REFERENCE

1. Bergmann, E and B. Pullman, Molecular and Quantum Pharmacology, D. Reidel Publ. Co., Dordrecht - Boston 1974.
2. Christoffersen, R.E., Quantum Mechanics of Molecular Conformations J. Wiley & Sons, London, 1976.
3. Peter, W.,"New Answer Through Chemotherapy," Experientia 40, 1351-1357, 1984.
4. Burger, A.,"Antimalarials," Medicinal Chemistry, 3rd ed., John Wiley & Sons, Inc., U.S.A., 492-521, 1970.
5. Peter , W.,"Antimalarial Drug Resistance:an Increasing Problem," Br. Med. Bull. 38, 187-192, 1982.
6. Maurice, J. and A.M.Pearce, Malaria, WHO , W.H.O. Pub., Geneva, Switzerland, 18-48, 1987.
7. Kim, K.H., C hansch, J.Y. Fukunaga, E.E. Steller and P.Y.C. Jow,"Quantitative Structure-Activity Relationships in 1-Aryl-2-(alkylamino) ethanol Antimalarials," J. Med. Chem., 22(4), 366-391, 1979.
8. Chien, Ping-Lu, D.J. McCausland, W.H. Burlori and C.C. Cheng,"Structure-Activity Relationship Studies on Antimalarial Phenan Threne Amino Alcohols Modification of the Side Chain," J. Med. Chem., 15(1), 28-33, 1972.
9. Kier, L.B., "Molecular Orbital Calculation of Prefered Conformations of Acetylchlorine, Muscarine and Muscarone," Mol. Pharmacol, 3, 487-494, 1967.
10. Kier, L.B.,"A Molecular Orbital Calculation of the referrred Conformation of Nicotin," Mol Pharmocol., 4, 70-76, 1968.

11. Ariens, E.J. and A.M. Simounes,"A Molecular Basis for Drug Action," J.Pharm.Pharmacol, 16, 137, 1964.

12. Bender, M.L., R.L. Van Etten,G.A. Clowes and J.F. Sebastian, "A Pictorial Description of the Lock and Key Theory," J.Am.Chem.Soc., 88,2318, 1966.

13. Silver, M.S.,"Hydrolysis of Conformationally Homogeneous Substrates by α - chymotrypsin Consequences of the Enzymatic Specificity," J.Am.Chem.Soc., 88, 4247, 1966.

14. Nemethy, G. and H.A. Scheraga,"The Structure of Water and Hysrophobic Bonding in Protein III. The Thermodynamic properties of Hydrophobic Bonds in Proteins," J.Phys.Chem., 66, 1773, 1962.

15. Wishnia, A. and T.W. Pinder,"Hydrophobic Interactions in Protiens. The Alkane Binding Site of β-Loctobuteines A and B," Biochemistry, 5, 1534, 1966.

16. Ralajezak, H. and W.J. Grville-Thomas (eds.), Molecular Interactions, Wiley, Chichesler, U.K., vol.1, 1980

17. Ginsburg, H. and T.G. Geary,"Current Concepts and New Ideas on the Mechanism of Action of Quinoline-Containing Antimolarials," Biochemical Pharmacological 36(10), 1567-1576, 1987.

18. Howell, R.E.,"Advanced in Chemotherapy," Med. Bull., 38(2), 193-199.

19. Pullman, B. and A. Pullman, Quantum Biochemistry, Interscience Publishers, New York, 1963.

20. Johnson, C.R. and R.E. Strepherd, "Aduacant Methyl to Remove Methyl Isomerization of (4-methylinidazole pentaamine cobalt (III)," Inorg.Chem., 22, 3506, 1983.

21. Strube, R.E., "The Search for New Antimalarial Drugs," J.Trop.Med.Hyg., 78, 1975.

22. Purcell, E.P. and J.M. Claylon, "Application of Regression Analysis to Antitumor Activity of Various Acetylanic Carbamates," J.Med.Chem, 11, 199, 1968.

23. Garnett, E.R., O.K. Wright, G.H. Miller, and K.L.Smith, J.Med.Chem., 9, 203, 1966.

24. Snedecor, G.W., Statistical Methods, ed.5, Iowa State Collage Press, Ames, 122-159, 413-446, 1956.

25. Silipo, C. and C. Hansch, "Correlation Analysis. Its Application to the Structure-Activity Relationships of Triazines Inhibiting Dihydrofolate Reductase," J. Am. Chem. Soc., 97(23), 6849-6891, 1975.

26. Brown, H.C. and Y.Okamoto, "Electrophillic Substituent Constannts," J.Am.Chem.Soc., 80, 4979, 1958.

27. Belleau, B., "The Molecular Theory of Drug Action Based on Induced Conformational Pertubations of Receptors," J.Med.Chem., 7, 776-784, 1964.

28. Kowalski, B.R. and C.F.Bender,"The Application of Pattern Recognition of Screening Prospective Anticancer Drugs. Adrenocarcinoma 755 Bilogical Activity test," J.Am.Chem.Soc., 96,918, 1974.

29. Schiffman, S.S.,"Physiochemical Correlates of Olfactory Quality," Science, 18, 112, 1974.

21. Strube, R.E., "The Search for New Antimalarial Drugs," J.Trop.Med.Hyg., 78, 1975.

22. Purcell, E.P. and J.M. Claylon, "Application of Regression Analysis to Antitumor Activity of Various Acetylanic Carbamates," J.Med.Chem, 11, 199, 1968.

23. Garnett, E.R., O.K. Wright, G.H. Miller, and K.L.Smith, J.Med.Chem., 9, 203, 1966.

24. Snedecor, G.W., Statistical Methods, ed.5, Iowa State Collage Press, Ames, 122-159, 413-446, 1956.

25. Silipo, C. and C. Hansch, "Correlation Analysis. Its Application to the Structure-Activity Relationships of Triazines Inhibiting Dihydrofolate Reductase," J. Am. Chem. Soc., 97(23), 6849-6891, 1975.

26. Brown, H.C. and Y.Okamoto, "Electrophillic Substituent Constannts," J.Am.Chem.Soc., 80, 4979, 1958.

27. Belleau, B., "The Molecular Theory of Drug Action Based on Induced Conformational Pertubations of Receptors," J.Med.Chem., 7, 776-784, 1964.

28. Kowalski, B.R. and C.F.Bender,"The Application of Pattern Recognition of screening Prospective Anticancer Drugs. Adrenocarcinoma 755 Bilogical Activity test," J.Am.Chem.Soc., 96,918, 1974.

29. Schiffman, S.S.,"Physiochemical Correlates of Olfactory Quality," Science, 18, 112, 1974.

30. Martin, Y.C., J.B. Holland, C.H.Jargoe, and N. Plotnikoff, "Discriminant Analysis of the Relationship Between Physical Properties and the Inhibition of Monoamine Oxidase by Aminotetralines and Aminoindants," J.Med.Chem., 17, 409, 1984.

31. Hansch, C., S.H. Unger, and A.B. Gorsythe,"Stratehy in Drug Design. Cluster Analysis as an Aid in the Selective of Substituent," J.Med.Chem, 16, 1217, 1973.

32. Hopfinger, A.J.,"A QSAR Investigation of Dihydrofolate Reductase Inhibition by Baker Triazines Based upon Molecular ShapeAnalysis," J. Am. Chem. Soc., 102, 7196-7206, 1980.

33. Bass, G.E., D.R. Hudson, J.E. Parker and W.P. Purcell, "Mechaism of Antimalarial Activity of Chloroqine Analogs from Quantitative Structure-Activity Studies Free Energy Related Model," J. Med. Chem, 14(4), 275-286, 1971.

34. Pople, J.A. and G.A. Segal, "Approximate Self-Consistent Molecular Orbital Theory. III. CNDO Results of AB_2 and AB_3 Systems,"

35. Mulliken, R.S., "Electronic Population Analysis on LCAO-MO Molecular Wave Functions I.," J. Chem. Phys., 29, 1833, 1955.

36. Schnaare, R.S. and A.N.Martin, "Quantum Chemistry in Drug Design," J. Pharm. Sci., 54(12), 1707-1713, 1965.

37. Gund, P., J.D. Andose, J.B. Rhodes and G.M. Smith, "Three-Dimensional Molecular Modeling and Drug Design," Science, 208(27), 1980.

38. Crow, J., O. Wassermann and W.C. Holland,"Molecular Orbital Calculations on a New Series of Substituted-Phynyl Choline Ehters," J. Med. Chem., 11, 764-766.

39. Boyd, R.J. and M.A. Whithead, "An SCF-MO-CNDO Study of Equilibrium Geometris, Force Constant, and Bonding Energies:CNDO/BW. Part I, Parameterization," J.C.S. Dalton, 3, 73-76, 1972.

40. Scabo, A. and N.S. Ostlund, Modern Quantum Chemistry, Macmillan Pub. Co., INC, New York, U.S.A., 1982.

41. Mulliken, R.S., "Electronic Population Analysis on LCAO-MO Molecular Wave Functions,"J.Chem.Phys., 23,1833,1955.

42. Doungdee, P.B., S.U. Kokpol and B.M. Rode, "A Comparative Study of the Hydration of Aliphatic Amino Acids by CNDO/2 and Ab initio Molecular Orbital Calculations," J.Sci.Soc.Thailand, 11, 113-119, 1985.

43. Rode, B.M., A. Engelbrecht and W. Jakubetz, "CNDO/2 Calculations of Halogenated Acetic Acids and Their Dimeric Adducts," Chem.Phys.Letts., 18(2), 285-289, 1973.

44. Pople, A.J. and D.L. Beveridge, Approximate Molecular Orbital Theory, McGraw-Hill Book company, 1970.

45. Pople, A.J. and G.A. Segal, J.Chem.Phys., 43, 5136, 1965.

46. Klopman, G. and R.C. Evans, "The Neglect-of-Differential-Overlap Methods of Molecular Orbital Theory," Semiempirical Methods of Electronic Structure Calculation (Part A)(G.A. Segal ed.), Plenum Press, New York, U.S.A., 1977.

47. Doungdee, P., "A Theoretical Study of the Influence of Solvent on the Internal Rotation of Glycine Zwitterion" M.S. Thesis, Chulalongkorn University, 1985.

48. Hahn, F.E., G.L. O'Brien, J.Ciak, J.L. Allison and J.G. Blenick, "Studies on Modes of Action of Chloroquine, Quinacrine and on Chloroquine Resistance," Military Medcine, 9, 1071-1089, 1966.

49. Mulliken, R.S., C.A. Rieke, D. Orloff, and M. Orloff, J.Chem.Phys., 17, 1248, 1949.

50. O'Brien, R.L. and F.E. Hahn,"Chloroquine Structural Requirements for Binding to Deoxyribonucleic Acid and Antimalarial Activity," Antimicrobial Agents and Chemotherapy, 315-320, 1965.

51. Ohnmacht, C.J., A.R. Pelel and R.E. Lutz, "Antimalarials. 7(Bis(trifluoromethyl)- ∝ -(2-piperidyl)-4-quinoline methanols," J. Med. Chem., 14(10), 926-928, 1971.

52. Singh, P.P. and G.P. Dutla,"Antimalarial Activity of Mefloquine and Chloroquine Against Blood Induced Plasmodium Knowlesi Infection in Rhesus Monkeys," Indian J. Med. Res, 73(suppl), 1, 23-28, 1981.

53. Rode, B.M., M.G. Schwengdinger, S.U. Kokpol, S.V. Hannongbua and S. Polman, "Quantum Pharmacological Studies on Antimalarial Drugs," (submitted for publication).

54. Mulliken, R.S., C.A. Rieke, D.orloff, and M. Orloff, "Electronic Population Analysis on LCAO-MO Molecular Wave Functions. II. Overlap Populations, Bond Orders, and Covalent Bond energies," J.Chem.Phys., 17, 1248, 1949.

