。

$$\psi_i = \sum C_{ir} \phi_r \quad (1)$$

これを、ハミルトニアンを一電子ハミルトニアンとの和で表わしたSchrödingerの波動方程式に代入すると、変分法により解くべき永年方程式は、以下ようになる。

$$|H_{rs} - S_{rs} \epsilon| = 0 \quad (2)$$

重なり積分 $S_{rs}$ の値はSlater型原子軌道を用いて計算する。

積分パラメータの値は、実験値や理論式に基づいて系統的に定める。クーロン積分 $H_{rr}$ は原子軌道のイオン化ポテンシャルの値 $I_r$ から次式で求める。

$$H_{rr} = -I_r \quad (3)$$

また共鳴積分は、通常次のようなWolfsberg-Helmholtz近似による式で評価される。

$$H_{rs} = \frac{1}{2} K \times S_{rs} \cdot (H_{rr} + H_{ss}) \quad (4)$$

H行列の固有値と固有ベクトルがそのまま分子軌道のエネルギーと係数に対応していないため、永年方程式を解く手続きはやや複雑になるが、SCFの繰り返し計算が必要でないため計算時間は短くて済むことになる。

ここで永年方程式を解く手順を説明すると

$$HC = SC \epsilon \quad (5)$$

(なおSが単位行列の場合は単純ヒュッケル法となる)。

$S^{-1}HC = C \epsilon$ となり、 $S^{-1}H$ を対角化すれば、固有値と固有ベクトルがそのまま分子軌道のエネルギーと係数になる。しかし $S^{-1}$ とHとは対称行列であるが、その積 $S^{-1}H$ が対称行列にならないために以下のように行う。

Sを対角化して、固有値行列(固有値を対角要素とする行列)をM、固有ベクトルを要素とする行列をVとする。

$$SV = VM \quad \text{より} \quad S = VMV^{-1} \quad (6)$$

(6)式を(5)式に代入して

$$HC = VMV^{-1}C \epsilon \quad (7)$$

両辺の左から $M^{-1/2}V^{-1}$ 、右から $C^{-1}VM^{-1/2}$ をかける。

$$M^{-1/2}V^{-1}HVM^{-1/2} = M^{1/2}V^{-1}C \epsilon C^{-1}VM^{-1/2} \quad (8)$$

ここで

$A = M^{-1/2}V^{-1}HVM^{-1/2}$ 、 $W = M^{1/2}V^{-1}C$  とすれば(8)式は  $A = W \epsilon W^{-1}$ となる。

つまり、Aの固有値はエネルギーに対応することになる。しかも行列Vの転置行列 $V^T$ は $V^{-1}$ と等しいので、行列Aは対称行列である。

また係数を要素とする行列  $C = VM^{-1/2}W$  と表せる。全電子エネルギー(eV)は軌道エネルギー $\epsilon_i$ の和となる。

$$E = 2 \sum \epsilon_i \quad (9)$$

Atomic population M<sub>a</sub>と原子核の電荷との差が、その原子の正味の電荷 Net charge となる。

#### 4. 拡張ヒュッケル法プログラムの概要

プログラムの変換は菊池<sup>2)</sup>によるFORTRANプログラムに基づいて行った。ただし、True BASICのプログラムでは行番号は使用せず、goto文も使用していない。

プログラムは外部サブルーチンを用いて行い、結果をファイルに書き出すようにした。メインと外部サブルーチン、また外部サブルーチンの間でもファイルを共通に使用するために引数に #1 を用いている。

配列の大きさの変更については、初めの設定より大きくしても問題はないので、最初に設定する配列の大きさを小さくした。

対角化には菊池<sup>2)</sup>と同じように JACOBI法を用いた。

取り扱える原子はH, C, N, O, Fである。

$A = M^{-1/2}V^{-1}HVM^{-1/2}$  の計算プログラム

```
MAT vt = trn(V)      Vの転置行列
MAT a = sdiag*vt
MAT a = a#h
MAT a = a*v
MAT a = a*sdiag
```

