Pak. j. sci. ind. res. vol. 33, nos. 1-2, January, February, 1990

HOME COMPUTER IN MOLECULAR ORBITAL CALCULATION
Part II. The Iterative Methods: W–Technique and SCF Methods

M. Mohammad*, A.Y. Khan and R. Qureshi

Department of Chemistry, Quaid-e-Azam University, Islamabad, Pakistan

(Received November 5, 1988; revised December 9, 1989)

Computer programs in BASIC language for iterative molecular orbital methods: ω-technique and self consistant field (SCF) methods have been developed for home computer. These programs can be used for pedagogical as well as research purposes. ω-Technique program can be used for both open and closed shell system while the SCF program can only be used for ground state singlet system. The programs are simple and tractable and can be used by nonprofessionals. Sample calculations are carried out on allyl and pyridine system.

Key words: Home computer, M.O. program, ω-technique, SCF calculations, BASIC programs.

Introduction

In Part I of this series [1], the use of home computer in molecular orbital calculations was described. The simple Huckel method had been described. The description of the computational technique, the main program, the subroutines, and form of input matrix were also given. Results of the calculations on allyl radical and pyridine were given. In this Part II of the series, the theory of the iterative methods- the ω-technique and the self consistent methods, as applied to pi-electron systems and description of the computational techniques with the main program, subroutines, listings and sample calculations are given.

Huckel molecular orbital method is simple and, surprisingly, works. The reason being its empirical nature and use of adjustable parameters. The methods, however, has serious limitations, both theoretical as well as practical. Weaknesses of the Huckel theory will not be discussed here, these are all well known and can be found in standard books [2-5].

To overcome the weaknesses of Huckel method both noniterative (modified Huckel) and iterative methods have been suggested and used [2-5]. The iterative method, the self consistent method, in various prescription, is theoretically sound, and hence, has been extensively used. A simpler method, which could infact be called iterative Huckel method was suggested by Streitwieser [2-5] and has been in use. The method is called ω-technique. Theories of the two methods are given below, however, only those parts are given which are basic and employed in the computer program.

It may be noted that most of the functions and subroutines included in the HMO computer program, published earlier [1] are used in both ω-technique and SCF calculations hence discussions about these subroutines will not be given here. (A user manual is given).

Molecular orbital methods. (i) ω-technique. A simple technique known as ω-technique incorporates interelectronic repulsion terms in the HMO method [1]. According to this technique, the value of the coulomb intergral, should be linearly correlated to the charge. This may be formulated as [2-5].

$$\alpha_r=\alpha_o + (1-q_r)\,\omega\,\beta_o \qquad (1)$$

where q_r is the charge density (or electron density) at an atom r (ref. 1 eqn. 8),α_o and β_o are the coulomb and resonance integrals respectively [1], ω is a dimensionless parameter the value of which may be so chosen as to give the best agreement with experiment. The ω-technique is very useful for calculating HMO energies and charge densities for systems with positive or negative charges or for lattices with uneven charge distribution. It must however be noted that ω-technique is an iterative method: the charge density q_r on which the matrix element α_r (the coulomb integral) depends must be known beforehand. This α_r is needed to solve the secular equation (ref 1 eqns. 2 & 5) from which q_r is obtained. Thus to start with, q_r are obtained from a Huckel calculation which , through eqn. (1), gives new set of α_r from which, on solving the secular problem a new set of q_r is obtained. The cycle is repeated to convergence.

(ii) *Self consistant field calculations* [4-5]. In advanced M.O. method explicit consideration is given to electron repulsions.

Hamiltonian H is written as

$$H = \Sigma_i (H_{core})_i + \Sigma_{i<} \Sigma_j \; 1/r_{ij} \qquad (2)$$

In which i and j refers to electrons.

$$(H_{core})_i = -1/2\nabla_i^2 + V_{(i)} \qquad (3)$$

Various levels of sophistication arise in the formulation of the core Hamiltonion. Treatment of organic compounds, however, has generally been confined to pi-electrons and corresponding cores in which the carbon nuclei are shielded by sigma and inner shell electrons.

The minimization of the variational energy leads to a set of non linear equations known as Roothan's equations. It is assumed that there are N basis functions and 2N electrons in a closed shell system.

$$h_{rs}C_{s} = C \quad S_{rs} C_{s} \qquad (4)$$

where

$$h_{rs} = f_{rs} + \quad _{t} \quad _{u} p_{tu} [(rs/tu) - 1/2(rtisu)] \qquad (5)$$

and

$$\psi = C_{r}\phi_{r} \qquad (6)$$

$$S_{rs} = \int \phi^{*}_{r} \phi_{s} \, dv \qquad (7)$$

$$P_{tu} = 2 \Sigma^{N}_{j=1} C_{tj} C_{uj} \qquad (8)$$

$$f_{rs} = \int \phi_{r} (1) h_{core} \phi_{s} (1) \, dv (1) \qquad (9)$$

$$(rs/tu) = \int \phi^{*}_{r} (1) \phi_{t} (2) (1/r_{12}) \phi_{s} (1) \phi_{u} (2) \, dv \qquad (10)$$

Pople and simulteneously pariser and Parr introduced simplification [4-6] that ultimately reduced Roothan's equation to a form comparable to those of Huckel theory. Firstly, overlap integrals, S_{rs} and all frame work resonance integrals f_{rs} between non-neighbouring conjugated atoms were ignored. Then, for consistency, all electron repulsion integrals that depend upon the overlap of charge clouds were similarly ignored, which leaves as non-zero only the two suffix terms.

$$(r\,r\,|s\,s) \\ = \gamma rs = \iint \phi^{*}_{r} (1) \phi^{*}_{s} (2) \, 1/r_{12} \phi_{r} (1) \phi_{s} (2) \, dv \qquad (11)$$

The equations then become

$$h_{rs} C_{s} = E \, C_{r} \, (r,s = 1,2,- - - - -N) \qquad (12)$$

$$h_{rr} = f_{rr} + 1/2 \, P_{rr} \gamma_{rs} + \quad _{s \neq r} P_{ss} \gamma_{rs} \qquad (13)$$

$$h_{rs} = \beta_{rs} + 1/2 \, P_{rs} \gamma_{rs} \qquad (14)$$

Pople introduced another simplification by approximating γ_{rs} (repulsion integral) and $(r \, |V_{s}| r)$ by the inverse distance law

$$P_{ss} \gamma_{rs} - (r|V_{s}|r) = (P_{ss} - Z_{s}). \, 1/R_{rs} \qquad (15)$$

Where R_{rs} is the distance of separation between atoms r and s and Z_{s} is the effective screened charge at the frame work ion S. Since, infact, the repulsion integrals enter, in part, as adjustable parameters, it has been found more acceptable to write instead.

$$P_{ss} \gamma_{rs} - (r|Vs|r) = (P_{ss} - Zs) \gamma_{rs} \qquad (16)$$

and the final form of the matrix elements h_{rs} becomes

$$h_{rr} = W_{r} + (1/2) P_{rr} \gamma_{rs} + \Sigma_{s,r} \, 1/2 \, (P_{ss} - Z_{s}) \gamma_{rs}$$

h

$$h_{rs} = \beta_{rs} - 1/2 \, P_{rs} \gamma_{rs}$$

which with the reduced form of Roothan's equations

$$h_{rs} C_{s} = \epsilon C_{r} \, (r=1, ... n) \qquad (19)$$

define the SCF equations for n–electrons systems. This system of equations is applicable to pi-systems with no odd electron.