55. Boykin, D.W., A.R. Palel and R.E. Leitz,"Antimalarials, IV. A New Synthesis of α-(2-Pyridyl)-and α-(2-Piperidyl)-2-aryl-4-quinolinemethancls," J. Med. Chem., 3, 273-277, 1968.

56. Pinder, R.M. and A. Burger, "Antimalarial II. (2-Piperidyl) and α-(2-Pyridyl)-2-trifluoromethyl-4-quinolinemethanols," J. Med Chem, 11, 267-269, 1968.

57. Huzinaga, S.,Gaussian Basis Sets for Molecular Calculations, Elsevier Science Pub., New York, 1984.

58. Sutton, L.E., O. Kennard, A.A. Meccnnell and A.D. Mitchell, Table of Interatomic Distances and Configuration in molecules and Ions, London, The Chemical Society, Burlington House, W1, 1965.

59. Davidson, D.E., A.L. Ager, J.L. Brown, F.E. Chapple, R.E Whitemire and R.N. Rossen,"New Tissue Schizontocidal Antimalarial Drugs," Bull. W.H.O., 59(3), 463-479, 1981.

60. Warhurst, D.C.,"Antimalarial Drugs:Mode of Action and Resistence," J. Antimicro. Chem., 18(suppl.), 51-59, 1986.

61. Kokpol, S.U,,S.V. Hannongbua, N. Thongrit, S. Polman, B.M. Rode and M.G. Schwengdinger,"Quantum Pharmacological Studies on Primaquine Antimalarial," (submitted for publication).

62. Davidson, M.W., B.G. Griggs, D.W. Boydin and W.D. Wilson,"Mefloquine, a Clinically seful uinolinemethanol Antimalarial Which Does Not Signigicantly Bind to DNA," Nature, 254(4), 632-634, 1975.

63. George, R.C., R.L. Nagal and M.E. Fabry,"On the Mechanism for The Red-Cell Occumulation of Mefloquine, An Antimalarial Drugs," Biochemica et Biophysica Acta, 803, 174-181, 1984.

64. Peters, W., R.E. Howells, J. Portes, B.L. Robinson, S. Thomas and D.C. Warherst,"The Chemotherapy of Rodent Malaria, XXVII Studies on Mefloquine (WR 142,490)," Annal. Trop. Med. Parasit., 71(4), 407-418, 1977.