$C = VM^{-1/2}W$  の計算プログラム

```
MAT c = v*sdiag
MAT c = c#w
```

行列 sdiag は対角要素のみからなり、1次元の配列としても取り扱える。しかし MAT 命令では本来の2次元での取り扱いになるためにメモリの無駄遣いにはなっている。MAT 命令を用いることによって行列演算が上記のように簡単に記述でき、わかりやすいことがわかる。

MAT PRINT USING 文では数値ごとに改行されるので、PRINT USING 文を代わりに使用している。

分子名、原子数、電子数、原子の種類、x y z 座標はファイルから読み込む。MAT INPUT文を用いて入力する原子の種類、座標のデータは1行でなければいけない。なお変数flagに2を代入すれば文献<sup>1)</sup>での入力方法もできる。この入力方法では各原子ごとにデータ入力の方式が3通りある。

入力方法1は座標を読み込ませる。

入力方法2は3個の参照原子と長さ2種類の角度か

ら座標を計算する。

入力方法3は参照原子と座標の値が同じ(あるいは符号を変えたもの)ことを利用する。

#### 読み込みファイルの例(メタノール)

```
6, 14          !原子数と電子数
2, 4, 1, 1, 1, 1 !原子の種類
2             !flagの値
1             !入力方法1
0, 0, 0       !座標
1
1. 4300, 0, 0
2             !入力方法2
-1, 2, 1      !座標決定のための原子の番号
1. 09, 109, 5, 0 !長さ、結合角、ねじれ角
2
-1, 2, 1
1. 09, 109, 5, 120
3             !入力方法3
4             !参照原子
1, 2, -3      !X, Y, Z座標
2
3, 1, 2
0. 96, 110, 180
```

実行結果の確認は文献<sup>21) 41) 51) 61)</sup>で行った。

ここで注意しなくてはならないのは、イオン化ポテンシャルの値の違いである。文献<sup>21) 41) 51)</sup>は同じ値を使用している。しかし文献<sup>31) 61) 71)</sup>では前者と別の値を使用している。プログラムにおいて、イオン化ポテンシャルを読み込むデータ文に前者の値には"古いデータ"、後者の値には"新しいデータ"と注釈をつけている。

先ほど述べたスレーター型の原子軌道を表す $\mu$ の値の違い(HとF原子)にも注意しなくてはならない。この場合もプログラムにおいて、読み込みのデータ文に注釈をつけている。

文献<sup>11)</sup>ではボーア半径の値として0.529167が用いられていたが、今回は、現在使われている0.529772を使用した。

文献<sup>31)</sup>を参考にして、Ab-initio MO法と同じような出力方式にした。

今後の改良点としては、結果のグラフィック表示、汎用性のあるサブルーチンをライブラリにすること、文献<sup>71)</sup>のプログラムを参考にして3d原子軌道まで取り扱えるようにし、SCC機能を持たせることが考えられる。

## 5. まとめ

重なり積分と拡張ヒュッケル法のプログラムをTrue BASICへ変換した。行番号、goto文は使用せずに構造化プログラムの作成を行った。サブルーチンは外部サブルーチンだけを使用した。この結果、MAT文を使用すれば行列演算が簡潔に表現できることがわかった。

さらに重なり積分の原子間距離による変化を画面上に容易に表示することができるようになった。

単純ヒュッケル法と違って、拡張ヒュッケル法では全ての共鳴積分と重なり積分とを考慮している。したがって拡張ヒュッケル法は定性的な面でMOの教育に有効であると考えられる。

## 参考文献

- 1) 榎原正明: "True BASICにおける特異性"、鳥取大学工学部研究報告、26巻、pp.167-173(1995)。
- 2) 菊池 修: "分子軌道法"、講談社(1971)。
- 3) 西本吉助、今村詮編、山辺信一、北浦和夫: "分子設計のための量子化学"第二部第1章 講談社サイエンティフィック(1989)。
- 4) 務台 潔: "量子化学"、昭晃堂
- 5) 米沢貞次郎、永田親義、加藤博史、今村詮、諸熊奎治: "三訂量子化学入門(上)化学同人(1983)。
- 6) 大澤映二編、木原 寛: "分子軌道法"第1部第2章 講談社サイエンティフィック(1994)。
- 7a) J. Howell, A. Rossi, D. Wallace, K. Haraki, and R. Hofmann: QCPE, 11, 344(1977)。
- 7b) 木原 寛、相川 泰: JCP Program No. 85, (1993)。
- 8) 菊池 修: "BASICによる化学"、共立出版(1984)。