Computational technique. The computer program calculates eigen-values, eigen-vectors, charge densities and bond orders of closed shell electron system. ω (omega)-technique can, however, be used for open shell system also.

(i) ω *-(omega)-technique.* It is basically the same as HMO technique except that after each diagonalization and calculation of charge densities the diagonal elements are calculated from the relationship (1). Thus after each diagonalization charge densities are calculated which are used in the recalculation of diagonal elements. This process is continued till consistant results are obtained.

Both HMO and ω-technique programs use the same subroutines for diagonalization, ordering the eigen-values and eigen-vectors and calculation of charge densities. Only the main program differs. In the ω-technique program a new array G(N) has been created. This array stores the diagonal elements just after the input matrix is keyed in. These diagonal elements are restored to HD (I) just before the recalculation of HD (I) so that the HD(I) now contains the original values instead of the diagonalized ones (eigen–values).

(ii)*SCF calculation.* This program calculates eigen-values, eigen-vectors, charge densities and bond orders of closed shell electron system according to the Roothan's SCF method with the Pariser, Parr and Pople approximation.

This program consists of a main program and is also divided into subroutines[1]. Most of the subroutines are the same as were used in the previous Huckel[1] and ω-technique calculations. Individuald as Subroutines can be defined as:

Input. This subroutine reads N, number of atoms; Array NE (1) and atomic co-ordinates. The input data specifying the atomic co-ordinates is based upon the use of hexagonal grid. Grid as multiples of co-ordinates are defined 31/2 (i.e. 1 Sin 60°) in the x-direction and multiple of 1/2 (i.e.1 Cos 60° Y-direction) 1 being the bond length. This subroutine also initializes the value of array Z (I) as I and array DW(1) as zero. Z(I) is the effective screened charge at the framework ion I and DW (I) represents $\delta W_{r} = W_{r} - W$. W_{r} can be defined as an atomic valence state ionization potential, that can, in principle, be estimated from experimental data. W is the atomic valence state ionization potential of carbon. W_{c} is assumed to be zero for simple hydrocarbons. The INPUT subroutine also takes the values NE (I) i.e. number of electrons in the ith M.O.

Gamma. This subroutine greatly simplifies the input specification by generating the bulk of input data internally. The grid co-ordinates stored in the arrays X(I) and Y(I) are converted to molecular co-ordinates and D, the distance between all pairs of atoms is computed. Matrix elements of the repulsion matrix G(I,I) may be computed as follows [7].

(i) IF D> 2.81, G(I,J) = 14.4/D(Charged sphere model)
(ii) 2.81>D>2.75 G(I,J) = 5.77
(iii) 1.42>D, G(I,J) = 7.19
G(I,I) = 11.35

This matrix G(I,J) contains the value of γ_{ij} expressed in electron volts as proposed by Praiser and Parr for hexagonally disposed atoms. The above mentioned values of repulsion integrals were determined by Parr[4,7] and from charged sphere model. Another option is to use Mataga formulation for repulsion integrals [8]. In this Model

G(I,J) = 14.397/(1.29 + D).

The repulsion integrals thus formed can be inspected and modified if desired. Thus any set of repulsion integrals can be incorporated in the calculations. Another array B(I,J) containing the core integrals is also defined. In this array the upper portion contains the SCF β'S i.e. B(I,J) = -2.37 (ev) if I and J are neighbours or G (I,J)>7 otherwise β (I,J)_ = 0. Values of β (I,J) are transferred to H (I,I) DW (I) are transferred to H (I,I) and (H(D,I), H(I,J)is the matrix which is now diagonalized.

Diagonalization, Order and Bond Order. Subroutines are the same as used for the HMO calculation [1].

SCF. This subroutine constructs a new H matrix using eqns. (17) and (18), which in matrix notations can be written as :

H (I,I) = DW (I) + [(1/2) Q(I) G(I,I) - G(I,I]+ [Q(S)-Z(I)]. G(I,S) (20)

H(I,J) = β(I,J) - 1/2 P(I,J). G (I,J) (21)

where Q(I) and Q(S) are the charge densities.

This matrix is now diagonalized. Again new bond orders and charge densities are calculated and SCF matrix elements recalculated and diagonalized.

This iterative procedure is repeated about 10 times after which, generally, it gives self consistent results.

Form of input matrix. This has already been elaborated in the earlier publication [1]. However, in the present case of SCF calculation input takes a different form. In this case grid coordinates would form the input data. Grid coordinates are defined as multiples of 31/2 in the x-direction and 1/2 in Y-direction.

A subroutine (Printout Input Data SCF) prints out the input data for checking, it is optional.

The other subroutines (Printout results,) have been described before [1].

Acknowledgement. Financial assistance from Pakistan Science Foundation (PSF-C-QU/Chem (137) is gratefully acknowledged.

References

1. M. Mohammad, A.Y. Khan and R. Qureshi, Pak. j. sci. ind. res., **30**, 359, (1987).
2. A. Streitwieser Jr., *"Molecular Orbital Theory for Organic Chemists.* (John Wiley, 1961), pp. 100, 449.
3. T.E. Peacock, *Electronic Properties of Aromatic and Heterocyclic Molecules* (Academic Press 1965), pp. 69.
4. Robert G. Parr, *Quantum Theory of Molecular Electronic Structure*, (W.A. Benjamin, 1963), pp 45.
5. L. Salem, *Molecular Orbital Theory of Conjugated System* (Benjamin, 1966), pp 48.
6. J.A. Pople and L.A. Beveridge, *Approximate Moleculc Orbital Theory* (McGraw Hill, 1970).
7. H.H. Greenwood, *Computing Methods in Orgar Chemistry* (John Wiley and Sons, New York, 197
8. K. Nishimoto and N. Mataga, Z. Phys. Chem., **12**, 335 (1953).

USER'S MANUAL

Software for Molecular Orbital Calculations

On the switching on the computer and inserting the MO calculation diskette a menu (list of programmes on the diskette) appears on the screen. Key in the number of the required programme and press Enter.

ω-Technique. If ω-technique is the chosen progrmme then the title of the programme and a question inquiring whether a testrun is required or not appears on the screen. Answer in yes (Y) or no (N). If the answer is in affirmative then the ω-technique programme would run for pyridine. After a few minutes the results would appear on the screen. In this run there was no need to load the input data as it was already provided to the computer.

After the test run the computer would proceed on the load the input data. If the test run was not required then the computer would directly go on to loading the input data.

Loading the input data. A question about the number of conjugated atoms (N) would apear on the screen. Key in the required number and press enter. For example for benzene or pyridine molecular key in 6. (It must be remember that after keying in each value the enter key must be pressed).

Now the matrix elements of a N x N matrix are to be keyed in. As the matrix is a symmetric matrix only the upper half of the matrix is required. For a benzene or pyridine molecule the display on the screen would now be (These matrix elements appear successively).

H (1, 1) = ?
H (1, 2) = ?
H (1, 3) = ? and so on

Value of each elements has to be given

The question here is that how do we determine the values of these matrix elements.

Key in

H(I,J) = 0 where 1 and J are non neighbouring carbon atoms.

H(I,J) = 1 for neighboiuring carbon atoms.

H(I,I) = 0 where I is a carbon atom.

H(I,I) = h_x for a heteroatom.

H(I,J) = k_{c-x} for neighbouring atoms.

when one of the atom is a heteroatom.

The above mentioned qualities can be clearly demonstrated by considering the case of pyridine.