APPENDIX I

KOGEN Program

```
0001 C     ****************
00002 C
00003      PROGRAM KOGEN
00004 C
00005 C     ****************
00006 C
00007 C     A SIMPLE PROGRAM FOR CALCULATING THE CARTESIAN COORDINATES OF THE
00008 C     ATOMS IN A MOLECULE AND PERFORMING SOME OTHER OPERATIONS.
00009 C     WRITTEN BY MICHAEL PROBST.
00010 C     VERSION 10/NOV/84 ( WITH CORRECTED TABLE FUNCTION )
00011 C
00012 C     HOW TO USE KOGEN :
00013 C     ------------------
00014 C
00015 C     A) SYNTAX :
00016 C
00017 C     GENERAL :
00018 C
00019 C
00020 C     ONE ENTERS A COMMAND ( 'INPUT', 'TABLE' ETC. ) AND IN MOST CASES
 0021 C     ONE OR MORE PIECES OF DATA FOR THAT COMMAND.
00022 C     THE COMMAND MUST ALWAYS BE AT THE BEGINNING OF THE LINE.
00023 C     THE SEPARATOR IS ALWAYS A BLANK.
00024 C     IF THE DATA FOLLOWS THE COMMAND IN THE SAME LINE, THE COMMAND
00025 C     IS EXECUTED IMMEDIATELY.
00026 C     IF ONLY THE COMMAND IS ENTERED IN THE LINE, THE PROGRAM TELLS
00027 C     YOU WHICH KIND OF DATA YOU HAVE TO ENTER.
00028 C
00029 C     DATA STRUCTURES :
00030 C
00031 C     EACH PIECE OF DATA CAN BE THE FOLLOWING :
00032 C
00033 C     1) A NUMBER
00034 C     2) A SINGLE ATOM
00035 C     3) A RANGE OF ATOMS
00036 C     4) A VECTOR
00037 C     5) A LINE
00038 C     6) A NAME
00039 C
00040 C     AD 1) A NUMBER IS A REAL NUMBER. THE INPUT CAN BE WITHOUT COMMA.
 0041 C     AD 2) A SINGLE ATOM IS SPECIFIED BY ITS NAME OR ITS NUMBER.
00042 C     AD 3) A RANGE OF ATOMS IS SPECIFIED BY ATOM1:ATOM2
00043 C           DEFAULT FOR ATOM1 AND ATOM2 : FIRST AND LAST
00044 C     AD 4) A VECTOR IS SPECIFIED BY NUMBER/NUMBER/NUMBER OR BY AN ATOM.
00045 C     AD 5) A LINE IS SPECIFIED BY VECTOR1//VECTOR2
00046 C             DEFAULT FOR VECTOR1 AND VECTOR2 : 0/0/0
00047 C     AD 6) A NAME IS A CHARACTERSTRING THAT MUST NOT BEGIN WITH A
00048 C           NUMBER.
```

```
00049 C
00050 C     B) COMMANDS
00051 C
00052 C     THE NAMES OF THE COMMANDS CAN BE ABBREVIATED AS LONG THE ARE
00053 C     UNIQUE.
00054 C
00055 C     1) INSERTING ONE ATOM INTO THE MOLECULE :
00056 C
00057 C      INPUT (TEXT) ((SPECIFICATION)) (VECTOR)
00058 C
00059 C      (TEXT) MUST BE A STRING WITH 1 TO 5 CHARACTERS.
00060 C      IT SHOULD BE UNIQUE FOR EACH ATOM.
 0061 C      (SPECIFICATION)  SPECIFIES THE COORDINATE SYSTEM.
00062 C        DEFAULT ... CARTESIAN
00063 C        CART    ... CARTESIAN
00064 C        POL     ... POLAR
00065 C        CYL     ... CYLINDRICAL
00066 C      (VECTOR) IS X/Y/Z       IF CART OR NOTHING
00067 C                  R/PHI/THETA IF POL
00068 C                  R/PHI/Z     IF CYL
00069 C
00070 C     2) I* (TEXT) ((SPECIFICATION)) (VECTOR)
00071 C
00072 C      ALLOWS SIMILAR INPUT OF MANY ATOMS.
00073 C      ( UNTIL TEXT='END' IS ENCOUNTERED).
00074 C
00075 C     3) DEL (RANGE)
00076 C
00077 C      REMOVES THE SPECIFIED ATOMS FROM THE MOLECULE.
00078 C
00079 C
00080 C     4) READ (FILENAME) ((STARTING LINE) ((FORMAT)))
 0081 C
00082 C      READS A SET OF ATOMS FROM A FILE AND INSERTS IT INTO THE MOLECULE
00083 C      (FILENAME) IS THE NAME OF A FILE WHERE THE COORDINATES APPEAR
00084 C      IN THE FORMAT THAT IS USED FOR THE IBMOL PROGRAM. THE FILE MUST
00085 C      ONLY CONTAIN THE COORDINATES.
00086 C
00087 C     4) WRITE (FILENAME)
00088 C
00089 C      WRITES THE MOLECULE TO A FILE.
00090 C      (FILENAME) IS THE NAME OF THE FILE TO WHICH THE COORDINATES ARE
00091 C      WRITTEN IN THE FORMAT USED BY IBMOL.IF THE FILENAME HAS ALREADY
00092 C      APPEARED IN A FORMER WRITE STATEMENT, THE COORDINATES ARE
00093 C      APPENDED TO THE END OF THE FILE.
00094 C
00095 C     5) DISP
00096 C
00097 C      THIS COMMAND DISPLAYS THE COORDINATES OF THE ATOMS.
00098 C
00099 C     6) DIST (LINE)
00100 C
 0101 C      DISPLAYS THE DISTANCE BETWEEN TWO ATOMS.
00102 C
00103 C     7) ANGLE (VECTOR1) (VECTOR2) (VECTOR3)
00104 C
00105 C      DISPLAYS THE ANGLE (1-2-3) BETWEEN THE ATOMS WITH THE TEXTS.
00106 C
```

```
00107 C     8) DIANG (LINE1) (LINE2)
00108 C
00109 C      DISPLAYS THE (DIHEDRAL) ANGLE BETWEEN THE 2 PLANES DEFINED
00110 C      BY THE ATOMS 1,2,3 AND 2,3,4.
00111 C
00112 C     9) ATOM (VECTOR) (LINE) (TEXT) (DIST) (ANG) (DIANG)
00113 C
00114 C      INSERTS A NEW ATOM INTO THE MOLECULE. THE NEW ATOM IS THE
00115 C      4 TH IN THE STRUCTURE 1-2-3-4. (DIST) IS THE DISTANCE BETWEEN
00116 C      3 AND 4, (ANG) IS THE ANGLE 2-3-4 AND (DIANG) IS THE ANGLE
00117 C      BETWEEN THE PLANES OF 1-2-3 AND 2-3-4.
00118 C
00119 C     10) SHIFT (RANGE) (VECTOR)
00120 C
 0121 C      SUBTRACTS THE VECTOR (X,Y,Z) FROM THE COORDINATES OF THE
00122 C      SPECIFIED ATOMS.
00123 C
00124 C     11) SHIFTBACK
00125 C
00126 C      ADDS THE VECTOR (X,Y,Z) OF THE LAST SHIFT TO ALL ATOMS.
00127 C
00128 C     12) TURN (RANGE) (LINE) (ANGLE)
00129 C
00130 C      TURNS THE SPECIFIED ATOMS AROUND THE AXIS DEFINED BY LINE
00131 C      BY THE ANGLE (ANGLE).
00132 C
00133 C     13) TURNBACK
00134 C
00135 C      TURNS THE MOLECULE AROUND THE AXIS DEFINED BY THE LAST TURN
00136 C      COMMAND BY THE OPPOSITE ANGLE OF THE LAST TURN COMMAND.
00137 C
00138 C     14) HELP
00139 C
00140 C      GIVES YOU INFORMATION ABOUT THE COMMANDS.
 0141 C
00142 C     15) XTURN, YTURN, OR ZTURN (RANGE) (VECTOR)
00143 C
00144 C      TURN THE VECTOR (X,Y,Z) AROUND THE ORIGIN INTO THE X, Y, OR Z-
00145 C      AXIS AND THE SPECIFIED PART OF THE MOLECULE WITH HIM.
00146 C
00147 C     16) XTURNBACK , YTURNBACK , ZTURNBACK
00148 C
00149 C      PERFORM THE INVERSE OPERATION AS THE LAST XTURN, YTURN OR ZTURN -
00150 C      COMMANDS.
00151 C
00152 C     17) TABLE (MINDIST)
00153 C
00154 C      GIVES YOU A TABLE OF THE DISTANCES AND ANGLES BETWEEN ATOMS,
00155 C      WHOSE DISTANCES ARE SMALLER THAN MINDIST.
00156 C
00157 C     18) MOVE (RANGE) (LINE) (DIST)
00158 C
00159 C      MOVES THE PART OF THE MOLECULE FROM TEXT1 UNTIL TEXT2 BY THE
00160 C      SPECIFIED DISTANCE ALONG THE VECTOR X,Y,Z .
 0161 C
00162 C     19) MOVCONT OR MOCO
00163 C
00164 C      REPEATES THE LAST MOVE - COMMAND ( IF THERE WAS ANY)
```

```
C
C    20) LINEUP (RANGE) (LINE1) (LINE2)
C
C     MOVES THE ATOMS IN THE RANGE SO, THAT LINE1 AND LINE2 BECOME
C     ONE LINE AND VECTOR2 OF LINE1 AND ATOM1 OF LINE2 COINCIDE.
C
C    21) /D
C
C     /D IS A POSTFIX THAT DISPAYS THE COORDINATES AFTER EXECUTION
C     OF THE COMMAND.
C
C    FORMATS OF THE TERMINAL INPUT :
C
C     ALL COMMAND NAMES HAVE TO START IN THE FIRST COLUMS OF A LINE.
C     AFTER ENTERING THE COMMAND NAME, THE PROGRAM ASKS FOR ENTERING
C     THE OTHER PARAMETERS, IF THERE ARE ANY.
C     FILENAMES ARE FORTRAN FILE NAMES AND HAVE TO BE ENCLOSED
C     IN ' CHARACTERS.
C     THE PARAMETERS ARE READ IN FREE FORMAT.
C     THE ANGLES APPEAR IN DEGREES.
C     IF TWO OR MORE ATOMS HAVE THE SAME TEXT, THE FIRST IS TAKEN.
C
C*****************************************************************
C
      PARAMETER ( NATOMS=100 )
      COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
      CHARACTER*5 TEXT(NATOMS)
      COMMON /B/ TEXT
C
      CHARACTER*16 INTEXT
C
      NUMBER = 0
C     *** INPUT DRIVER ***
C
      WRITE (*,'(/A//)') ' *** ENTER COMMANDS OR HELP ***'
  100 CONTINUE
      WRITE (*,'(A)') ' ->'
      READ (*,'(A)',END=110) INTEXT
      IF (INTEXT.EQ.'INPUT'.OR.INTEXT(:2).EQ.'I ') THEN
        CALL INPUT (*120)
      ELSEIF (INTEXT.EQ.'I*') THEN
        CALL ISTAR (*120)
      ELSEIF (INTEXT(:4).EQ.'READ'.OR.INTEXT(:2).EQ.'R ') THEN
        CALL READ (*120)
      ELSEIF (INTEXT(:5).EQ.'WRITE'.OR.INTEXT(:2).EQ.'W ') THEN
        CALL WRITE (*120)
      ELSEIF (INTEXT(:4).EQ.'DISP'.OR.INTEXT(:2).EQ.'D ') THEN
        CALL DISP
      ELSEIF (INTEXT(:4).EQ.'DIST') THEN
        CALL DIST(*120)
      ELSEIF (INTEXT(:3).EQ.'ANG') THEN
        CALL ANG(*120)
      ELSEIF (INTEXT(:5).EQ.'DIANG') THEN
        CALL DIANG(*120)
      ELSEIF (INTEXT(:4).EQ.'ATOM'.OR.INTEXT(:2).EQ.'A ') THEN
        CALL ATOM(*120)
      ELSEIF (INTEXT(:9).EQ.'SHIFTBACK'.OR.INTEXT(:2).EQ.'SB') THEN
```

```
00223          CALL SHBACK(*120)
00224        ELSEIF (INTEXT(:5).EQ.'SHIFT'.OR.INTEXT(:2).EQ.'S ') THEN
 0225          CALL SHIFT(*120)
00226        ELSEIF (INTEXT(:8).EQ.'TURNBACK'.OR.INTEXT(:2).EQ.'TB') THEN
00227          CALL TBACK(*120)
00228        ELSEIF (INTEXT(:4).EQ.'TURN'.OR.INTEXT(:2).EQ.'T ') THEN
00229          CALL TURN(*120)
00230        ELSEIF (INTEXT(:4).EQ.'HELP'.OR.INTEXT(:2).EQ.'H ') THEN
00231          CALL HELP
00232        ELSEIF (INTEXT(:3).EQ.'DEL') THEN
00233          CALL DEL(*120)
00234        ELSEIF (INTEXT(:9).EQ.'XTURNBACK'.OR.INTEXT(:4).EQ.'XTB '.OR.
00235       +         INTEXT(:9).EQ.'YTURNBACK'.OR.INTEXT(:4).EQ.'YTB '.OR.
00236       +         INTEXT(:9).EQ.'ZTURNBACK'.OR.INTEXT(:4).EQ.'ZTB ') THEN
00237          CALL AXBACK(*120)
00238        ELSEIF (INTEXT(:5).EQ.'XTURN') THEN
00239          CALL XTURN(*120)
00240        ELSEIF (INTEXT(:5).EQ.'YTURN') THEN
00241          CALL YTURN(*120)
00242        ELSEIF (INTEXT(:5).EQ.'ZTURN') THEN
00243          CALL ZTURN(*120)
00244        ELSEIF (INTEXT(:3).EQ.'TAB') THEN
 0245          CALL TABLE(*120)
00246        ELSEIF (INTEXT(:3).EQ.'MOV') THEN
00247          CALL MOVE(*120)
00248        ELSEIF (INTEXT(:6).EQ.'MOVECO'.OR.INTEXT(:4).EQ.'MOCO') THEN
00249          CALL MOCONT
00250        ELSE
00251          GOTO 200
00252        ENDIF
00253          GOTO 300
00254    200 CONTINUE
00255        IF (INTEXT(:6).EQ.'LINEUP') THEN
00256          CALL LINEUP(*120)
00257        ELSEIF (INTEXT(:4).EQ.'TEST') THEN
00258          CALL TEST(*120)
00259        ELSE
00260          WRITE (*,*) ' * ERROR: COMMAND ',INTEXT,' DOES NOT EXIST. ***'
00261        ENDIF
00262 C
00263        GOTO 300
00264 C
 0265 C      *** POSTFIX ***
00266 C
00267    300 CONTINUE
00268        IF (INDEX(INTEXT(:6),'/D').NE.0) CALL DISP
00269        GOTO 100
00270 C
00271 C      *** POSTFIX ***
00272 C
00273 C      *** INPUT DRIVER ***
00274 C
00275    120 CONTINUE
00276        WRITE (*,*) ' * ERROR: WRONG PARAMETERS FOR ',INTEXT,'. ***'
00277        GOTO 100
00278 C
00279    110 CONTINUE
00280        STOP ' *** FINISHED ***'