## 拡張ヒュッケル法

```
!拡張ヒュッケル分子軌道法  EXTENDED HUCKEL MO METHOD
! 1996/8/19 ファイル から データを読み込む
OPTION NOLET
DIM n1(10), n2(10), salom(10), co(10, 3), r(10, 10), s(10, 10), b(10, 10)
DIM v(10, 10), vt(10, 10), h(10, 10), a(10, 10), w(10, 10), c(10, 10), reop(10, 10)
DIM aobp(10, 10), abp(10, 10), sdiag(10, 10), rdiag(10, 10), pote(5, 5), e(10, 10)
DIM vmyu(5), reoop(10, 10)
CLEAR
```

```

INPUT prompt "file name=":f1$
! 書き出しファイル
OPEN #1: NAME "B:Yeh" & f1$ & ".TRU", CREATE NEWOLD.ORG TEXT, ACCESS OUTIN
ESS OUTIN
ERASE #1
! 読み出しファイル
OPEN #2: NAME "B:" & f1$ & ".TRU", CREATE NEWOLD.ORG TEXT, ACCESS OUTIN
S OUTIN
MAT READ pote ! イオン化ポテンシャルの読み込み
DATA -13.60, 0, 0, 0, 0

DATA 0, -21.40, -11.40, -11.40, -11.40 ! 炭素の新しいデータ
!DATA 0, -21.43, -11.42, -11.42, -11.42 ! 炭素の古いデータ
DATA 0, -26.0, -13.4, -13.4, -13.4 ! 窒素の新しいデータ
!DATA 0, -27.50, -14.49, -14.49, -14.49 ! 窒素の古いデータ
DATA 0, -32.30, -14.80, -14.80, -14.80 ! 酸素の新しいデータ
!DATA 0, -35.30, -17.76, -17.76, -17.76 ! 酸素の古いデータ
DATA 0, -40.0, -18.1, -18.1, -18.1 ! フッ素のデータ
!
FOR i=1 TO 5
  READ vmyu(i) ! ヒュッケルパラメータ
NEXT i
DATA 1.3, 1.625, 1.95, 2.275, 2.425
!DATA 1.0, 1.625, 1.95, 2.275, 2.60 ! 古いデータ
!
INPUT #2: natom, nelec ! 原子数 と 電子数
!配列の大きさの再定義
MAT REDIM satom(natom)
MAT REDIM co(natom, 3)
MAT REDIM r(natom, natom)
MAT REDIM abp(natom, natom)
MAT REDIM reop(natom, natom)
MAT REDIM n1(natom*4)
MAT REDIM n2(natom*4)
!
MAT INPUT #2:satom ! 原子の種類
INPUT #2:flg
SELECT CASE flg
CASE 1
  MAT INPUT #2:co ! 座標の読み込み
CASE 2
  CALL calxyz(natom, co, #2) ! 座標の計算
END SELECT
nocc=nelec/2
index=0
FOR i=1 TO natom
  SELECT CASE satom(i)
  CASE 1 ! 水素原子の場合
    index=index+1
    n1(index)=1
    n2(index)=i
  CASE ELSE ! 水素原子以外の場合
    FOR j=1 TO 4
      index=index+1
      n1(index)=j+1
      n2(index)=i
    NEXT j
  END SELECT
NEXT i
nao=index ! 原子軌道の数
!配列の大きさの再定義
MAT REDIM aobp(nao, nao)
MAT REDIM reop(nao, nao)
MAT REDIM v(nao, nao)
MAT REDIM vt(nao, nao)
MAT REDIM a(nao, nao)

MAT REDIM h(nao, nao)
MAT REDIM sdiag(nao, nao)
MAT REDIM c(nao, nao)
MAT REDIM w(nao, nao)
MAT REDIM s(nao, nao)
MAT REDIM b(nao, nao)
FOR i=1 TO natom
  FOR j=i TO natom
    r(i, j)=sqr((co(i, 1)-co(j, 1))^2+(co(i, 2)-co(j, 2))^2+(co(i, 3)-co(j, 3))^2)
    r(j, i)=r(i, j)
  NEXT j
NEXT i
CALL ovlap(nao, n1(), n2(), satom(), co(), r(), s(), vmyu())
PRINT #1:
PRINT #1: "EXTENDED HUCKEL MO METHOD"
PRINT #1: "INPUT DATA"
PRINT #1: "原子数=":natom, "電子数=":nelec
FOR i=1 TO natom
  PRINT #1: satom(i);
NEXT i
PRINT #1:
PRINT #1: "座標(Coordinate)"
PRINT #1: " X Y Z"
FOR i=1 TO natom
  FOR j=1 TO 3
    PRINT #1, USING"-##.###": co(i, j);
  NEXT j
  PRINT #1:
NEXT i
PRINT
PRINT "EXTENDED HUCKEL MO METHOD"
PRINT "INPUT DATA"
PRINT "原子数=":natom, "電子数=":nelec
FOR i=1 TO natom
  PRINT satom(i);
NEXT i
PRINT
PRINT "座標(Coordinate)"
PRINT " X Y Z"
FOR i=1 TO natom
  FOR j=1 TO 3
    PRINT USING"-##.###": co(i, j);
  NEXT j
  PRINT
NEXT i
PRINT #1:
PRINT #1: " 原子間距離(Atomic Distance)"
PRINT #1:
CALL write2(natom, r(), #1)
! 行列を求める
FOR i=1 TO nao
  FOR j=i TO nao
    n1=n1(i)
    n1j=n1(j)
    n3i=satom(n2(i))
    n3j=satom(n2(j))
    IF i=j THEN
      h(i, i)=pote(n3i, n1i)
    ELSE
      h(i, j)=.875*s(i, j)*pote(n3i, n1i)*pote(n3j, n1j)
      h(j, i)=h(i, j)
    END IF
  NEXT j
NEXT i
PRINT #1:

```

```

PRINT #1:"MUCKEL matrix"
CALL write2(nao, h(.), #1)
PRINT #1:
PRINT #1: " 重なり積分 s (Overlap integral)"
CALL write2(nao, s(.), #1)
MAT b=s
!重なり積分行列sの対角化と固有ベクトルv
CALL diag(b(.), v(.), nao, 5. e-8)
!
FOR i=1 TO nao
  IF b(i, i) < 0 THEN
    PRINT "INSOLUBLE"
    STOP
  END IF
  sdiag(i, i)=1. 0/sqr(b(i, i))
NEXT i
!
MAT vt = lrn(v)           !Vの転置行列
MAT a=sdiag*vt
MAT a = a*b
MAT a = a*v
MAT a= a*sdiag
!行列Aの対角化と固有ベクトルw
CALL diag(a(.), w(.), nao, 5. E-7)
!係数行列cの計算
MAT c = v*sdiag
MAT c = c*w
CALL reag(a(.), c(.), nao)
! AO and AO Bond Population の計算
FOR i=1 TO nao
  FOR j=i TO nao
    IF i=j THEN
      FOR k=1 TO nocc
        FOR l=1 TO nao
          aobp(i, i)=aobp(i, i)+2*c(i, k)*c(l, k)*s(i, l)
        NEXT l
        reaop(i, i)=reaop(i, i)+2*c(i, k)*c(l, k)
      NEXT k
    ELSE
      FOR k=1 TO nocc
        aobp(i, j)=aobp(i, j)+4*c(i, k)*c(j, k)*s(i, j)
        reaop(i, j)=reaop(i, j)+4*c(i, k)*c(j, k)*s(i, j)
      NEXT k
      aobp(j, i)=aobp(i, j)
      reaop(j, i)=reaop(i, j)
    END IF
  NEXT j
NEXT i
! Atomic and Atomic Bond Population の計算
FOR i=1 TO natom
  FOR j=i TO natom
    IF i=j THEN
      FOR k=1 TO nao
        IF n2(k)=i THEN
          abp(i, i)=abp(i, i)+aobp(k, k)
          reop(i, i)=reop(i, i)+reaop(k, k)
        END IF
      NEXT k
    ELSE
      FOR k=1 TO nao
        FOR l=1 TO nao
          IF n2(k)=i AND n2(l)=j THEN abp(i, j)=abp(i, j)+aobp(k,
          reop(i, j)=abp(i, j)
        NEXT l
      NEXT k
      reop(i, j)=abp(i, j)
    END IF
  NEXT j
NEXT i
          reop(j, i)=reop(i, j)
        END IF
      NEXT j
    END IF
  NEXT j
  !全電子エネルギーを求める
  FOR i=1 TO nocc
    total=total+2*a(i, i)
  NEXT i
  PRINT #1:
  PRINT #1: " Reduced Overlap Population matrix"
  CALL write2(natom, reop(.), #1)
  !
  CALL write1(nao, a(.), c(.), #1)
  PRINT #1:
  PRINT #1: " AO and AO Bond population "
  CALL write2(nao, aobp(.), #1)
  PRINT #1:
  PRINT #1: " Atomic and Atomic Bond Population "
  CALL write2(natom, abp(.), #1)
  PRINT #1:
  PRINT #1: " total energy = ";
  PRINT #1: total
  PRINT #1:
  PRINT #1: "   atom         charge         net charge "
  FOR i=1 TO natom
    SELECT CASE satom(i)
    CASE 1
      moji$=" H"
      nodensi=1
    CASE 2
      moji$=" C"
      nodensi=4
    CASE 3
      moji$=" N"
      nodensi=5
    CASE 4
      moji$=" O"
      nodensi=6
    CASE 5
      moji$=" F"
      nodensi=7
    END SELECT
    PRINT #1:i;" : ";moji$, abp(i, i), nodensi-abp(i, i)
  NEXT i
  PRINT #1:
  CLOSE #1
  END
  !
  SUB overlap(nao, n1(), n2(), satom(), co(.), r(.), s(.), vmyu())
  ! 重なり積分
  bohrr=0. 529772
  FOR i=1 TO nao
    s(i, i)=1          ! 対角項を1にする
    FOR j=i+1 TO nao
      IF n2(i) < n2(j) THEN
        IF n1(i) > n1(j) THEN
          n1a=n1(j)
          n1b=n1(i)
          n2a=n2(j)
          n2b=n2(i)
        ELSE
          n1a=n1(i)
          n1b=n1(j)
          n2a=n2(i)
          n2b=n2(j)
        END IF
      NEXT j
    NEXT i
  END SUB