H(1,1)=0 H(2,2)=0 H(3,3)=0 H(4,4)=) 0.5 H(5,5)=0 H(6,6)=1,

H(1,2)=1 H(2,3)=1 H(3,4)=1 H(4,5)=1 H(5,6)=1

H(1,3)=0 H(2,4)=0 H(3,5)=0 H(4,6)=0

H(1,4)=0 H(2,5)=0 H(3,6)=0

H(1,5)=0 H(2,6)=0

H(1,6)=1

For N atom h_N=0.5 k_{c-x}=1

For allyl radical the input matrix is

H(1,1)=0 H(2,2)=0 H(3,3)=0 C_1=C_2-C_3

H(1,2)=1 H(2,3)=1 H(1,3)=0

The next input which is to be loaded is the number of electrons in each molecular orbital. The display on the screen would now be (for benzene or pyridin molecule)

NE(1) = ?

NE(2) = ?

NE (3) = ? where NE (1) is the number of

NE(4) = ? Orbitals in the ith

NE(5) = ? Orbital

NE(6) = ?

If each conjugated atom contribute one electron, the total number of electrons would be 6 for benzene or pyridine. According to Pauli exclusion principle these would occupy the three lowest energy orbitals. The values to be keyed in would be:

NE(1)=2, NE(2)=2, NE(3)=2, NE(4)=0, NE(5)=0, NE(6)=0.

It must be remembered that these values are for neutral pyridine. For cations these would be:

NE(1)=2, NE(2)=2, NE(3)=1, NE(4)=0, NE(5)=0, NE(6)=0.

While for anion these would be :

NE(1)=2, NE(2)=2, NE(3)=2, NE(4)=0, NE(5)=0, NE(6)=0.

Loading of input data has now been completed. The computer would inquire whether a printout of this input data is required or not. If the answer is in affermative this data is printed otherwise the calculations are carried out and 'Busy' appears on the screen. After some time (few minutes or few hrs. depending upon the size of the molecule) results in the form of eigen values, eigen vectors, charge densities, bond orders would appear on the screen.

Once these results had been displayed the computer would ask if a printout of result is required or not. An answer in Y (Yes) would give the required printout.

II. *SCF Calculation*: The SCF calculations can also start with a test run if desired. The molecule used for testrun in this case is also pyridine.

The number of conjugated atoms N and the array NE(1) (No. of electrons in the ith orbital) is filled in the same way as for the ω-technique calculations (see previous section for detail).

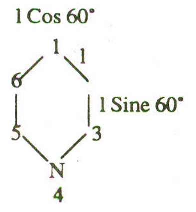
The input now reads the atomic co-ordinates. The atomic co-ordinate are given in the form of hexagonal grid co-ordinates. Grid co-ordinates are defined as multiples of 31/2 (i.e. 1 Sin 60°) and 1/21 =(lcos 60°) in the x and y directions respectively. l is the c-c bond length and is taken equal to 1.4.

Two arrays X(N) and Y(N) to be filled where N is the number of conjugated atoms. The display on the screen for a benzene or pyridene molecule would be :

X(1) =?	Y(1) =?
X(2) =?	Y(2) =?
X(3) =?	Y(3) =?
X(4) =?	Y(4) =?
X(5) =?	Y(5) =?
X(6) =?	Y(6) =?

The questions is that how do we evaluate the grid co-ordinates. Again consider the case of pyridine (Note that the numbering is arbitary).

X(1) =0	Y(1) =+2
X(2) =1	Y(2) =+1
X(3) =1	Y(3) =-1
X(4) =0	Y(4) =-2
X(5) =-1	Y(5) =-1
X(6) =-1	Y(6) =+1



i.e. l cos 60° (1/2 l) is considered a unit in the Y direction while l sin 60° (3/21) is considered to be a unit in the x direction.

After keying in the grid-co-ordinates the user would have to opt for charged sphere model or Nishimoto Mataga formulation of repulsion integral matrix. Either of these methods can be used. The repulsion integral matrix thus formed can also be inspected and modified if desired. This inspection and modification facility however, is optional.

Thus the user is free to choose any value of the repulsion integrals. In case of default, however, charged sphere model, with the specifically mentioned set of repulsion integrals is used. This is not end of the input data. It has not been specified that atom 4 is a nitrogen atom. The computer would ask whether a system is a substituted or an hetero aromatic one.

If the answer is yes then some more input data is to be keyed in.

The next question would be the number of substituted or hetero aromatic atoms. For the case of pyridine the answer to this question would be 1. Next it would inquire about the position of the substituted or hetero atom. For pyridine the answer would be 4 (according to the numbering in the above diagram).

The computer would now ask the values of Z(4) =?, DW(4) =?, C(4,4) =?.

Where the above values represent the charge, coulumb integral and the repulsion integral. These values can be taken from the literature.

For pyridene these values can be Z(4) =1, DW(4) = −1.659, G(4,4) =11.35.

Once these values are keyed in the input data is complete. A printout of input data can be taken if required.