```

```
00281       END
00282 C
00283 C     ********************
00284 C
 0285       SUBROUTINE TEST (*)
00286 C
00287 C     ********************
00288 C
00289 C     THIS SUBROUTINE IS BEING USED ONLY FOR VARIOUS TEST REASONS.
00290 C
00291 C     ******************** INPROC - DECLARATIONS ************************
00292       INTEGER ARGMAX
00293       PARAMETER ( ARGMAX=10 )
00294       CHARACTER LINE*60
00295       INTEGER INTARG(ARGMAX,2)
00296       CHARACTER*15 TXTARG(ARGMAX,2)
00297       REAL REARG(ARGMAX,6)
00298       CHARACTER*7 TYPE(ARGMAX)
00299 C     READ(*,'(A)') LINE
00300 C     CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00301 C     ******************** INPROC - DECLARATIONS ************************
00302 C
00303   110 CONTINUE
00304       WRITE(*,'(/A)') ' ENTER AN INPUTLINE :'
 0305       READ(*,'(A)',END=100) LINE
00306       WRITE (*,'(/A//T2,A/)') ' YOU ENTERED THE LINE :',LINE
00307       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00308       WRITE (*,'(/A/)') ' DECODED :'
00309       WRITE (*,*) ' NUMBER OF ARGUMENTS : ',NARGS
00310       WRITE(*,'(A/)')
00311      +' NUMBER  TYPE      TXTARG1     TXTARG2       INTARG1     INTARG2'
00312       WRITE(*,'(I5,T12,3A,2I10)')
00313      + (I,TYPE(I),TXTARG(I,1),TXTARG(I,2),INTARG(I,1),INTARG(I,2),I=1,
00314      + ARGMAX)
00315       WRITE(*,'(//A/)') ' REARGS = '
00316       WRITE(*,'(I5,TR10,6F10.5)') (I,(REARG(I,J),J=1,6),I=1,ARGMAX)
00317 C
00318       IF (TXTARG(1,1)(:3).NE.'END') GOTO 110
00319       RETURN
00320 C
00321   100 CONTINUE
00322       RETURN 1
00323       END
00324 C
 0325 C     ********************
00326 C
00327       SUBROUTINE INPUT(*)
00328 C
00329 C     ********************
00330 C
00331 C     *** INSERT AN ATOM INTO THE MOLECULE ***
00332 C
00333 C     ******************** INPROC - DECLARATIONS ************************
00334       INTEGER ARGMAX
00335       PARAMETER ( ARGMAX=10 )
00336       CHARACTER LINE*60
00337       INTEGER INTARG(ARGMAX,2)
00338       CHARACTER*15 TXTARG(ARGMAX,2)
```

```
00339       REAL REARG(ARGMAX,6)
00340       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
00341       LOGICAL CHEK
00342 C     ****************** INPROC - DECLARATIONS **********************
00343 C
00344       PARAMETER ( NATOMS=100 )
 0345       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
00346       CHARACTER*5 TEXT(NATOMS)
00347       COMMON /B/ TEXT
00348       CHARACTER*4 MODE
00349       REAL XX,YY,ZZ
00350       INTEGER POS,NUM
00351 C
00352 C     ****************** INPROC - PROCEDURE **********************
00353       WRITE(*,*) ' ENTER  TEXT X/Y/Z :'
00354       READ(*,'(A)',END=100) LINE
00355       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00356       NARCHK = 2
00357       TYPCHK(1) = 'TEXT'
00358       TYPCHK(2) = 'VECTOR'
00359       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00360       IF (CHEK) THEN
00361          MODE = 'CART'
00362          POS = 2
00363       ELSE
00364          NARCHK = 3
 0365          TYPCHK(1) = 'TEXT'
00366          TYPCHK(2) = 'TEXT'
00367          TYPCHK(3) = 'VECTOR'
00368          CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
00369          MODE = TXTARG(2,1)
00370          POS = 3
00371       ENDIF
00372 C     ****************** INPROC - PROCEDURE **********************
00373 C
00374       NUM = NUMBER + 1
00375       TEXT(NUM) = TXTARG(1,1)
00376       IF (MODE(:4).EQ.'CART') THEN
00377          X(NUM) = REARG(POS,1)
00378          Y(NUM) = REARG(POS,2)
00379          Z(NUM) = REARG(POS,3)
00380       ELSEIF (MODE(:3).EQ.'POL') THEN
00381          CALL POLAR (XX,YY,ZZ,REARG(POS,1),REARG(POS,2),REARG(POS,3))
00382          X(NUM) = XX
00383          Y(NUM) = YY
00384          Z(NUM) = ZZ
 0385       ELSEIF (MODE(:3).EQ.'CYL') THEN
00386          ANGRAD = REARG(POS,2)*3.141592654/180.0
00387          X(NUM) = REARG(POS,1)*COS(ANGRAD)
00388          Y(NUM) = REARG(POS,1)*SIN(ANGRAD)
00389          Z(NUM) = REARG(POS,3)
00390       ELSE
00391          WRITE(*,*) ' MODE NOT EQUAL CART, POL OR CYL'
00392          GOTO 100
00393       ENDIF
00394 C
00395       NUMBER = NUM
00396       RETURN
00397 C
```

```
00398   100 CONTINUE
00399       RETURN 1
00400       END
00401 C
00402 C     ************************************
00403 C
00404       SUBROUTINE POLAR (X,Y,Z,R,PHI,THETA)
 0405 C
00406 C     ************************************
00407 C
00408 C     *** CALCULATES CARTESIAN COORDINATES FROM POLARCOORDINATES ***
00409 C
00410       REAL X,Y,Z,R,PHI,THETA,PI
00411       PARAMETER (PI=3.141592654)
00412 C
00413       PHIRAD = PHI*PI/180.0
00414       THERAD = THETA*PI/180.0
00415       X = R*SIN(THERAD)*COS(PHIRAD)
00416       Y = R*SIN(THERAD)*SIN(PHIRAD)
00417       Z = R*COS(THERAD)
00418 C
00419       RETURN
00420       END
00421 C
00422 C
00423 C     ********************************************
00424 C
 0425       SUBROUTINE CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*)
00426 C
00427 C     ********************************************
00428 C
00429 C     *** CHECK, IF INPUT FROM SUBROUTINE INPROC WAS ACCEPTABLE ***
00430 C     *** THIS SUBROUTINE IS USED IF THERE IS ONLY ONE POSSIBILITY ***
00431 C     *** OF CORRECT INPUT STRUCTURE.
00432 C
00433       INTEGER NARGS,NARCHK
00434       CHARACTER*6 TYPE(*),TYPCHK(*)
00435 C
00436       IF ( NARGS.NE.NARCHK ) THEN
00437          WRITE(*,'(A,I2,A,I2,A)')
00438      +    ' *** ERROR : ',NARGS,' INSTEAD OF ',NARCHK,' ARGUMENTS.'
00439          GOTO 100
00440       ENDIF
00441       DO 200,I=1,NARGS
00442          IF (TYPE(I).NE.TYPCHK(I)) THEN
00443             WRITE(*,'(A,I2,A)')
00444      +       ' *** ERROR : WRONG TYPE OF ARGUMENT ',I,' ***'
 0445             GOTO 100
00446          ENDIF
00447   200 CONTINUE
00448 C
00449       RETURN
00450 C
00451   100 CONTINUE
00452       RETURN 1
00453       END
00454 C
00455 C     ********************************************
```

```
00456 C
00457       SUBROUTINE CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00458 C
00459 C     ************************************************
00460 C
00461 C     *** CHECK, IF INPUT FROM SUBROUTINE INPROC WAS ACCEPTABLE   ***
00462 C     *** THIS SUBROUTINE IS USED IF THERE IS MORE THAN ONE       ***
00463 C     *** POSSIBILITY OF CORRECT INPUT STRUCTURE.                 ***
00464 C
 0465       INTEGER NARGS,NARCHK
00466       CHARACTER*6 TYPE(*),TYPCHK(*)
00467       LOGICAL CHEK
00468 C
00469 C     PRINT*,'*',TYPE(1),'* SHOULD BE *',TYPCHK(1),'*'
00470       CHEK = .TRUE.
00471       IF ( NARGS.NE.NARCHK ) THEN
00472          CHEK= .FALSE.
00473          RETURN
00474       ELSE
00475          DO 200,I=1,NARGS
00476             IF (TYPE(I).NE.TYPCHK(I)) THEN
00477                CHEK = .FALSE.
00478                RETURN
00479             ENDIF
00480   200    CONTINUE
00481       ENDIF
00482 C
00483       RETURN
00484       END
 0485 C
00486 C     ******************************************************
00487 C
00488       SUBROUTINE INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*)
00489 C
00490 C     ******************************************************
00491 C
00492 C     *** DECODES INPUT INTO KOGEN - SYNTAX ***
00493 C
00494 C     THE IDEA OF THIS SUBROUTINE IS THE FOLLOWING :
00495 C
00496 C     THE INPUTLINE IS SEPARATED IN PIECES OF INPUTDATA.
00497 C     EACH PIECE OF INPUTDATA IS INTERPRETED ACCORDING TO ITS TYPE
00498 C     AND SIMPLIFIED AS MUCH AS POSSIBLE ( WITHOUT KNOWLEDGE WHICH
00499 C     DATA IS EXPECTED ). ( THIS IS THE 'CATALOGUED AS' ROW IN THE
00500 C     FOLLOWING TABLE.) THEN THE SINGLE VALUES AND THEIR MEANING
00501 C     IS RETURNED TO THE CALLING SUBROUTINE.
00502 C
00503 C
00504 C     THE FOLLOWING KINDS OF DATA ARE POSSIBLE (I,J,K ... INTEGER,
 0505 C                                              X,Y,Z ... REAL ):
00506 C
00507 C          INPUT        CATALOGUED AS      CAN BE
00508 C
00509 C     INTEGER            INTEGER           GROUP, SCALAR, VECTOR
00510 C     REAL               REAL              SKALAR
00511 C     INTEGER:INTEGER    RANGE             GROUP
00512 C     INTEGER:TEXT       RANGE             GROUP
00513 C     TEXT:INTEGER       RANGE             GROUP
```

```
00514 C     TEXT:TEXT           RANGE              GROUP
00515 C     :INTEGER            RANGE              GROUP
00516 C     INTEGER:            RANGE              GROUP
00517 C     :TEXT               RANGE              GROUP
00518 C     TEXT:               RANGE              GROUP
00519 C     :                   RANGE              GROUP
00520 C     X/Y/Z               VECTOR             VECTOR
00521 C     I/J/K               VECTOR             VECTOR
00522 C     TEXT                TEXT               GROUP, VECTOR
00523 C     X1/Y1/Z1//X2/Y2/Z2  LINE               LINE
00524 C     TEXT//TEXT          LINE               LINE
 0525 C     I//X/Y/Z            LINE               LINE
00526 C     I//TEXT             LINE               LINE
00527 C     I//I                LINE               LINE
00528 C     //TEXT              LINE               LINE
00529 C     TEXT//              LINE               LINE
00530 C
00531       INTEGER ARGMAX
00532       PARAMETER ( ARGMAX=10 )
00533       PARAMETER ( NATOMS=100 )
00534       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
00535       CHARACTER*5 TEXT(NATOMS)
00536       COMMON /B/ TEXT
00537 C
00538       INTEGER NARGS,SEMICO,SLASH1,SLASH2,DOSLSH,SP1,INTARG(ARGMAX,2)
00539       INTEGER INTERM
00540       CHARACTER*30 WORD(ARGMAX),W,W2
00541       REAL REARG(ARGMAX,6)
00542       CHARACTER*(*) TYPE(ARGMAX),TXTARG(ARGMAX,2),LINE
00543       CHARACTER*80 LINREM,LIREM2
00544 C
 0545 C     *** TESTMODE ***
00546 C
00547 C     WRITE(*,*) '*** ENTERING INPROC ***'
00548 C     WRITE(*,'(A)') LINE
00549       DO 101,I=1,ARGMAX
00550       INTARG(I,1) = 0
00551       INTARG(I,2) = 0
00552       REARG(I,1) = 0
00553       DO 102,J=1,6
00554       REARG(I,J) = 0
00555   102 CONTINUE
00556       TYPE(I) = ' '
00557       TXTARG(I,1) = ' '
00558       TXTARG(I,2) = ' '