```

```

END IF
n3a=satom(n2a)
n3b=satom(n2b)
a=vmyu(n3a)+r(n2a,n2b)/bohrr
b=vmyu(n3b)+r(n2a,n2b)/bohrr
p=.5*(a+b)
q=.5*(a-b)
a0=exp(-p)/p
a1=a0+a0/p
a2=a0+2*a1/p
a3=a0+3*a2/p
a4=a0+4*a3/p
IF n3a<n3b THEN
  g1=-exp(-q)/q
  g2=exp(q)/q
  b0=g1+g2
  b1=g1-g2+b0/q
  b2=g1+g2+2*b1/q
  b3=g1-g2+3*b2/q
  b4=g1+g2+4*b3/q
ELSE ! 同じ種類の原子間
  b0=2
  b1=0
  b2=2/3
  b3=0
  b4=.4
END IF
SELECT CASE n1a
CASE 1 ! 1s
  SELECT CASE n1b
  CASE 1 ! 1s-1s
    s(i,j)=.25*sqrt((a+b)^3)*(a2+b0-a0*b2)
    s(j,i)=s(i,j)
  CASE 2 ! 1s-2s
    s(i,j)=.0721688*sqrt(a^3*b^5)*(a3*b0-a2*b1-a1*b2+
a0*b3)
    s(j,i)=s(i,j)
  CASE 3,4,5 ! 1s-2p
    x1=(co(n2a,n1b-2)-co(n2b,n1b-2))/r(n2a,n2b)
    s(i,j)=.125*x1*sqrt(a^3*b^5)*(-a3*b1+a2*b0+a1*b3-
a0*b2)
    s(j,i)=s(i,j)
  END SELECT
CASE 2 ! 2s
  SELECT CASE n1b
  CASE 2 ! 2s-2s
    s(i,j)=.0208333*sqrt((a+b)^5)*(a4*b0-2*a2*b2+a0*b
4)
    s(j,i)=s(i,j)
  CASE 3,4,5 ! 2s-2p
    x1=(co(n2a,n1b-2)-co(n2b,n1b-2))/r(n2a,n2b)
    s(i,j)=.0360844*x1*sqrt((a+b)^5)*(-a4*b1+a3*(b0-b
2)+a2*(b1+b3)+a1*(-b2+b4)-a0*b3)
    s(j,i)=s(i,j)
  CASE ELSE
    s(i,j)=s(i,j)
  END SELECT
CASE 3,4,5
  sigma=.0625*sqrt((a+b)^5)*(-a4*b2+a2*(b0+b4)-a0*b2)
  pai=.03125*sqrt((a+b)^5)*(a4*(b0-b2)+a2*(-b0+b4)+a0*(b
2-b4))
  IF n1a=n1b THEN ! p軌道の種類が同じ
    x1=((co(n2a,n1b-2)-co(n2b,n1b-2))/r(n2a,n2b))^2
    s(i,j)=-sigma*x1*pai*(1-x1)
    s(j,i)=s(i,j)
  ELSE ! p軌道の種類が異なる
    x1=((co(n2a,n1a-2)-co(n2b,n1a-2))/(co(n2a,n1b-2)-c
o(n2b,n1b-2))/r(n2a,n2b))^2
    s(i,j)=x1*(sigma+pai)
    s(j,i)=s(i,j)
  END IF
END SELECT
!同一原子内
s(i,j)=0
s(j,i)=s(i,j)
END IF
NEXT j
NEXT i
END SUB
!
! JACOBI METHOD (subroutine program)
SUB diag(a(.),c(.),n,d)
FOR i=1 TO n
  FOR j=1 TO n
    IF i=j THEN
      c(i,j)=1
    ELSE
      c(i,j)=0
    END IF
  NEXT j
NEXT i
DO
  amax=abs(a(2,1))
  FOR i=2 TO n
    i1=i-1
    FOR j=1 TO i1
      IF abs(a(i,j))>amax THEN
        amax=abs(a(i,j))
      END IF
    NEXT j
  NEXT i
  ! IF amax<=d THEN EXIT DO
  sm1=0.1*amax
  FOR j=2 TO n
    j1=j-1
    FOR k=1 TO j1
      IF abs(a(j,k))>sm1 THEN
        akk=a(k,k)
        ajj=a(j,j)
        tanb=(ajj-akk)/(2*a(k,j))
        IF tanb>0 THEN
          tn=tanb+sqrt(1+tanb^2)
        ELSE
          tn=tanb-sqrt(1+tanb^2)
        END IF
        cs=1/sqrt(1+tn^2)
        sn=cs*tn
        FOR l=1 TO n
          IF l=k THEN
            a(k,k)=akk*cs^2+a(j,k)*2*sn*cs+ajj*sn^2
            a(j,j)=akk+ajj-a(k,k)
          ELSE IF l<j THEN
            ak1=a(k,l)
            aj1=a(j,l)
            a(k,l)=ak1*cs+aj1*sn
            a(j,l)=aj1*cs-ak1*sn
            a(l,k)=a(k,l)
            a(l,j)=a(j,l)
          END IF
        NEXT l
        a(j,k)=a(j,k)*(cs^2-sn^2)+(ajj-akk)*sn*cs
        a(k,j)=a(j,k)
      END IF
    NEXT k
  NEXT j
NEXT i