The program would run for a few minutes or few hrs. and the results would appear on the screen. The results are in some form as for ω-technique calculation i.e. values of Eigen values eigen vectors, bond orders, charge densities are displayed on the screen. A printout of results can also be taken..

```
1 REM  MAIN PROGRAMME -HMO CALCULATIONS WITH w-TECHNIQUE
2 PRINT"HMO CALCULATIONS WITH w-TECHNIQUE
3 INPUT "TEST RUN REQUIRED(Y/N)";T$
4 IF T$="Y" THEN GOSUB 1200
7 IF T$="y" THEN GOSUB 1200
8 GOSUB 80
10 FOR I = 1 TO N
12  FOR J = I TO N
14   IF I = J THEN GOTO 18
16   H(J,I) = H(I,J)
18   H(I,I) = G(I) : HD(I) = H(I,I)
20  NEXT J
22 NEXT I
25 FOR KK = 1 TO 8
28  GOSUB 230
30  GOSUB 600
32  GOSUB 700
34  IF KK = 8 THEN 56
35  GOSUB 700
36   FOR I = 1 TO N
38    HD(I) = G(I)
40    HD(I) = HD(I) + ( 1 - P(I,I) ) * 1.4 * (-1)
42   NEXT I
44   FOR I = 1 TO N-1
45    FOR J = I+1 TO N
48     H(J,I) = H(I,J)
50    NEXT J
52   NEXT I
54 NEXT KK
56 GOSUB 800
57 ERASE HD,NE,U,UT,P,G
58 T=T-1
60 IF T=-1 THEN 65
62 PRINT "TEST RUN COMPLETE"
63 GOTO 8
65 SYSTEM
79 '
80 '    *******       ******      INPUT SUBROUTINE      *******
81 '
83 IF T=1 THEN 225
85  INPUT "NAME OF THE MOLECULE= ";N$
90 INPUT"NUMBER OF CONJUGATED ATOMS";N
95 DIM HD(N): DIM NE(N): DIM U(N,N): DIM UT(N): DIM P(N,N): DIM G(N)
100 FOR I = 1 TO N
105  FOR J = I TO N
110   PRINT "H(";I;",";J;") = ";
115   INPUT H(I,J)
120  NEXT J
125 NEXT I
140 FOR I = 1 TO N
145  PRINT "NE(";I;") = ";
150  INPUT NE(I)
155 NEXT I
159 PRINT
160 PRINT "Verification of H(i,j)": PRINT" To Confirm, press space-bar"
161 PRINT "To Alter, press the = key, then key-in the correct value, then press
 return"
165 FOR I = 1 TO N
170  FOR J = I TO N
175   PRINT "H(";I;",";J;") = ";H(I,J),
177   U$ = INKEY$: IF U$ = "" THEN 177
178   IF U$ = " " THEN PRINT " O.K. ": GOTO 185
179   IF U$ = "=" THEN PRINT " = ": INPUT H(I,J): GOTO 185
185   H(I,J) = -H(I,J) : H(J,I) = H(I,J) : HD(I) = H(I,I) : G(I) = HD(I)
200  NEXT J
205 NEXT I
210 T=0
215 PRINT "Printout of input data required ? (y/n) "
220 U$ = INKEY$ : IF U$ = "" THEN 220
221 PRINT U$ : IF U$ = "y" THEN GOSUB 3000
222 IF U$ = "Y" THEN GOSUB 3000
223 PRINT "BUSY"
225 RETURN
229 '
230 '     ********      ********     DIAGONALISATION SUBROUTINE    ******
231 ' This subroutine diagonalizes the input matrix
240 FOR J=1 TO N
245  FOR I=1 TO N
250   U(I,J)=0: U(I,I)=1
260  NEXT I
265 NEXT J
270 EP=1E-16
275 MX=0
280 FOR I = 2 TO N
285  FOR J=1 TO (I-1)
290   H(I,I)=HD(I): H(J,J)=HD(J): SQ=(H(I,J))^2
310   IF SQ > MX THEN LET MX = SQ
315   IF SQ <= EP THEN 510
320   D = HD(I) - HD(J)
325   IF D >= 0 THEN 340
330   SN = -2 : D = -D
335   GOTO 345
340   SN = 2
345   TD = D + SQR(( D*D )+( 4*SQ )) : TN = ( SN*H(I,J) )/TD
355   C = 1/(SQR(1+(TN*TN))) : S = C*TN
370   FOR K = 1 TO N
380    XJ = C*U(K,J)-S*U(K,I): U(K,I) = S*U(K,J)+C*U(K,I): U(K,J) = XJ
390    IF K = J THEN 485
400    IF K > J THEN 435
```

```
410    XJ = C*H(J,K) - S*H(I,K)
420    H(I,K) = S*H(J,K) + C*H(I,K)
425    H(J,K) = XJ
430    GOTO 485
435    IF K = I THEN 485
440    IF K > I THEN 470
450    XJ = C*H(K,J) - S*H(I,K) : H(I,K) = S*H(K,J) + C*H(I,K) : H(K,J) = XJ
465    GOTO 485
470    XJ = C*H(K,J) - S*H(K,I) : H(K,I) = S*H(K,J) + C*H(K,I) : H(K,J) = XJ
485   NEXT K
490   HD(I) = (C*C*H(I,I)) + (S*S*H(J,J)) + (2*S*C*H(I,J))
500   HD(J) = (C*C*H(J,J)) + (S*S*H(I,I)) - (2*S*C*H(I,J))
505   H(I,J) = 0
510  NEXT J
515 NEXT I
525 IF MX > EP THEN 275
530 RETURN
532 '
534 '
599 '
600 '    ******        *******      ORDER SUBROUTINE    *******
601 'This subroutine orders the eigenvalues and eigenvectors and adjusts the
 electronic configuration for degenerate spaces
605 FOR K = 1 TO N
615  HT = HD(K) : JT = K
620  FOR J = K TO N
625   IF HD(J) >= HT THEN 640
630   HT = HD(J) : JT = J
640  NEXT J
645  HD(JT) = HD(K) : HD(K) = HT
655  FOR I = 1 TO N
660   UT(I) = U(I,JT) : U(I,JT) = U(I,K) : U(I,K) = UT(I)
670  NEXT I
675 NEXT K
678 IF KK<>1 THEN GOTO 699
680 FOR I=1 TO N-1
690 IF ABS(HD(I)-HD(I+1))>.0001 THEN GOTO 698
692 X=(NE(I)+NE(I+1))/2
695 NE(I)=X:NE(I+1)=X
698 NEXT I
699 RETURN
700 '        *********       ********   SUBROUTINE BOND-ORDERS  *****
701 '
705 FOR R = 1 TO N
710  FOR S = R TO N
715   SM = 0
720   FOR J = 1 TO N
725    SM = SM + NE(J) * U(R,J) * U(S,J)
730   NEXT J
740   P(R,S) = SM
745  NEXT S
750 NEXT R
755 RETURN
799 '
800 '      *******         ******    OUTPUT SUBROUTINE     ******
801 '
803 PRINT "          ******RESULTS******"
805 BEEP : FOR CTR = 1 TO 10 : NEXT
810 INPUT "EIGENVALUES REQUIRED ? (Y/N) ";E$
812 IF E$ = "N" THEN 840
814 IF E$ = "n" THEN 840
815 FOR I = 1 TO N
825  PRINT "E(";I;") = ";HD(I)
830 NEXT I
840 INPUT "EIGENVECTORS REQUIRED ? (Y/N) ";U$
845 IF U$ = "N" THEN 875
846 IF U$ = "n" THEN 875
```

```
850 FOR J = 1 TO N
855  FOR I = 1 TO N
860   PRINT " U(";I;",";J;") = ";U(I,J)
865  NEXT I
870 NEXT J
875 INPUT "CHARGE DENSITIES REQUIRED ? (Y/N) ";C$
880 IF C$ = "N" THEN 900
881 IF C$ = "n" THEN 900