00559   101 CONTINUE
00560 C
00561 C     *** TESTMODE ***
00562 C
00563 C     *** SEPARATE THE ARGUMENTS ***
00564 C
 0565       IEND = 0
00566       LINREM = LINE
00567       LINLEN = LEN (LINREM)
00568       DO 200,I=1,ARGMAX
00569 C        *** FIND FIRST POSITION IN WORD ***
00570          DO 500,IANF=1,LINLEN
00571             IF (LINREM(IANF:IANF).NE.' ') GOTO 600
```

```
00572   500    CONTINUE
00573          GOTO 700
00574   600    CONTINUE
00575          LIREM2 = LINREM(IANF:)
00576          LINREM = LIREM2
00577 C        *** FIND FIRST POSITION IN WORD ***
00578 C        *** FIND LAST POSITION IN WORD ***
00579          IEND = INDEX(LINREM,' ')-1
00580          IF ( IEND.EQ.-1 ) IEND = LINLEN
00581 C        *** FIND LAST POSITION IN WORD ***
00582          WORD(I) = LINREM(:IEND)
00583          LIREM2 = LINREM(IEND+1:)
00584          LINREM = LIREM2
 0585   200 CONTINUE
00586       WRITE (*,*) ' TOO MANY ARGUMENTS '
00587       GOTO 100
00588   700 CONTINUE
00589       NARGS = I-1
00590 C
00591 C     *** SEPARATE THE ARGUMENTS ***
00592 C     *** CLASSIFY THE ARGUMENTS ***
00593 C
00594       DO 850,I=1,NARGS
00595          SEMICO = INDEX(WORD(I),':')
00596          DOSLSH = INDEX(WORD(I),'//')
00597 C        IF(DOSLSH.NE.0) PRINT*,' THERE IS A DOSLSH '
00598          SLASH1 = INDEX(WORD(I),'/')
00599          IF ( SEMICO.NE.0 ) THEN
00600 C        *** RANGE ***
00601             TYPE(I) = 'RANGE'
00602             IF ( SEMICO.EQ.1 ) THEN
00603 C           *** LEFT DEFAULT RANGE BORDER***
00604                INTARG (I,1) = 1
 0605 C           *** LEFT DEFAULT RANGE BORDER ***
00606             ELSEIF ( WORD(I)(:1).EQ.'1'.OR.WORD(I)(:1).EQ.'2'.OR.
00607      +               WORD(I)(:1).EQ.'3'.OR.WORD(I)(:1).EQ.'4'.OR.
00608      +               WORD(I)(:1).EQ.'5'.OR.WORD(I)(:1).EQ.'6'.OR.
00609      +               WORD(I)(:1).EQ.'7'.OR.WORD(I)(:1).EQ.'8'.OR.
00610      +               WORD(I)(:1).EQ.'9'.OR.WORD(I)(:1).EQ.'0' ) THEN
00611 C           *** INDEX DEFINED LEFT RANGE BORDER ***
00612                W = WORD(I)(:SEMICO-1)
00613                READ (W,'(BN,I15)') INTARG(I,1)
00614 C           *** INDEX DEFINED LEFT RANGE BORDER ***
00615             ELSE
00616 C           *** TEXT DEFINED LEFT RANGE BORDER ***
00617                W = WORD(I)(:SEMICO-1)
00618                CALL ATIND ( W,INTARG(I,1),*100 )
00619 C           *** TEXT DEFINED LEFT RANGE BORDER ***
00620             ENDIF
00621             SP1 = SEMICO+1
00622             IF ( WORD(I)(SP1:).EQ.' ' ) THEN
00623 C           *** RIGHT DEFAULT RANGE BORDER***
00624                INTARG (I,2) = NUMBER
 0625 C           *** RIGHT DEFAULT RANGE BORDER***
00626             ELSEIF ( WORD(I)(SP1:SP1).EQ.'1'.OR.
00627      +               WORD(I)(SP1:SP1).EQ.'2'.OR.
00628      +               WORD(I)(SP1:SP1).EQ.'3'.OR.
00629      +               WORD(I)(SP1:SP1).EQ.'4'.OR.
```

```
00630     +              WORD(I)(SP1:SP1).EQ.'5'.OR.
00631     +              WORD(I)(SP1:SP1).EQ.'6'.OR.
00632     +              WORD(I)(SP1:SP1).EQ.'7'.OR.
00633     +              WORD(I)(SP1:SP1).EQ.'8'.OR.
00634     +              WORD(I)(SP1:SP1).EQ.'9'.OR.
00635     +              WORD(I)(SP1:SP1).EQ.'0' ) THEN
00636 C            *** INDEX DEFINED RIGHT RANGE BORDER ***
00637                W = WORD(I)(SP1:)
00638                READ (W,'(BN,I15)') INTARG(I,2)
00639                IF (INTARG(I,2).GT.NUMBER) GOTO 100
00640 C            *** INDEX DEFINED RIGHT RANGE BORDER ***
00641              ELSE
00642 C            *** TEXT DEFINED RIGHT RANGE BORDER ***
00643                W = WORD(I)(SP1:)
00644                CALL ATIND ( W,INTARG(I,2),*100 )
 0645 C            *** TEXT DEFINED RIGHT RANGE BORDER ***
00646              ENDIF
00647 C          *** RANGE ***
00648            ELSEIF ( DOSLSH.NE.0 ) THEN
00649 C          *** LINE ***
00650              TYPE(I) = 'LINE'
00651 C            PRINT*,' ITS A LINE'
00652              IF ( DOSLSH.EQ.1 ) THEN
00653 C            *** LEFT DEFAULT VECTOR ***
00654                REARG (I,1) = 0
00655                REARG (I,2) = 0
00656                REARG (I,3) = 0
00657 C            *** LEFT DEFAULT VECTOR ***
00658              ELSEIF ( SLASH1.EQ.DOSLSH ) THEN
00659                IF ( WORD(I)(:1).EQ.'1'.OR.WORD(I)(:1).EQ.'2'.OR.
00660     +               WORD(I)(:1).EQ.'3'.OR.WORD(I)(:1).EQ.'4'.OR.
00661     +               WORD(I)(:1).EQ.'5'.OR.WORD(I)(:1).EQ.'6'.OR.
00662     +               WORD(I)(:1).EQ.'7'.OR.WORD(I)(:1).EQ.'8'.OR.
00663     +               WORD(I)(:1).EQ.'9'.OR.WORD(I)(:1).EQ.'0' ) THEN
00664 C              *** INDEX DEFINED LEFT VECTOR ***
 0665                  W = WORD(I)(:DOSLSH-1)
00666                  READ (W,'(BN,I10)') INTARG(I,1)
00667 C                PRINT*,' LEFT INDEX = ',INTARG(I,1)
00668 C              *** INDEX DEFINED LEFT VECTOR ***
00669                ELSE
00670 C              *** TEXT DEFINED LEFT VECTOR ***
00671                  W = WORD(I)(:DOSLSH-1)
00672                  CALL ATIND ( W,INTARG(I,1),*100 )
00673 C              *** TEXT DEFINED LEFT VECTOR ***
00674                ENDIF
00675                REARG(I,1) = X(INTARG(I,1))
00676                REARG(I,2) = Y(INTARG(I,1))
00677                REARG(I,3) = Z(INTARG(I,1))
00678                TXTARG(I,1) = TEXT(INTARG(I,1))
00679              ELSE
00680 C            *** NUMERICAL DEFINED LEFT VECTOR ***
00681                W = WORD(I)(:SLASH1-1)
00682                IF( INDEX(W,'.').EQ.0 ) THEN
00683 C                PRINT*,' INTEGER , W= *',W,'*'
00684                  READ (W,'(BN,I10)') INTERM
 0685                  REARG(I,1) = INTERM
00686                ELSE
00687                  READ (W,'(F15.14)') REARG(I,1)
```

```
00688                ENDIF
00689                WORD(I)(SLASH1:SLASH1) = ' '
00690                SLASH2 = INDEX(WORD(I),'/')
00691                W = WORD(I)(SLASH1+1:SLASH2-1)
00692                IF( INDEX(W,'.').EQ.0 ) THEN
00693                   READ (W,'(BN,I10)') INTERM
00694                   REARG(I,2) = INTERM
00695                ELSE
00696                   READ (W,'(F15.14)') REARG(I,2)
00697                ENDIF
00698                W = WORD(I)(SLASH2+1:DOSLSH-1)
00699                IF( INDEX(W,'.').EQ.0 ) THEN
00700                   READ (W,'(BN,I10)') INTERM
00701                   REARG(I,3) = INTERM
00702                ELSE
00703                   READ (W,'(F15.14)') REARG(I,3)
00704                ENDIF
 0705 C            *** NUMERICAL DEFINED LEFT VECTOR ***
00706             ENDIF
00707             IX = DOSLSH+2
00708             W = WORD(I)(IX:)
00709             IF ( WORD(I)(IX:IX).EQ.' ' ) THEN
00710 C           *** RIGHT DEFAULT VECTOR ***
00711                REARG (I,4) = 0
00712                REARG (I,5) = 0
00713                REARG (I,6) = 0
00714 C           *** RIGHT DEFAULT VECTOR ***
00715             ELSEIF ( INDEX(WORD(I)(IX:),'/').EQ.0 ) THEN
00716                IF ( W(:1).EQ.'1'.OR.W(:1).EQ.'2'.OR.
00717      +              W(:1).EQ.'3'.OR.W(:1).EQ.'4'.OR.
00718      +              W(:1).EQ.'5'.OR.W(:1).EQ.'6'.OR.
00719      +              W(:1).EQ.'7'.OR.W(:1).EQ.'8'.OR.
00720      +              W(:1).EQ.'9'.OR.W(:1).EQ.'0' ) THEN
 0721 C             *** INDEX DEFINED RIGHT VECTOR ***
00722                   READ (W,'(BN,I10)') INTARG(I,2)
00723 C             *** INDEX DEFINED RIGTH VECTOR ***
00724                ELSE
00725 C             *** TEXT DEFINED RIGHT VECTOR ***
00726                   CALL ATIND ( W,INTARG(I,2),*100 )
00727 C             *** TEXT DEFINED RIGHT VECTOR ***
00728                ENDIF
00729                REARG(I,4) = X(INTARG(I,2))
00730                REARG(I,5) = Y(INTARG(I,2))
00731                REARG(I,6) = Z(INTARG(I,2))
00732                TXTARG(I,2) = TEXT(INTARG(I,2))
00733             ELSE
00734 C              *** NUMERICAL DEFINED RIGHT VECTOR ***
00735                SLASH1 = INDEX(W,'/')
00736                W(SLASH1:SLASH1) = ' '
00737                SLASH2 = INDEX(W,'/')
00738                W2 = W(:SLASH1-1)
00739                IF( INDEX(W2,'.').EQ.0 ) THEN
00740                   READ (W2,'(BN,I10)') INTERM
 0741                   REARG(I,4) = INTERM
00742                ELSE
00743                   READ (W2,'(F15.14)') REARG(I,4)
00744                ENDIF
00745                W2 = W(SLASH1+1:SLASH2-1)
00746                IF( INDEX(W2,'.').EQ.0 ) THEN
```

```
00747                READ (W2,'(BN,I10)') INTERM
00748                REARG(I,5) = INTERM
00749              ELSE
00750                READ (W2,'(F15.14)') REARG(I,5)
00751              ENDIF
00752              W2 = W(SLASH2+1:)
00753              IF( INDEX(W2,'.').EQ.0 ) THEN
00754                READ (W2,'(BN,I10)') INTERM
00755                REARG(I,6) = INTERM
00756              ELSE
00757                READ (W2,'(F15.14)') REARG(I,6)
00758              ENDIF
00759 C            *** NUMERICAL DEFINED RIGHT VECTOR ***
00760            ENDIF
 0761          ELSEIF ( SLASH1.NE.0 ) THEN
00762            TYPE(I) = 'VECTOR'
00763            W = WORD(I)(:SLASH1-1)
00764            IF( INDEX(W,'.').EQ.0 ) THEN
00765              READ (W,'(BN,I10)') INTERM
00766              REARG(I,1) = INTERM
00767            ELSE
00768              READ (W,'(F15.14)') REARG(I,1)
00769            ENDIF
00770            WORD(I)(SLASH1:SLASH1) = ' '
00771            SLASH2 = INDEX(WORD(I),'/')
00772            W = WORD(I)(SLASH1+1:SLASH2-1)
00773            IF( INDEX(W,'.').EQ.0 ) THEN
00774              READ (W,'(BN,I10)') INTERM
00775              REARG(I,2) = INTERM
00776            ELSE
00777              READ (W,'(F15.14)') REARG(I,2)
00778            ENDIF
00779            W = WORD(I)(SLASH2+1:)
00780            IF( INDEX(W,'.').EQ.0 ) THEN
 0781              READ (W,'(BN,I10)') INTERM
00782              REARG(I,3) = INTERM
00783            ELSE
00784              READ (W,'(F15.14)') REARG(I,3)
00785            ENDIF
00786          ELSEIF (INDEX(WORD(I),'.').NE.0 ) THEN
00787            TYPE(I) = 'REAL'