```



```

      FOR i=1 TO n
        cij=c(i,j)
        cik=c(i,k)
        c(i,j)=cij*cs-cik*sn
        c(i,k)=cij*sn+cik*cs
      NEXT i
    END IF
  NEXT k
NEXT j
LOOP until amax<=d
END SUB
!
! REARRANGEMENT OF MATRIX
SUB reag (e(.),c(.),n)
  FOR i=1 TO n
    FOR k=1 TO n
      IF e(k,k)>=e(i,i) THEN
        y=e(i,i)
        e(i,i)=e(k,k)
        e(k,k)=y
      FOR j=1 TO n
        yc(j,i)
        c(j,i)=c(j,k)
        c(j,k)=y
      NEXT j
    END IF
  NEXT k
NEXT i
END SUB
!
! PRINTING OF MO ENERGIES AND LCAO COEFFICIENTS
SUB write1(n,e(.),c(.),#1)
  PRINT #1:
  PRINT #1: "Orbital Energies and LCAO Coefficients"
  PRINT #1:
  index=(n-1)/15+1
  FOR i=1 TO index
    i1=15*(i-1)+1
    i2=15*i
    IF n<i2 THEN i2=n
    FOR k=i1 TO i2
      PRINT #1,USING"-###.####": e(k,k);
    NEXT k
    PRINT #1:
    FOR j=i1 TO n
      FOR k=i1 TO i2
        PRINT #1,USING"-###.####": c(j,k);
      NEXT k
    PRINT #1:
  NEXT j
NEXT i
END SUB
!
! PRINTING OF MATRIX
SUB write2(n,a(.),#1)
  index=(n-1)/15+1
  FOR i=1 TO index
    PRINT #1:
    i1=15*(i-1)+1
    i2=15*i
    IF n<i2 THEN i2=n
    FOR j=1 TO n
      FOR k=i1 TO i2
        PRINT #1,USING "-###.####":a(j,k);
      NEXT k
    PRINT #1:
  NEXT i