885 FOR I = 1 TO N
890  PRINT "Q(";I;") = ";P(I,I)
895 NEXT I
900 INPUT "BOND-ORDERS REQUIRED ? (Y/N) ";P$
905 IF P$ = "N" THEN 930
906 IF P$ = "n" THEN 930
910 FOR I = 1 TO (N-1)
915  FOR J = (I+1) TO N
920   PRINT "P(";I;",";J;") = ";P(I,J)
925  NEXT J
926 NEXT I
928 IF T=1 THEN 960
930 INPUT "PRINT-OUT OF RESULTS REQUIRED ? (Y/N) ";S$
950 IF S$ = "Y" THEN GOSUB 4000
951 IF S$ = "y" THEN GOSUB 4000
960 RETURN
963 '
965 '
1200 REM ******SUBROUTINE TESTRUN******
1205 REM This programme runs the sample programme for pyridene
1210 PRINT"TEST RUN FOR PYRIDENE-HMO WITH w-TECHNIQUE"
1215  DIM HD(6): DIM NE(6): DIM U(6,6): DIM UT(6): DIM P(6,6): DIM G(6)
1218 N=6
1220 PRINT"FOR PYRIDENE hx=0.5,Kc_x=0.8
1223 DATA 0.5,0.8,0,0,0,0.8,0,1,0,0,0,0,1,0,0,0,1,0,0,1,0
1225 FOR I=1 TO N
1230  FOR J=I TO N
1235   READ H(I,J)
1240    PRINT "H(";I;",";J;") = ";H(I,J),
1242   H(I,J) = -H(I,J) : H(J,I) = H(I,J) : HD(I) = H(I,I) : G(I) = HD(I)
1245  NEXT J
1250 NEXT I
1255 DATA 2,2,2,0,0,0
1260 FOR I=1 TO N
1265  READ NE(I)
1270  PRINT "NE(";I;") = ";NE(I)
1275 NEXT I
1280 T=1
1288 RETURN
1290 '
1300 '
3000 REM ******Subroutine printout input data******
3010 REM This subroutine prints the input data (optional)
3020 LPRINT"HMO with w-TECHNIQUE"
3025 '
3030 LPRINT M$
3035 LPRINT
3040 LPRINT "INPUT DATA"
3050 FOR I=1 TO N
3055  FOR J=I TO N
3065   LPRINT "H(";I;",";J;") = ";-H(I,J),
3070  NEXT J
3075 NEXT I
3080 LPRINT
3090 FOR I=1 TO N
3095  LPRINT "NE(";I;") = ";NE(I)
3100 NEXT I
3110 RETURN
3120 '
```

```
3130 '
4000 REM *******Results printout subroutine******
4010 REM This subroutine prints the results(optional)
4020 LPRINT  "******RESULTS******"
4030 LPRINT
4040 LPRINT N$
4050 LPRINT
4060 LPRINT "ITERATION NO.";KK
4070 LPRINT
4080 LPRINT "   *****Eigenvalues*****
4085 FOR I=1 TO N
4090  LPRINT "E(";I;")=";HD(I)
4100 NEXT I
4110 LPRINT
4120 LPRINT "   *****Charge densities*****"
4130 FOR I=1 TO N
4140  LPRINT"Q(";I;")=";P(I,I)
4150 NEXT I
4160 LPRINT
4170 LPRINT"    ******EIGENVECTORS***** "
4180 FOR I=1 TO N
4190  FOR J=I TO N
4210   LPRINT " U(";I;",";J;") = ";U(I,J)
4220  NEXT J
4230 NEXT I
4270 FOR I = 1 TO (N-1)
4280  FOR J = (I+1) TO N
4290   LPRINT "P(";I;",";J;") = ";P(I,J)
4300  NEXT J
4310 NEXT I
4320 RETURN


51 REM  MAIN PROGRAMME-HMO CALCULATIONS
2 PRINT"HMO CALCULATIONS "
3 INPUT "TEST RUN REQUIRED(Y/N)";T$
4 IF T$="Y" THEN GOSUB 1200
7 IF T$="y" THEN GOSUB 1200
8 GOSUB 80
28  GOSUB 230
30  GOSUB 600
35  GOSUB 700
56 GOSUB 800
57 ERASE HD,NE,U,UT,P,G
58 T=T-1
60 IF T=-1 THEN 65
62 PRINT "TEST RUN COMPLETE"
63 GOTO 8
65 SYSTEM
79 '
80 '     *******       ******     INPUT SUBROUTINE     *******
81 '
83 IF T=1 THEN 225
85  INPUT "NAME OF THE MOLECULE= ";N$
90 INPUT"NUMBER OF CONJUGATED ATOMS";N
95 DIM HD(N): DIM NE(N): DIM U(N,N): DIM UT(N): DIM P(N,N): DIM G(N)
100 FOR I = 1 TO N
105  FOR J = I TO N
110   PRINT "H(";I;",";J;") = ";
115   INPUT H(I,J)
120  NEXT J
125 NEXT I
140 FOR I = 1 TO N
145  PRINT "NE(";I;") = ";
150  INPUT NE(I)
155 NEXT I
159 PRINT
160 PRINT "Verification of H(i,j)": PRINT" To Confirm, press space-bar"
161 PRINT "To Alter, press the = key, then key-in the correct value, then press
 return"
165 FOR I = 1 TO N
170  FOR J = I TO N
175   PRINT "H(";I;",";J;") = ";H(I,J),
177   U$ = INKEY$: IF U$ = "" THEN 177
178   IF U$ = " " THEN PRINT " O.K. ": GOTO 185
179   IF U$ = "=" THEN PRINT " = ": INPUT H(I,J): GOTO 185
185   H(I,J) = -H(I,J) : H(J,I) = H(I,J) : HD(I) = H(I,I) : G(I) = HD(I)
200  NEXT J
205 NEXT I
210 T=0
215 PRINT "Printout of input data required ? (y/n) "
220 U$ = INKEY$ : IF U$ = "" THEN 220
221 PRINT U$ : IF U$ = "y" THEN GOSUB 3000
222 IF U$ = "Y" THEN GOSUB 3000
223 PRINT "BUSY"
225 RETURN
229 '
230 '        ********     ********   DIAGONALISATION SUBROUTINE   ******
231 ' This subroutine diagonalizes the input matrix
240 FOR J=1 TO N
245  FOR I=1 TO N
250   U(I,J)=0: U(I,I)=1
260  NEXT I
265 NEXT J
270 EP=1E-16
275 MX=0
280 FOR I = 2 TO N
285  FOR J=1 TO (I-1)
290   H(I,I)=HD(I): H(J,J)=HD(J): SQ=(H(I,J))^2
310   IF SQ > MX THEN LET MX = SQ
315   IF SQ <= EP THEN 510
320   D = HD(I) - HD(J)
325   IF D >= 0 THEN 340
330   SN = -2 : D = -D
335   GOTO 345
340   SN = 2
345   TD = D + SQR(( D*D )+( 4*SQ )) : TN = ( SN*H(I,J) )/TD
355   C = 1/(SQR(1+(TN*TN))) : S = C*TN
370   FOR K = 1 TO N
380    XJ = C*U(K,J)-S*U(K,I): U(K,I) = S*U(K,J)+C*U(K,I): U(K,J) = XJ
390    IF K = J THEN 485
400    IF K > J THEN 435
410    XJ = C*H(J,K) - S*H(I,K)
420    H(I,K) = S*H(J,K) + C*H(I,K)
425    H(J,K) = XJ
430    GOTO 485
435    IF K = I THEN 485
440    IF K > I THEN 470
450    XJ = C*H(K,J) - S*H(I,K) : H(I,K) = S*H(K,J) + C*H(I,K) : H(K,J) = XJ
465    GOTO 485
470    XJ = C*H(K,J) - S*H(K,I) : H(K,I) = S*H(K,J) + C*H(K,I) : H(K,J) = XJ
485   NEXT K
490   HD(I) = (C*C*H(I,I)) + (S*S*H(J,J)) + (2*S*C*H(I,J))
500   HD(J) = (C*C*H(J,J)) + (S*S*H(I,I)) - (2*S*C*H(I,J))
505   H(I,J) = 0
510  NEXT J
515 NEXT I
525 IF MX > EP THEN 275