00788            READ (WORD(I),'(F15.14)') REARG(I,1)
00789          ELSEIF ( WORD(I)(:1).EQ.'1'.OR.WORD(I)(:1).EQ.'2'.OR.
00790     +             WORD(I)(:1).EQ.'3'.OR.WORD(I)(:1).EQ.'4'.OR.
00791     +             WORD(I)(:1).EQ.'5'.OR.WORD(I)(:1).EQ.'6'.OR.
00792     +             WORD(I)(:1).EQ.'7'.OR.WORD(I)(:1).EQ.'8'.OR.
00793     +             WORD(I)(:1).EQ.'9'.OR.WORD(I)(:1).EQ.'0'.OR.
00794     +             WORD(I)(:1).EQ.'-' ) THEN
00795            TYPE (I) = 'INTEGER'
00796            READ(WORD(I),'(BN,I15)') INTARG(I,1)
00797            REARG(I,1) = INTARG(I,1)
00798          ELSE
00799            TYPE (I) = 'TEXT'
00800            TXTARG(I,1) = WORD(I)
 0801          ENDIF
00802   850 CONTINUE
00803 C
00804 C     *** CLASSIFY THE ARGUMENTS ***
```

```
00805 C
00806       RETURN
00807   100 CONTINUE
00808       RETURN 1
00809       END
00810 C
00811 C     ********************
00812 C
00813       SUBROUTINE ISTAR(*)
00814 C
00815 C     ********************
00816 C
00817 C     *** INSERT ATOMS INTO THE MOLECULE ***
00818 C
00819 C     ******************** INPROC - DECLARATIONS ***********************
00820       INTEGER ARGMAX
 0821       PARAMETER ( ARGMAX=10 )
00822       CHARACTER LINE*60
00823       INTEGER INTARG(ARGMAX,2)
00824       CHARACTER*15 TXTARG(ARGMAX,2)
00825       REAL REARG(ARGMAX,6)
00826       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
00827       LOGICAL CHEK
00828 C     ******************** INPROC - DECLARATIONS ***********************
00829 C
00830       PARAMETER ( NATOMS=100 )
00831       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
00832       CHARACTER*5 TEXT(NATOMS)
00833       COMMON /B/ TEXT
00834       CHARACTER*4 MODE
00835       REAL XX,YY,ZZ
00836       INTEGER POS,NUM
00837 C
00838 C     ******************** INPROC - PROCEDURE **********************
00839       WRITE(*,*) ' ENTER  TEXT //CART/POL/CYL X/Y/Z :'
00840   300 CONTINUE
 0841       READ(*,'(A)',END=100) LINE
00842       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00843       IF (NARGS.EQ.1.AND.TYPE(1).EQ.'TEXT'.AND.TXTARG(1,1).EQ.'END')
00844      + GOTO 400
00845       NARCHK = 2
00846       TYPCHK(1) = 'TEXT'
00847       TYPCHK(2) = 'VECTOR'
00848       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00849       IF (CHEK) THEN
00850          MODE = 'CART'
00851          POS = 2
00852       ELSE
00853          NARCHK = 3
00854          TYPCHK(1) = 'TEXT'
00855          TYPCHK(2) = 'TEXT'
00856          TYPCHK(3) = 'VECTOR'
00857          CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
00858          MODE = TXTARG(2,1)
00859          POS = 3
00860       ENDIF
 0861 C     ******************** INPROC - PROCEDURE **********************
00862 C
```

```
00863       NUM = NUMBER + 1
00864       TEXT(NUM) = TXTARG(1,1)
00865       IF (MODE(:4).EQ.'CART') THEN
00866          X(NUM) = REARG(POS,1)
00867          Y(NUM) = REARG(POS,2)
00868          Z(NUM) = REARG(POS,3)
00869       ELSEIF (MODE(:3).EQ.'POL') THEN
00870          CALL POLAR (XX,YY,ZZ,REARG(POS,1),REARG(POS,2),REARG(POS,3))
00871          X(NUM) = XX
00872          Y(NUM) = YY
00873          Z(NUM) = ZZ
00874       ELSEIF (MODE(:3).EQ.'CYL') THEN
00875          ANGRAD = REARG(POS,2)*3.141592654/180.0
00876          X(NUM) = REARG(POS,1)*COS(ANGRAD)
00877          Y(NUM) = REARG(POS,1)*SIN(ANGRAD)
00878          Z(NUM) = REARG(POS,3)
00879       ELSE
00880          WRITE(*,*) ' MODE NOT EQUAL CART, POL OR CYL'
 0881          GOTO 100
00882       ENDIF
00883 C
00884       NUMBER = NUM
00885       GOTO 300
00886   400 CONTINUE
00887       RETURN
00888 C
00889   100 CONTINUE
00890       RETURN 1
00891       END
00892 C
00893 C     ******************
00894 C
00895       SUBROUTINE DEL(*)
00896 C
00897 C     ******************
00898 C
00899 C     *** REMOVE ONE OR MORE ATOMS FROM THE MOLECULE ***
00900 C
 0901 C     ******************* INPROC - DECLARATIONS ************************
00902       INTEGER ARGMAX
00903       PARAMETER ( ARGMAX=10 )
00904       CHARACTER LINE*60
00905       INTEGER INTARG(ARGMAX,2)
00906       CHARACTER*15 TXTARG(ARGMAX,2)
00907       REAL REARG(ARGMAX,6)
00908       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
00909       LOGICAL CHEK
00910 C     ******************* INPROC - DECLARATIONS ************************
00911 C
00912       PARAMETER ( NATOMS=100 )
00913       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
00914       CHARACTER*5 TEXT(NATOMS)
00915       COMMON /B/ TEXT
00916 C
00917       INTEGER FIRST,LAST,SHIFT
00918 C
00919 C     ******************* INPROC - PROCEDURE ***************************
00920       WRITE(*,*) ' ENTER  RANGE :'
```

```
 0921       READ(*,'(A)',END=100) LINE
00922       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00923 C     PRINT*,' TYPE(1) = ',TYPE(1)
00924       NARCHK = 1
00925       TYPCHK(1) = 'TEXT'
00926       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00927       IF (CHEK) THEN
00928          CALL ATIND (TXTARG(1,1),FIRST,*100)
00929          LAST = FIRST
00930 C        PRINT*,' IT''S A TEXT'
00931       ELSE
00932          TYPCHK(1) = 'RANGE'
00933          CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00934          IF (CHEK) THEN
00935             FIRST = INTARG(1,1)
00936 C           PRINT*,' IT''S A RANGE'
00937             LAST = INTARG(1,2)
00938 C           PRINT*,' FIRST = ',FIRST,' LAST = ',LAST
00939          ELSE
00940             TYPCHK(1) = 'INTEGER'
 0941             CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
00942             FIRST = INTARG(1,1)
00943 C           PRINT*,' IT''S A INTEGER'
00944             LAST = FIRST
00945          ENDIF
00946       ENDIF
00947 C     ******************** INPROC - PROCEDURE ************************
00948       NUMBER = NUMBER-LAST+FIRST-1
00949       DO 200,I=FIRST,NUMBER
00950          SHIFT = I+LAST-FIRST+1
00951          X(I) = X(SHIFT)
00952          Y(I) = Y(SHIFT)
00953          Z(I) = Z(SHIFT)
00954          TEXT(I) = TEXT(SHIFT)
00955   200 CONTINUE
00956 C
00957       RETURN
00958 C
00959   100 CONTINUE
00960       RETURN 1
 0961       END
00962 C
00963 C     ********************
00964 C
00965       SUBROUTINE READ(*)
00966 C
00967 C     ********************
00968 C
00969 C     *** INSERT ATOMS FROM A FILE ***
00970 C
00971 C     ******************** INPROC - DECLARATIONS ************************
00972       INTEGER ARGMAX
00973       PARAMETER ( ARGMAX=10 )
00974       CHARACTER LINE*60
00975       INTEGER INTARG(ARGMAX,2)
00976       CHARACTER*15 TXTARG(ARGMAX,2)
00977       REAL REARG(ARGMAX,6)
00978       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
00979       LOGICAL CHEK
```

```
00980 C     ****************** INPROC - DECLARATIONS **********************
 0981       PARAMETER ( NATOMS=100 )
00982       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
00983       CHARACTER*5 TEXT(NATOMS)
00984       COMMON /B/ TEXT
00985       INTEGER NLINE1
00986       LOGICAL OPEN
00987       CHARACTER*6 FILNAM,FMT*20
00988 C     ****************** INPROC - PROCEDURE **********************
00989       WRITE (*,*) ' ENTER : FILENAME  (STARTING LINE (FORMAT)) :'
00990       READ(*,'(A)',ERR=100) LINE
00991       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
00992       NARCHK = 3
00993       TYPCHK(1) = 'TEXT'
00994       TYPCHK(2) = 'INTEGER'
00995       TYPCHK(3) = 'TEXT'
00996       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
00997       IF (.NOT.CHEK) THEN
00998          NARCHK = 2
00999          CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
01000          TXTARG(3,1) = 'FREE'
 1001          IF (.NOT.CHEK) THEN
01002             NARCHK = 1
01003             CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
01004             INTARG(2,1) = 1
01005          ENDIF
01006       ENDIF
01007       FILNAM = TXTARG(1,1)
01008       NLINE1 = INTARG(2,1)
01009       FMT = '('//TXTARG(3,1)//')'
01010 C     WRITE(*,'(3A,I10,3A)') '*',FILNAM,'*',NLINE1,'*',FMT,'*'
01011 C     ****************** INPROC - PROCEDURE **********************
01012       CLOSE (3)
01013       INQUIRE (FILE=FILNAM,OPENED=OPEN)
01014       IF ( OPEN ) CLOSE (4)
01015       OPEN (3,FILE=FILNAM,STATUS='OLD',ERR=100)
01016       DO 200,I=2,NLINE1
01017          READ(3,'()')
01018   200 CONTINUE
01019       IF (TXTARG(3,1).NE.'FREE') THEN
01020          READ(3,FMT,END=110,ERR=100)
 1021      +    (TEXT(N),X(N),Y(N),Z(N),N=NUMBER+1,NATOMS)
01022       ELSE
01023          N = NUMBER
01024          NARCHK = 4
01025          TYPCHK(1) = 'TEXT'
01026          TYPCHK(2) = 'REAL'
01027          TYPCHK(3) = 'REAL'
01028          TYPCHK(4) = 'REAL'
01029   300    CONTINUE
01030          N = N+1
01031          READ(3,'(A)',ERR=100,END=110) LINE
01032          CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*400)
01033          CALL CHECK1(NARGS,NARCHK,TYPE,TYPCHK,*400)
01034          TEXT(N) = TXTARG(1,1)
01035          X(N) = REARG(2,1)
01036          Y(N) = REARG(3,1)
01037          Z(N) = REARG(4,1)
```

```
01038          GOTO 300
01039        ENDIF
01040        WRITE(*,*) ' *** ERROR: TOO MANY ATOMS IN FILE ',FILNAM,'. ***'
 1041        GOTO 100
01042   400 CONTINUE
01043        WRITE(*,*) ' *** ERROR DURING READING FREE FORMAT ***'
01044        GOTO 100
01045   110 CONTINUE
01046        NUMBER = N-1
01047        CLOSE (3)
01048        RETURN
01049 C
01050   100 CONTINUE
01051        RETURN 1
01052        END
01053 C
01054 C      ********************
01055 C
01056        SUBROUTINE WRITE(*)
01057 C
01058 C      ********************
01059 C
01060 C      *** WRITE THE MOLECULE TO A FILE ***
 1062        PARAMETER ( NATOMS=100 )
01063        COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01064        CHARACTER*5 TEXT(NATOMS)
01065        COMMON /B/ TEXT
01066 C
01067        CHARACTER*6 FILNAM
01068        LOGICAL OPEN
01069        INTEGER CHARGE(NATOMS)
01070 C
01071        WRITE (*,*) ' ENTER FILENAME:'
01072        READ(*,'(A)',ERR=100) FILNAM
01073        INQUIRE (FILE=FILNAM,OPENED=OPEN)