```

```

      NEXT j
    END SUB
SUB calxyz(n,co(.),#2) ! 座標計算
  DIM cs(20,3),ib(20,6),px(20),py(20),n3(20)
  DIM zz(20),nz(20),sr(20),cr(3,3),xx(3)
  DIM ir(20,3),ra(20,3),sm(20),sn(20,3),dm(4,3),in(20),ix(20)
  DATA 0,0,0,1,0,0,1,1,0,0,1,0
  FOR i=1 TO 4
    FOR j=1 TO 3
      READ dm(i,j)
    NEXT j
  NEXT i
  FOR i=1 TO n
    PRINT ""
    PRINT "ATOM ";i
    INPUT #2: in(i) ! Data Form
    SELECT CASE in(i)
    CASE 1
      INPUT #2: co(i,1),co(i,2),co(i,3)
    CASE 2
      INPUT #2: ir(i,1),ir(i,2),ir(i,3) ! Reference Atoms
      INPUT #2: ra(i,1),ra(i,2),ra(i,3) ! R. SHITA, PHAI
    CASE 3
      INPUT #2:sm(i) ! Reference Atom
      INPUT #2:sn(i,1),sn(i,2),sn(i,3) ! Symm. Ccoordinate
    END SELECT
  NEXT i
  CLOSE #2
  FOR i=1 TO n
    SELECT CASE in(i)
    CASE 2
      FOR k=1 TO 3
        IF IR(i,k) <= 0 THEN
          FOR l=1 TO 3
            cr(k,l)=dm(ir(i,k)+4,l)
          NEXT l
        ELSE
          IF ix(ir(i,k))=1 THEN
            FOR l=1 TO 3
              cr(k,l)=co(ir(i,k),l)
            NEXT l
          ELSE
            EXIT FOR
          END IF
        END IF
      NEXT k
      rr=ra(i,1)
      a1=ra(i,2)
      a2=ra(i,3)
      CALL xyz (rr,a1,a2,cr,xx)
      FOR j=1 TO 3
        co(i,j)=xx(j)
      NEXT j
    CASE 3
      IF ix(sm(i)) = 1 THEN
        FOR l=1 TO 3
          cl=1
          IF sn(i,l)<0 THEN cl=-1
          co(i,l)=cl*co(sm(i),abs(sn(i,l)))
        NEXT l
      END IF
    CASE ELSE
      END SELECT
    ix(i)=1
  NEXT i