530 RETURN
532 '
534 '
599 '
600 '   ******        *******      ORDER SUBROUTINE    *******
601 'This subroutine orders the eigenvalues and eigenvectors and also
```

```
602 ' readjusts the electronic configuration for degenerate levels
605 FOR K = 1 TO N
615  HT = HD(K) : JT = K
620  FOR J = K TO N
625   IF HD(J) >= HT THEN 640
630   HT = HD(J) : JT = J
640  NEXT J
645  HD(JT) = HD(K) : HD(K) = HT
655  FOR I = 1 TO N
660   UT(I) = U(I,JT) : U(I,JT) = U(I,K) : U(I,K) = UT(I)
670  NEXT I
675 NEXT K
680 FOR I=1 TO N-1
690 IF ABS(HD(I)-HD(I+1))>.0001 THEN GOTO 698
692 X=(NE(I)+NE(I+1))/2
695 NE(I)=X:NE(I+1)=X
698 NEXT I
699 RETURN
700 '        *********      ********   SUBROUTINE BOND-ORDERS  *****
701 '
705 FOR R = 1 TO N
710  FOR S = R TO N
715   SM = 0
720   FOR J = 1 TO N
725    SM = SM + NE(J) * U(R,J) * U(S,J)
730   NEXT J
740   P(R,S) = SM
745  NEXT S
750 NEXT R
755 RETURN
799 '
800 '       *******         ******    OUTPUT SUBROUTINE     ******
801 '
803 PRINT "           ******RESULTS******"
805 BEEP : FOR CTR = 1 TO 10 : NEXT
810 INPUT "EIGENVALUES REQUIRED ? (Y/N) ";E$
812 IF E$ = "N" THEN 840
814 IF E$ = "n" THEN 840
815 FOR I = 1 TO N
825  PRINT "E(";I;") = ";HD(I)
830 NEXT I
840 INPUT "EIGENVECTORS REQUIRED ? (Y/N) ";U$
845 IF U$ = "N" THEN 875
846 IF U$ = "n" THEN 875
850 FOR J = 1 TO N
855 FOR I = 1 TO N
860   PRINT " U(";I;",";J;") = ";U(I,J)
865  NEXT I
870 NEXT J
875 INPUT "CHARGE DENSITIES REQUIRED ? (Y/N) ";C$
880 IF C$ = "N" THEN 900
881 IF C$ = "n" THEN 900
885 FOR I = 1 TO N
890  PRINT "Q(";I;") = ";P(I,I)
895 NEXT I
900 INPUT "BOND-ORDERS REQUIRED ? (Y/N) ";P$
905 IF P$ = "N" THEN 930
906 IF P$ = "n" THEN 930
910 FOR I = 1 TO (N-1)
915  FOR J = (I+1) TO N
920   PRINT "P(";I;",";J;") = ";P(I,J)
925  NEXT J
926 NEXT I
928 IF T=1 THEN 960
930 INPUT"printout of results required";S$
950 IF S$ = "Y" THEN GOSUB 4000
951 IF S$ = "y" THEN GOSUB 4000
960 RETURN
963 '
965 '
1200 REM ******SUBROUTINE TESTRUN******
1205 REM This programme runs the sample programme for pyridene
1210 PRINT"TEST RUN FOR PYRIDENE FOR HMO CALCULATIONS"
1215  DIM HD(6): DIM NE(6): DIM U(6,6): DIM UT(6): DIM P(6,6): DIM G(6)
1218 N=6
1220 PRINT"FOR PYRIDENE hx=0.5,Kc_x=0.8
1223 DATA 0.5,0.8,0,0,0,0.8,0,1,0,0,0,0,1,0,0,0,1,0,0,1,0
1225 FOR I=1 TO N
1230  FOR J=I TO N
1235   READ H(I,J)
1240    PRINT "H(";I;",";J;") = ";H(I,J),
1242   H(I,J) = -H(I,J) : H(J,I) = H(I,J) : HD(I) = H(I,I) : G(I) = HD(I)
1245  NEXT J
1250 NEXT I
1255 DATA 2,2,2,0,0,0
1260 FOR I=1 TO N
1265  READ NE(I)
1270  PRINT "NE(";I;") = ";NE(I)
1275 NEXT I
1280 T=1
1288 RETURN
1290 '
1300 '
3000 REM ******Subroutine printout input data******
3010 REM This subroutine prints the input data (optional)
3020 LPRINT"HMO CALCULATIONS"
3025 '
3030 LPRINT N$
3035 LPRINT
3040 LPRINT "INPUT DATA"
3050 FOR I=1 TO N
3055  FOR J=I TO N
3065   PRINT "H(";I;",";J;") = ";-H(I,J),
3070  NEXT J
3075 NEXT I
3080 LPRINT
3090 FOR I=1 TO N
3095  LPRINT "NE(";I;") = ";NE(I)
3100 NEXT I
3110 RETURN
3120 '
3130 '
4000 REM *******Results printout subroutine******
4010 REM This subroutine prints the results(optional)
4020 LPRINT   "*****RESULTS*******"
4030 LPRINT
4040 LPRINT N$
4050 LPRINT
4070 LPRINT
4080 LPRINT "    *****Eigenvalues*****
4085 FOR I=1 TO N
4090  LPRINT "E(";I;")=";HD(I)
4100 NEXT I
4110 LPRINT
4120 LPRINT "   *****Charge densities*****"
4130 FOR I=1 TO N
4140  LPRINT"Q(";I;")=";P(I,I)
4150 NEXT I
4160 LPRINT
4170 LPRINT"    ******EIGENVECTORS***** "
4180 FOR I=1 TO N
4190  FOR J=I TO N
4210   LPRINT " U(";I;",";J;") = ";U(I,J)
4220  NEXT J
```

```
4230 NEXT I
4270 FOR I = 1 TO (N-1)
4280  FOR J = (I+1) TO N
4290   LPRINT "P(";I;",";J;") = ";P(I,J)
4300  NEXT J
4310 NEXT I
4320 RETURN
```

```
5 CLS : REM SCF-PPP CALCULATIONS
10 PRINT "SCF CALCULATIONS WITH PARISER PARR AND POPLE'S(PPP) APPROXIMATION"
12 INPUT"TEST RUN REQUIRED(Y/N)";T$
13 IF T$="Y" THEN GOSUB 1200
14 IF T$="y" THEN GOSUB 1200
15 REM
20 GOSUB 2065
25 GOSUB 2300
30 GOSUB 2190
35 FOR KS=1 TO 10
40  GOSUB 230
45  GOSUB 600
50  GOSUB 700
55  IF KS=10 THEN GOTO 70
60  GOSUB 2500
65 NEXT KS
70 GOSUB 800
71 ERASE  X,Y,NE,U,UT,P,G,Z,DW,B,H,HD
72 T=T-1
74 IF T=-1 THEN 78
75 PRINT"TEST RUN COMPLETE"
76 GOTO 20
78 SYSTEM
230 '        *******      *******     DIAGONALISATION SUBROUTINE    ******
231 '
240 FOR J=1 TO N
245  FOR I=1 TO N
250   U(I,J)=0: U(I,I)=1
260  NEXT I
265 NEXT J
270 EP=1E-16
275 MX=0
280 FOR I = 2 TO N
285  FOR J=1 TO (I-1)
290   H(I,I)=HD(I): H(J,J)=HD(J): SQ=(H(I,J))^2
300
310   IF SQ > MX THEN LET MX = SQ
315   IF SQ <= EP THEN 510
320   D = HD(I) - HD(J)
325   IF D >= 0 THEN 340
330   SN = -2 : D = -D
335   GOTO 345
340   SN = 2
345   TD = D + SQR(( D*D )+( 4*SQ )) : TN = ( SN*H(I,J) )/TD
355   C = 1/(SQR(1+(TN*TN))) : S = C*TN
370   FOR K = 1 TO N
380    XJ = C*U(K,J)-S*U(K,I): U(K,I) = S*U(K,J)+C*U(K,I): U(K,J) = XJ
390    IF K = J THEN 485
400    IF K > J THEN 435
410    XJ = C*H(J,K) - S*H(I,K)
420    H(I,K) = S*H(J,K) + C*H(I,K)