01074        IF ( .NOT. OPEN ) OPEN (4,FILE=FILNAM,STATUS='NEW',ERR=100)
01075 C
01076        DO 200,I=1,NUMBER
01077           IF ( TEXT(I)(:1).EQ.'H' ) THEN
01078              CHARGE(I) = 1
01079           ELSEIF ( TEXT(I)(:1).EQ.'C' ) THEN
01080              CHARGE(I) = 6
 1081           ELSEIF ( TEXT(I)(:1).EQ.'N' ) THEN
01082              CHARGE(I) = 7
01083           ELSEIF ( TEXT(I)(:1).EQ.'O' ) THEN
01084              CHARGE(I) = 8
01085           ELSEIF ( TEXT(I)(:1).EQ.'S' ) THEN
01086              CHARGE(I) = 16
01087           ELSE
01088              CHARGE(I) = 999
01089           ENDIF
01090   200 CONTINUE
01091 C
01092        WRITE(4,'(A4,3F15.8,I3,I3)')
01093       + (TEXT(N),X(N),Y(N),Z(N),CHARGE(N),1,N=1,NUMBER)
01094   110 CONTINUE
01095        NUMBER = N-1
01096        RETURN
01097 C
```

```
01098   100 CONTINUE
01099       RETURN 1
01100       END
 1101 C
01102 C     ********************
01103 C
01104       SUBROUTINE DISP
01105 C
01106 C     ********************
01107 C
01108 C     *** DISPLAY THE COORDINATES ***
01109 C
01110       PARAMETER ( NATOMS=100 )
01111       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01112       CHARACTER*5 TEXT(NATOMS)
01113       COMMON /B/ TEXT
01114 C
01115       IF ( NUMBER.EQ.0 ) THEN
01116          WRITE (*,*) ' NO ATOMS IN THE MOLECULE.'
01117       ELSE
01118          IHALF = NUMBER/2
01119          WRITE (*,100) (N,TEXT(N),X(N),Y(N),Z(N),N+IHALF,TEXT(N+IHALF),
01120      +                  X(N+IHALF),Y(N+IHALF),Z(N+IHALF),N=1,IHALF)
 1121   100    FORMAT (/2(' N  TEXT',TR8,'X',TR8,'Y',TR8,'Z    ')
01122      +          //2(I4,TR1,A6,3F9.5))
01123          IF ( MOD(NUMBER,2).NE.0 ) THEN
01124             WRITE(*,110) NUMBER,TEXT(NUMBER),X(NUMBER),Y(NUMBER),
01125      +                   Z(NUMBER)
01126   110       FORMAT (T39,I4,TR1,A6,3F9.5)
01127          ENDIF
01128       ENDIF
01129 C
01130       RETURN
01131       END
01132 C
01133 C     ********************
01134 C
01135       SUBROUTINE DIST(*)
01136 C
01137 C     ********************
01138 C
01139 C     *** DISPLAY THE DISTANCE BETWEEN TWO ATOMS ***
01140 C
 1141       PARAMETER ( NATOMS=100 )
01142       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01143       CHARACTER*5 TEXT(NATOMS)
01144       COMMON /B/ TEXT
01145 C
01146 C     ******************* INPROC - DECLARATIONS ***********************
01147       INTEGER ARGMAX
01148       PARAMETER ( ARGMAX=10 )
01149       CHARACTER LINE*60
01150       INTEGER INTARG(ARGMAX,2)
01151       CHARACTER*15 TXTARG(ARGMAX,2)
01152       REAL REARG(ARGMAX,6)
01153       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01154 C     ******************* INPROC - DECLARATIONS ***********************
01155 C
```

```
01156 C     ******************* INPROC - PROCEDURE ***********************
01157       WRITE(*,*) ' ENTER :  LINE'
01158       READ(*,'(A)',END=100) LINE
01159       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01160       NARCHK = 1
 1161       TYPCHK(1) = 'LINE'
01162       CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
01163       XX1 = REARG(1,1)
01164       YY1 = REARG(1,2)
01165       ZZ1 = REARG(1,3)
01166       XX2 = REARG(1,4)
01167       YY2 = REARG(1,5)
01168       ZZ2 = REARG(1,6)
01169 C     ******************* INPROC - PROCEDURE ***********************
01170       DIST1 = SQRT ((XX1-XX2)**2+(YY1-YY2)**2+(ZZ1-ZZ2)**2)
01171       DIST2 = SQRT (XX1**2+YY1**2+ZZ1**2)
01172       DIST3 = SQRT (XX2**2+YY2**2+ZZ2**2)
01173       IF ( DIST2.GT.DIST3 ) DIST1 = -DIST1
01174       WRITE (*,'(7(A,F8.4))') ' DIST((',XX1,'/',YY1,'/',ZZ1,'),(',
01175      +     XX2,'/',YY2,'/',ZZ2,')) = ',DIST1
01176       RETURN
01177 C
01178   100 CONTINUE
01179       RETURN 1
01180       END
 1182 C     ********************
01183 C
01184       SUBROUTINE ANG(*)
01185 C
01186 C     ********************
01187 C
01188 C     *** DISPLAY THE ANGLE OF THREE ATOMS ***
01189 C
01190       PARAMETER ( NATOMS=100 )
01191       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01192       CHARACTER*5 TEXT(NATOMS)
01193       COMMON /B/ TEXT
01194 C
01195       INTEGER N,I
01196       REAL XX(3),YY(3),ZZ(3),X1,Y1,Z1,X2,Y2,Z2
01197 C     ******************* INPROC - DECLARATIONS ***********************
01198       INTEGER ARGMAX
01199       PARAMETER ( ARGMAX=10 )
01200       CHARACTER LINE*60
 1201       INTEGER INTARG(ARGMAX,2)
01202       CHARACTER*15 TXTARG(ARGMAX,2)
01203       REAL REARG(ARGMAX,6)
01204       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01205 C     ******************* INPROC - DECLARATIONS ***********************
01206 C
01207 C     ******************* INPROC - PROCEDURE ***********************
01208       WRITE(*,*) ' ENTER  VECTOR1 VECTOR2 VECTOR3 '
01209       READ(*,'(A)',END=100) LINE
01210       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01211       IF (NARGS.NE.3) THEN
01212          WRITE(*,*) ' WRONG NUMBER OF ARGUMENTS '
01213          GOTO 100
01214       ENDIF
```

```
01215       DO 300,I=1,3
01216          IF (TYPE(I).EQ.'TEXT') THEN
01217             CALL ATIND(TXTARG(I,1),N,*100)
01218             XX(I) = X(N)
01219             YY(I) = Y(N)
01220             ZZ(I) = Z(N)
 1221 C           PRINT*,N
01222          ELSEIF (TYPE(I).EQ.'INTEGER') THEN
01223             N = INTARG(I,1)
01224             XX(I) = X(N)
01225             YY(I) = Y(N)
01226             ZZ(I) = Z(N)
01227 C           PRINT*,N
01228          ELSEIF (TYPE(I).EQ.'VECTOR') THEN
01229             XX(I) = REARG(I,1)
01230             YY(I) = REARG(I,2)
01231             ZZ(I) = REARG(I,3)
01232          ELSE
01233             WRITE(*,'(A,I3,2A)') ' WRONG TYPE : ARGUMENT',I,'=',TYPE(I)
01234             GOTO 100
01235          ENDIF
01236   300 CONTINUE
01237 C     ****************** INPROC - PROCEDURE **********************
01238       X1 = XX(1)-XX(2)
01239       Y1 = YY(1)-YY(2)
01240       Z1 = ZZ(1)-ZZ(2)
 1241       X2 = XX(3)-XX(2)
01242       Y2 = YY(3)-YY(2)
01243       Z2 = ZZ(3)-ZZ(2)
01244       CALL ANGLE (ALP,X1,Y1,Z1,X2,Y2,Z2,*100)
01245       WRITE (*,'(A,9(F5.2,A),F7.2)')
01246      +' ANGLE((',XX(1),'/',YY(1),'/',ZZ(1),')-(',
01247      +          XX(2),'/',YY(2),'/',ZZ(2),')-(',
01248      +          XX(3),'/',YY(3),'/',ZZ(3),')) = ',ALP
01249       RETURN
01250 C
01251   100 CONTINUE
01252       RETURN 1
01253       END
01254 C
01255 C     ********************
01256 C
01257       SUBROUTINE DIANG(*)
01258 C
01259 C     ********************
01260 C
 1261 C     *** DISPLAY THE DIHEDRAL ANGLE OF FOUR ATOMS ***
01262 C
01263 C     ****************** INPROC - DECLARATIONS **********************
01264       INTEGER ARGMAX
01265       PARAMETER ( ARGMAX=10 )
01266       CHARACTER LINE*60
01267       INTEGER INTARG(ARGMAX,2)
01268       CHARACTER*15 TXTARG(ARGMAX,2)
01269       REAL REARG(ARGMAX,6)
01270       CHARACTER*7 TYPE(ARGMAX),TYPCH((ARGMAX)
01271 C     ****************** INPROC - DECLARATIONS **********************
01272       PARAMETER ( NATOMS=100 )
```

```
      COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
      CHARACTER*5 TEXT(NATOMS)
      COMMON /B/ TEXT
C     ******************** INPROC - PROCEDURE ***********************
      WRITE(*,*) ' ENTER  TWO LINES :'
      READ(*,'(A)',END=100) LINE
      CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
      NARCHK = 2
      TYPCHK(1) = 'LINE'
      TYPCHK(2) = 'LINE'
      CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,CHEK,*100)
      XN1 = REARG(1,1)
      YN1 = REARG(1,2)
      ZN1 = REARG(1,3)
      XN2 = REARG(1,4)
      YN2 = REARG(1,5)
      ZN2 = REARG(1,6)
      XN3 = REARG(2,1)
      YN3 = REARG(2,2)
      ZN3 = REARG(2,3)
      XN4 = REARG(2,4)
      YN4 = REARG(2,5)
      ZN4 = REARG(2,6)
C     ******************** INPROC - PROCEDURE ***********************
C
      X1 = XN1-XN2
      Y1 = YN1-YN2
      Z1 = ZN1-ZN2
      X2 = XN2-XN3
      Y2 = YN2-YN3
      Z2 = ZN2-ZN3
      X3 = XN4-XN3
      Y3 = YN4-YN3
      Z3 = ZN4-ZN3
      X4 = -X2
      Y4 = -Y2
      Z4 = -Z2
      CALL CROSS (X5,Y5,Z5,X4,Y4,Z4,X1,Y1,Z1)
      CALL CROSS (X6,Y6,Z6,X3,Y3,Z3,X2,Y2,Z2)
      CALL CROSS (X7,Y7,Z7,X5,Y5,Z5,X6,Y6,Z6)
      PROD = X7*X4+Y7*Y4+Z7*Z4
      CALL ANGLE (BETA,X5,Y5,Z5,X6,Y6,Z6,*100)
      BETA = SIGN (BETA,PROD)
      WRITE (*,'(A,F8.3)') ' DIANG = ',BETA
      RETURN
C
  100 CONTINUE
      RETURN 1
      END
C
C     ********************
C
      SUBROUTINE TABLE(*)
C
C     ********************
C
C     *** DISPLAY TABLES OF THE DISTANCES AND ANGLES ***
C
```

```
01331       INTEGER PPLINE
01332       PARAMETER ( PPLINE=14 )
01333       PARAMETER ( NATOMS=100 )
01334       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01335       CHARACTER*5 TEXT(NATOMS)
01336       COMMON /B/ TEXT
01337 C
01338       CHARACTER*5 LINE(PPLINE)
01339       CHARACTER*10 LINE1(7),LINE2(7),LINE3(7)
01340 C
 1341       WRITE (*,*) ' ENTER MAXIMAL DISTANCE :'
01342       READ(*,*,ERR=100,END=100) DISTM
01343 C
01344       WRITE(*,'(A,F5.2,A/)') ' INTERATOMIC DISTANCES (= ',DISTM,' :'
01345       DO 250,K=1,NUMBER-1,PPLINE
01346          IANF1 = MAX(2,K)
01347          IEND1 = MIN(K+PPLINE-1,NUMBER)
01348          WRITE (*,'(T10,100(A5))') (TEXT(I),I=IANF1,IEND1)
01349          WRITE (*,'(T7,100(I5))') (I,I=IANF1,IEND1)
01350          WRITE (*,'(T2,78(''-''))')
01351          IEND2 = MIN(K+PPLINE-2,NUMBER-1)
01352          DO 200,I=1,IEND2
01353             IPLACE = 0
01354             DO 800,J=1,14
01355                LINE(J) = ' '
01356   800       CONTINUE
01357             IANF2 = MAX(I+1,K)
01358             JJ = MOD(IANF2,PPLINE)
01359             IF (JJ.EQ.0) JJ = PPLINE
01360             IF (K.EQ.1) JJ = JJ-1
 1361             DO 300,J=IANF2,IEND1
01362                DIST = SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2+(Z(I)-Z(J))**2)
01363                IF (DIST.LE.DISTM) THEN
01364                   WRITE (LINE(JJ),'(F5.2)') DIST
01365                   IPLACE = IPLACE+1
01366                ENDIF
01367                JJ = JJ+1