```

```

FOR i=1 TO n
  FOR j=1 TO 3
    co(i, j)=round(co(i, j), 3) ! 小数第4位を四捨五入
  NEXT j
NEXT i
CLEAR
SET CURSOR 1, 10
PRINT "X"
SET CURSOR 1, 20
PRINT "Y"
SET CURSOR 1, 30
PRINT "Z"
FOR i=1 TO n
  SET CURSOR i+2, 3
  PRINT i
  SET CURSOR i+2, 8
  PRINT USING "-##.###":co(i, 1)
  SET CURSOR i+2, 18
  PRINT USING "-##.###":co(i, 2)
  SET CURSOR i+2, 28
  PRINT USING "-##.###":co(i, 3)
NEXT i
END SUB
SUB xyz(rr, a1, a2, cr(, ), xx(, ))
DIM t(3, 3), tt(3, 3), xxx(3)
aa=abs(a1)
IF abs(aa-180) <= .001 THEN ! 結合角が180度の場合
  x5=cr(3, 1)-cr(2, 1)
  y5=cr(3, 2)-cr(2, 2)
  z5=cr(3, 3)-cr(2, 3)
  r5=sqr(x5*x5+y5*y5+z5*z5)
  IF r5 < .0001 then
    PRINT "Impossible to Determine Coordinates"
    STOP
  END IF
  dd=rr/r5
  xx(1)=cr(3, 1)+dd*x5
  xx(2)=cr(3, 2)+dd*y5
  xx(3)=cr(3, 3)+dd*z5
ELSE ! 結合角が180度でない場合
  x0=cr(2, 1)
  y0=cr(2, 2)
  z0=cr(2, 3)
  FOR is=1 TO 3
    cr(is, 1)=cr(is, 1)-x0
    cr(is, 2)=cr(is, 2)-y0
    cr(is, 3)=cr(is, 3)-z0
  NEXT is
  x5=sqr(cr(3, 1)*cr(3, 1)+cr(3, 2)*cr(3, 2))
  r5=sqr(cr(3, 3)*cr(3, 3)+x5*x5)
  IF x5 >= .0001 THEN
    s1=x5/r5
    c1=cr(3, 3)/r5
    s2=cr(3, 2)/x5
    c2=cr(3, 1)/x5
    !
    t(1, 1)=-c1*c2
    t(1, 2)=-c1*s2
    t(1, 3)=s1
    t(2, 1)=s2
    t(2, 2)=-c2
    t(2, 3)=0
    t(3, 1)=s1*c2
    t(3, 2)=s1*s2
    t(3, 3)=c1
  END IF
  FOR is=1 TO 3
    IF is<2 THEN
      FOR k=1 TO 3
        xxx(k)=cr(is, k)
      NEXT k
      MAT xxx=i*xxx
      FOR k=1 TO 3
        cr(is, k)=xxx(k)
      NEXT k
    END IF
    NEXT is
  END IF
  x3=sqr(cr(1, 1)*cr(1, 1)+cr(1, 2)*cr(1, 2))
  IF x3 < .0001 THEN
    PRINT "Impossible to Determine Coordinates"
    STOP
  END IF
  s3=cr(1, 2)/x3
  c3=cr(1, 1)/x3
  s4=sin(pi*a2/180)
  c4=cos(pi*a2/180)
  xx(3)=cr(3, 3)-rr*cos(pi*a1/180)
  x7=abs(rr*sin(pi*a1/180))
  xx(2)=x7*(s4+c3*c4*s3)
  xx(1)=x7*(c4+c3-s4*s3)
  IF x5 >= .0001 THEN
    MAT tt=trn(1) ! tの転置行列
    MAT xx=tt*xx
  END IF
  xx(1)=xx(1)+x0
  xx(2)=xx(2)+y0
  xx(3)=xx(3)+z0
END IF
END SUB

```