425    H(J,K) = XJ
430    GOTO 485
435    IF K = I THEN 485
440    IF K > I THEN 470
450    XJ = C*H(K,J) - S*H(I,K) : H(I,K) = S*H(K,J) + C*H(I,K) : H(K,J) = XJ
465    GOTO 485
470    XJ = C*H(K,J) - S*H(K,I) : H(K,I) = S*H(K,J) + C*H(K,I) : H(K,J) = XJ
485   NEXT K
490   HD(I) = (C*C*H(I,I)) + (S*S*H(J,J)) + (2*S*C*H(I,J))
500   HD(J) = (C*C*H(J,J)) + (S*S*H(I,I)) - (2*S*C*H(I,J))
505   H(I,J) = 0
510  NEXT J
515 NEXT I
525 IF MX > EP THEN 275
530 RETURN
599 '
600 '    ******         *******      ORDER SUBROUTINE    *******
601 'This subroutine orders the eigenvalues and eigenvectors and also
602 'readjusts the electronic configuration for degenerate spaces
605 FOR K = 1 TO N
615  HT = HD(K) : JT = K
620  FOR J = K TO N
625   IF HD(J) >= HT THEN 640
630   HT = HD(J) : JT = J
640  NEXT J
645  HD(JT) = HD(K) : HD(K) = HT
655  FOR I = 1 TO N
660    UT(I) = U(I,JT) : U(I,JT) = U(I,K) : U(I,K) = UT(I)
670  NEXT I
675 NEXT K
678 IF KS<>1 THEN GOTO 699
680 FOR I=1 TO (N-1)
690 IF ABS(HD(I)-HD(I+1))>.0001 THEN GOTO 698
692 X=(NE(I)+NE(I+1))/2
695 NE(I)=X:NE(I+1)=X
698 NEXT I
699 RETURN
700 '         *********       ********   SUBROUTINE BOND-ORDERS  *****
701 '
705 FOR R = 1 TO N
710  FOR S = R TO N
715   SM = 0
720   FOR J = 1 TO N
725    SM = SM + NE(J) * U(R,J) * U(S,J)
730   NEXT J
740   P(R,S) = SM
745  NEXT S
750 NEXT R
755 RETURN
799 '
800 '      *******          ******    OUTPUT SUBROUTINE     ******
801 '
805 BEEP : FOR CTR = 1 TO 10 : NEXT
810 PRINT "EIGENVALUES REQUIRED?(Y/N)";E$
812 IF E$ = "N" THEN 840
814 IF E$ = "n" THEN 840
815 FOR I = 1 TO N
825  PRINT "E(";I;") = ";HD(I)
830 NEXT I
832 REM Eigenvalues are calculated considering W+Gc/2 as zero.W=-11 and Gc=11.35
 are the default values.For absolute eigenvalues substract the term 5.32 from
the calculated value.
840 INPUT "EIGENVECTORS REQUIRED ? (Y/N) ";U$
845 IF U$ = "N" THEN 875
846 IF U$ = "n" THEN 875
850 FOR J = 1 TO N
855  FOR I = 1 TO N
860   PRINT " U(";I;",";J;") = ";U(I,J)
865  NEXT I
870 NEXT J
875 INPUT "CHARGE DENSITIES REQUIRED ? (Y/N) ";C$
880 IF C$ = "N" THEN 900
881 IF C$ = "n" THEN 900
```

```
885 FOR I = 1 TO N
890  PRINT "Q(";I;") = ";P(I,I)
895 NEXT I
900 INPUT "BOND-ORDERS REQUIRED ? (Y/N) ";P$
905 IF P$ = "N" THEN 930
906 IF P$ = "n" THEN 930
910 FOR I = 1 TO (N-1)
915  FOR J = (I+1) TO N
920   PRINT "P(";I;",";J;") = ";P(I,J)
925  NEXT J
928 NEXT I
930 INPUT "PRINT OUT OF OUTPUT DATA REQUIRED ? (Y/N) ";S$
950 IF S$ = "Y" THEN GOSUB 4000
951 IF S$ = "y" THEN GOSUB 4000
960 RETURN
1200 '******SUBROUTINE TESTRUN SCF******
1202 PRINT "TEST RUN FOR PYRIDENE"
1205 T=1
1210 N=6
1215 DIM X(N):DIM Y(N):DIM NE(N):DIM U(N,N):DIM UT(N):DIM P(N,N):DIM G(N,N)
1220 DIM Z(N):DIM DW(N)
1225 FOR I=1 TO N
1230  Z(I)=1:DW(I)=0
1235 NEXT I
1240 DATA 2,2,2,0,0,0
1255 FOR I = 1 TO N
1260  READ NE(I)
1265  PRINT "NE(";I;") = "; NE(I)
1270 NEXT I
1272 DATA 0,2,1,1,1,-1,0,-2,-1,-1,-1,1
1275 FOR I=1 TO N
1280  READ X(I) : READ Y(I)
1285  PRINT "X-coordinate of atom (";I;")=";X(I);
1287  PRINT TAB(40);"Y-coordinate of atom (";I;")=";Y(I)
1290 NEXT I
1311 PRINT
1312 PRINT "The repulsion integrals used are:G(I,J)=14.4/D:For 2.81<D<2.75 G(I,J
)=4.97:      For 2.75<D<1.42 G(I,J)=5.77:For D<1.42 G(I,J)=7.19:G(I,I)=11.35"
1313 PRINT"    where D is the interatomic distance (charged sphere model)":PRINT
1314 PRINT "Also DW(I)=0 and Z(I)=1 for conjugated carbon atoms"
1315 PRINT "Pyridene is a heteroaromatic system.";
1330 PRINT " The heteroatom(N Atom) is at position 1"
1335 G(1,1)=11.35
1340 DW(1)=-1.659
1342 PRINT" G(1,1)=11.35"
1345 PRINT" DW(1)=-1.659                                         *** BUSY ***"
1348 PRINT" Z(1)=1                                              (please wait...)"
1349 RI=1
1350 RETURN
2065 REM       ******SCF INPUT SUBROUTINE******
2068 IF T=1 THEN 2185
2070 REM This subroutine inputs NE(I){number of electrons in the i-th MO},X(I)
 and Y(I){grid coordinates of the conjugated atom I}
2075 INPUT "NAME OF THE MOLECULE= ";N$
2078 INPUT"NUMBER OF CONJUGATED ATOMS";N
2080 DIM X(N):DIM Y(N):DIM NE(N):DIM U(N,N):DIM UT(N):DIM P(N,N):DIM G(N,N)
2082 DIM Z(N):DIM DW(N)
2085 FOR I=1 TO N
2090  Z(I)=1:DW(I)=0
2095 NEXT I
2104 PRINT"NE(I)=NUMBER OF ELECTRONS IN THE I-th ORBITAL"
2105 FOR I = 1 TO N
2110  PRINT "NE(";I;") = ":
2115  INPUT NE(I)
2120 NEXT I
2125 FOR I=1 TO N
2130  PRINT "X-coordinate of atom (";I;")=";
2135  INPUT X(I)
2140 NEXT I
2145 FOR I=1 TO N
2150  PRINT "Y-coordinate of atom (";I;")=";
2155  INPUT Y(I)
2160 NEXT I
2170 PRINT "Printout of input data required ? (y/n) "
2172 PRINT
2175 I$ = INKEY$ : IF I$ = "" THEN 2175
2180 IF I$ = "Y" THEN GOSUB 3000
2182 IF I$ = "y" THEN GOSUB 3000
2185 RETURN
2190 REM       ******Modification subroutine******
2195 REM            THIS SUBROUTINE MODIFIES THE SCF MATRIX FOR HETERAROMATIC OR
 SUBSTITUTED    SYSTEM
2198 IF T=1 THEN 2299
2200 PRINT "Heteoaromatic or substituted system (y/n)";
2210 INPUT A$
2215 IF A$="N" THEN 2299
2220 IF A$="n" THEN 2299
2224 INPUT "Number of substituted or heteroaromatic atoms";NH
2225 DIM PS(NH)
2235 FOR I= 1 TO NH
2245  PRINT "Position of heteroatom";I;
2250  INPUT PS(I)
2255  K=PS(I)
2260  PRINT "DW(";K;")=";
2265  INPUT DW(K)
2268  H(K,K)=DW(K):HD(K)=DW(K)
2270  PRINT "Z(";K;")=";
2275  INPUT Z(K)
2280  PRINT "G(";K;",";K;")=";
2285  INPUT G(K,K)
2290 NEXT I
2291 INPUT"Printout of the modified input data reqd.";I$
2292 IF I$="Y" THEN GOSUB 3100
2294 IF I$="y" THEN GOSUB 3100
2299 RETURN
2300 REM       ******Subroutine Gamma******
2301 REM This subroutine computes the repulsion matrix G(N,N) after computing
interatomic distances from grid co-ordinates.
2302 IF T=1 THEN 2308
2303 PRINT "The repulsion integrals which may be used are:
           1. For D>2.81 G(I,J)=14.4/D:For 2.81<D<2.75 G(I,J)=4.97:
              For 2.75<D<1.42 G(I,J)=5.77:For D<1.42 G(I,J)=7.19:G(I,I)=11.35"
2304 PRINT"    where D is the interatomic distance (charged sphere model)":PRINT
2305 PRINT "2. G(I,J)=14.397/(1.29+D) (Nishomoto-Mataga formulation)":PRINT
2306 INPUT "Enter choice number";RI:PRINT
2308 DIM B(N,N):DIM H(N,N):DIM HD(N)
2310 FOR I=1 TO N
2315  X(I)=X(I)*1.4*.866025:Y(I)=Y(I)*1.4*.5
2325 NEXT I
2326 IF RI=2 THEN 2421
2330 FOR I=1 TO N
2335  FOR J=1 TO N
2340   IF I<>J THEN 2355
2345   G(I,J)=11.35:GOTO 2414
2355   D=SQR(((X(J)-X(I))^2)+((Y(J)-Y(I))^2))
2360   IF D<=2.81 THEN 2375
2365   G(I,J)=14.4/D:GOTO 2410
2375   IF D<=2.75 THEN GOTO 2390
2380   G(I,J)=4.97:GOTO 2410
2390   IF D<=1.42 THEN 2405
2395.  G(I,J)=5.77:GOTO 2410
2405   G(I,J)=7.19
```

```
2410   G(J,I)=G(I,J)
2414   GC=11.35
2415  NEXT J
2418 NEXT I
2420 IF RI=1 THEN 2429
2421 FOR I=1 TO N
2422  FOR J=I TO N
2423   D=SQR(((X(J)-X(I))^2)+((Y(J)-Y(I))^2))
2424   G(I,J)=14.937/(1.293+D):G(J,I)=G(I,J):GC=14.937/1.293
2425  NEXT J
2426 NEXT I
2429 FOR I=1 TO N
2430  FOR J=1 TO N
2435   IF I=J THEN 2460
2440   IF RI=1 THEN IF G(I,J)<7 THEN 2470
2442   IF RI=2 THEN IF G(I,J)<5.2 THEN 2470
2445   B(I,J)=-2.37:H(I,J)=B(I,J):H(J,I)=H(I,J)
2460   H(I,I)=DW(I):HD(I)=H(I,I)
2470  NEXT J
2472 NEXT I
2473 IF T=1 THEN 2499
2474 INPUT "Inspection or modification of repulsion integrals required(Y/N)";MR$
2475 IF MR$="n" THEN  2499
2476 IF MR$="N" THEN  2499
2477 PRINT "Inspection of G(I,J)":PRINT "To confirm press space bar"
2478 PRINT "To alter ,press the = key, then key in the modified value and press
 enter"
2479 FOR I=1 TO N
2480  FOR J=I TO N
2481   PRINT "G(";I;",";J;")=";G(I,J)
2482   M$=INKEY$:IF M$="" THEN 2482
2483   IF M$=" " THEN 2486
2484   IF M$="=" THEN PRINT "=";:INPUT G(I,J)
2485   G(J,I)=G(I,J)
2486  NEXT J
2487 NEXT I
2490    PRINT"The repulsion integral G(I,I)=11.35,(or 11.55 for Nishimoto Mataga
formula) where I is a carbon atom (default mode).   If G(I,I) for carbon was cha
nged, then key in the new value."
2491    PRINT "Press enter if the displayed value is ok, else key in new value t
hen press enter"
2492   GC=11.35 : PRINT "G(I,I) for carbon=";GC ; : INPUT NGC
2493   IF NGC <> 0 THEN GC=NGC
2499 RETURN
2500 REM      ***Subroutine Gamma******
2502 REM This subroutine forms the SCF matrix iteratively
2505 FOR I=1 TO N
2510  SM=0
2515  FOR S=1 TO N
2520   IF S=I THEN 2530
2525   SM=SM+(P(S,S)-Z(S))*G(I,S)
2530  NEXT S
2535  H(I,I)=DW(I)+((1/2)*P(I,I)*G(I,I))-GC/2+SM
2540 NEXT I
2545 FOR I=1 TO N
2550  FOR J=I TO N
2555   IF I=J THEN 2570
2560   H(I,J)=B(I,J)-((1/2)*P(I,J)*G(I,J))
2565   H(J,I)=H(I,J)
2570   HD(I)=H(I,I)
2575  NEXT J
2580 NEXT I
2590 RETURN
3000 REM ******Subroutine printout input data******
3010 REM This subroutine prints the input data for SCF calculations(optional)
3015 LPRINT "   ******INPUT DATA*******"
3020 LPRINT N$
3025 FOR I=1 TO N
3030  LPRINT "X(";I;")=";X(I);"Y(";I;")=";Y(I)
3035 NEXT I
3040 FOR I = 1 TO N
3045  LPRINT "NE(";I;") = ";NE(I)
3050 NEXT I
3052 LPRINT"   **************"
3055 RETURN
3100 REM ******Subroutine printout modified input data******
3120 REM This subroutine prints the modified elements of the input data for
heteroaromatics(optional)
3130 LPRINT"Modified Elements"
3140 FOR I=1 TO NH
3150  K=PS(I)
3155  LPRINT "Z(";K;")=";Z(K)
3160  LPRINT "DW(";K;")=";DW(K)
3180  LPRINT "G(";K;",";K;")=";G(K,K)
3190 NEXT I
3200 RETURN
4000 REM ******Subroutine printout results******
4010 REM This subroutine prints the results(optional)
4012 LPRINT
4015 LPRINT  "  ******RESULTS*******"
4020 LPRINT "Iteration no.";KS
4040 LPRINT"    ******Eigenvalues******  "
4050 FOR I = 1 TO N
4060  LPRINT "E(";I;") = ";HD(I)
4070 NEXT I
4090 LPRINT"    ******Eigenvectors******"
5000 FOR J = 1 TO N
5010  FOR I = 1 TO N
5020   LPRINT " U(";I;",";J;") = ";U(I,J)
5030  NEXT I
5035 NEXT J
5050 LPRINT "  ******CHARGE DENSITIES ******"
5060 FOR I = 1 TO N
5070  LPRINT "Q(";I;") = ";P(I,I)
5080 NEXT
5090 LPRINT "   ******BOND-ORDERS******"
6000 FOR I = 1 TO (N-1)
6010  FOR J = (I+1) TO N
6020   LPRINT "P(";I;",";J;") = ";P(I,J)
6030  NEXT J
6040 NEXT I
6045 LPRINT"    *************"
6050 RETURN
```