01368   300       CONTINUE
01369             IF (IPLACE.NE.0) WRITE (*,'(I3,T5,A4,T9,14(A))')
01370      +                       I,TEXT(I),LINE
01371   200    CONTINUE
01372          WRITE (*,'(T2,78(''-''))')
01373   250 CONTINUE
01374 C
01375       WRITE (*,'(A)') ' ANGLES BETWEEN NEIGHBOUR ATOMS :'
01376       DO 400,I=1,NUMBER
01377          IPLACE = 0
01378          DO 900,J=1,7
01379            LINE1(J) = ' '
01380            LINE2(J) = ' '
 1381            LINE3(J) = ' '
01382   900    CONTINUE
01383          DO 500,J=1,NUMBER
01384            IF (J.NE.I) THEN
01385               DO 600,K=J+1,NUMBER
01386                  IF (K.NE.I) THEN
01387                    X1 = X(J)-X(I)
01388                    Y1 = Y(J)-Y(I)
```

```
01389                  Z1 = Z(J)-Z(I)
01390                  X2 = X(K)-X(I)
01391                  Y2 = Y(K)-Y(I)
01392                  Z2 = Z(K)-Z(I)
01393                  DIST1 = SQRT (X1**2+Y1**2+Z1**2)
01394                  DIST2 = SQRT (X2**2+Y2**2+Z2**2)
01395                  IF ( DIST1.LE.DISTM.AND.DIST2.LE.DISTM ) THEN
01396                     IPLACE = IPLACE+1
01397                     IF ( IPLACE.GT.6 ) THEN
01398 C                      PRINT*,' XXX '
01399                        WRITE (*,'(/T10,7(A,TR1))') LINE1
01400                        WRITE (*,'(I3,T5,A4,TR1,7(A,TR1))')
 1401     +                         I,TEXT(I),LINE2
01402                        WRITE (*,'(T10,7(A,TR1))') LINE3
01403                        DO 950,L=1,7
01404                           LINE1(L) = ' '
01405                           LINE2(L) = ' '
01406                           LINE3(L) = ' '
01407   950                  CONTINUE
01408                        IPLACE = 1
01409                     ENDIF
01410                     WRITE ( LINE1(IPLACE),'(I3,2A)' ) J,'.',TEXT(J)
01411                     CALL ANGLE (ANG,X1,Y1,Z1,X2,Y2,Z2,*100)
01412                     WRITE ( LINE2(IPLACE),'(F5.1)' ) ANG
01413                     WRITE ( LINE3(IPLACE),'(I3,2A)' ) K,'.',TEXT(K)
01414                  ENDIF
01415               ENDIF
01416   600      CONTINUE
01417         ENDIF
01418   500   CONTINUE
01419         IF ( IPLACE.NE.0 ) THEN
01420           WRITE (*,'(/T10,7(A,TR1))') LINE1
 1421           WRITE (*,'(I3,T5,A4,TR1,7(A,TR1))') I,TEXT(I),LINE2
01422           WRITE (*,'(T10,7(A,TR1))') LINE3
01423         ENDIF
01424   400 CONTINUE
01425       WRITE (*,'(T2,78(''-''))')
01426 C
01427       RETURN
01428 C
01429   100 CONTINUE
01430       RETURN 1
01431       END
01432 C
01433 C     ****************************************
01434 C
01435       SUBROUTINE ANGLE (ANG,X1,Y1,Z1,X2,Y2,Z2,*)
01436 C
01437 C     ****************************************
01438 C
01439 C     *** CALCULATE THE ANGLE BETWEEN TWO VECTORS ***
01440 C
 1441       PARAMETER ( PI = 3.141592654 )
01442 C
01443       BET1 = SQRT(X1**2+Y1**2+Z1**2)
01444       BET2 = SQRT(X2**2+Y2**2+Z2**2)
01445       IF ( BET1.EQ.0.0.OR.BET2.EQ.0.0 ) THEN
01446          WRITE(*,'(A)') ' ERROR: ANGLE IS UNDEFINED, DISTANCE IS ZERO.'
```

```
01447          RETURN 1
01448        ENDIF
01449        COSALP = (X1*X2+Y1*Y2+Z1*Z2)/(BET1*BET2)
01450        ANG = 180.0/PI*ACOS(COSALP)
01451 C
01452        RETURN
01453        END
01454 C
01455 C      ************************************
01456 C
01457        SUBROUTINE CROSS (X,Y,Z,A,B,C,D,E,F)
01458 C
01459 C      ************************************
01460 C
 1461 C      *** CALCULATE THE CROSS PRODUCT OF TWO VECTORS ***
01462 C
01463        X = B*F-C*E
01464        Y = C*D-A*F
01465        Z = A*E-B*D
01466 C
01467        RETURN
01468        END
01469 C
01470 C      ****************************************
01471 C
01472        SUBROUTINE ONE (X1,Y1,Z1,X2,Y2,Z2,*)
01473 C
01474 C      ****************************************
01475 C
01476 C      *** CALCULATE THE UNIT VECTOR OF A VECTOR ***
01477 C
01478        BET = SQRT(X2**2+Y2**2+Z2**2)
01479        IF ( BET.EQ.0.0) THEN
01480           WRITE(*,*) ' ERROR :  UNITVECTOR OF 0/0/0.'
 1481           RETURN 1
01482        ENDIF
01483        X1= X2/BET
01484        Y1= Y2/BET
01485        Z1= Z2/BET
01486 C
01487        RETURN
01488        END
01489 C
01490 C      ******************************
01491 C
01492        SUBROUTINE ATIND (TEXT1,NUM,*)
01493 C
01494 C      ******************************
01495 C
01496 C      *** FIND THE LOCATION OF AN ATOM WITH A SPECIFIED TEXT ***
01497 C
01498        PARAMETER ( NATOMS=100 )
01499        COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01500        CHARACTER*5 TEXT(NATOMS)
 1501        COMMON /B/ TEXT
01502 C
01503        CHARACTER TEXT1*(*)
01504 C
```

```
01505       DO 200,I=1,NUMBER
01506       IF ( TEXT1.EQ.TEXT(I) ) THEN
01507          NUM = I
01508          RETURN
01509       ENDIF
01510   200 CONTINUE
01511       WRITE (*,*) '*** THE REFERENCED ATOM ',TEXT1,' DOES NOT EXIST ***'
01512       RETURN 1
01513       END
01514 C
01515 C     ******************
01516 C
01517       SUBROUTINE ATOM(*)
01518 C
01519 C     ******************
01520 C
 1521 C     *** INSERT A ATOM WHOSE LOCATION IS GIVEN RELATIVE TO OTHERS ***
01522 C
01523       PARAMETER ( NATOMS=100 )
01524       PARAMETER ( PI = 3.141592654 )
01525       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01526       CHARACTER*5 TEXT(NATOMS)
01527       COMMON /B/ TEXT
01528 C
01529       CHARACTER*5 TEXT1
01530 C
01531 C     ****************** INPROC - DECLARATIONS **********************
01532       INTEGER ARGMAX
01533       PARAMETER ( ARGMAX=10 )
01534       CHARACTER LINE*60
01535       INTEGER INTARG(ARGMAX,2)
01536       CHARACTER*15 TXTARG(ARGMAX,2)
01537       REAL REARG(ARGMAX,6)
01538       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01539       LOGICAL CHEK
01540 C     ****************** INPROC - DECLARATIONS **********************
 1541 C
01542 C     ****************** INPROC - PROCEDURE **********************
01543       WRITE(*,*) ' ENTER : VECTOR LINE TEXT DIST ANG DIANG'
01544       READ(*,'(A)',END=100) LINE
01545       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01546       IF ( NARGS.NE.6 ) THEN
01547          WRITE(*,*) ' ERROR :  WRONG NUMBER OF ARGUMENTS.'
01548          GOTO 100
01549       ENDIF
01550       IF (TYPE(1).EQ.'VECTOR') THEN
01551          XX1 = REARG(1,1)
01552          YY1 = REARG(1,2)
01553          ZZ1 = REARG(1,3)
01554       ELSEIF (TYPE(1).EQ.'TEXT') THEN
01555          CALL ATIND(TXTARG(1,1),N,*100)
01556          XX1 = X(N)
01557          YY1 = Y(N)
01558          ZZ1 = Z(N)
01559       ELSEIF (TYPE(1).EQ.'INTEGER') THEN
01560          XX1 = X(INTARG(2,1))
 1561          YY1 = Y(INTARG(2,1))
01562          ZZ1 = Z(INTARG(2,1))
```

```
01563       ELSE
01564         WRITE(*,*) ' ERROR :  WRONG TYPE OF ARGUMENT 1.'
01565         GOTO 100
01566       ENDIF
01567       IF (TYPE(2).EQ.'LINE') THEN
01568         PRINT*,'LINE=',(REARG(2,I),I=1,6)
01569         XX2 = REARG(2,1)
01570         YY2 = REARG(2,2)
01571         ZZ2 = REARG(2,3)
01572         XX3 = REARG(2,4)
01573         YY3 = REARG(2,5)
01574         ZZ3 = REARG(2,6)
01575        ELSE
01576         WRITE(*,*) ' ERROR : ARGUMENT 2 IS NOT LINE.'
01577         GOTO 100
01578        ENDIF
01579        IF (TYPE(3).EQ.'TEXT') THEN
01580         TEXT1 = TXTARG(3,1)
 1581        ELSE
01582         WRITE(*,*) ' ERROR :   ARGUMENT 4 IS NOT TEXT.'
01583         GOTO 100
01584       ENDIF
01585       IF ((TYPE(4).NE.'REAL'.AND.TYPE(4).NE.'INTEGER').OR.
01586      +    (TYPE(5).NE.'REAL'.AND.TYPE(5).NE.'INTEGER').OR.
01587      +    (TYPE(6).NE.'REAL'.AND.TYPE(6).NE.'INTEGER')) THEN
01588         WRITE(*,*) ' ERROR :  WRONG ARGUMENT 4 OR 5 OR 6'
01589         GOTO 100
01590       ELSE
01591         DIST = REARG(4,1)
01592         ANG = REARG(5,1)
01593         DIANG = REARG(6,1)
01594       ENDIF
01595 C     ******************* INPROC - PROCEDURE **********************
01596       X1 = XX1-XX2
01597       Y1 = YY1-YY2
01598       Z1 = ZZ1-ZZ2
01599       X2 = XX3-XX2
01600       Y2 = YY3-YY2
 1601       Z2 = ZZ3-ZZ2
01602 C     PRINT*,'XYZ2= ',X2,Y2,Z2
01603 C     PRINT*,'XXYYZZ1= ',XX1,YY1,ZZ1
01604 C     PRINT*,'XXYYZZ2= ',XX2,YY2,ZZ2
01605 C     PRINT*,'XYZ1= ',X1,Y1,Z1
01606       CALL CROSS (X3,Y3,Z3,X2,Y2,Z2,X1,Y1,Z1)
01607       CALL CROSS (X4,Y4,Z4,X3,Y3,Z3,X2,Y2,Z2)
01608 C     PRINT*,'XYZ3= ',X3,Y3,Z3
01609 C     PRINT*,'XYZ4= ',X4,Y4,Z4
01610 C     PRINT*,'XYZ4= ',X4,Y4,Z4
01611       CALL ONE (X5,Y5,Z5,X3,Y3,Z3,*100)
01612       CALL ONE (X6,Y6,Z6,X4,Y4,Z4,*100)
01613       ANGLE = DIANG*PI/180.0
01614       C = COS (ANGLE)
01615       S = SIN (ANGLE)
01616       X7 = X6*C+Y5*S
01617       Y7 = Y6*C+Y5*S
01618       Z7 = Z6*C+Z5*S
01619 C     PRINT*,'XYZ2= ',X2,Y2,Z2
01620       CALL ONE (X8,Y8,Z8,X2,Y2,Z2,*100)
```

```
      ANGLE = ANG*PI/180.0-PI/2.0
      C = COS (ANGLE)
      S = SIN (ANGLE)
      X9 = X7*C+X8*S
      Y9 = Y7*C+Y8*S
      Z9 = Z7*C+Z8*S
      X10 = X9*DIST+XX3
      Y10 = Y9*DIST+YY3
      Z10 = Z9*DIST+ZZ3
C
      NUMBER = NUMBER+1
      X(NUMBER) = X10
      Y(NUMBER) = Y10
      Z(NUMBER) = Z10
      TEXT(NUMBER) = TEXT1
      RETURN
C
  100 CONTINUE
      RETURN 1
      END
C
C     ********************
C
      SUBROUTINE SHIFT(*)
C
C     ********************
C
C     *** SHIFT THE MOLECULE ALONG A GIVEN VECTOR ***
C
      PARAMETER ( NATOMS=100 )
      COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
      CHARACTER*5 TEXT(NATOMS)
      COMMON /B/ TEXT
      LOGICAL FLAG
      SAVE N1,N2,XX,YY,ZZ,FLAG
C     ******************* INPROC - DECLARATIONS ************************
      INTEGER ARGMAX
      PARAMETER ( ARGMAX=10 )
      CHARACTER LINE*60
      INTEGER INTARG(ARGMAX,2)
      CHARACTER*15 TXTARG(ARGMAX,2)
      REAL REARG(ARGMAX,6)
      CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
      LOGICAL CHEK
C     ******************* INPROC - DECLARATIONS ************************
      DATA FLAG /.TRUE./
C
      GOTO 200
      ENTRY SHBACK
      IF (.NOT.FLAG) THEN
         WRITE(*,*) ' ERROR : SHIFTBACK WITHOUT PRECEDING SHIFT.'
         GOTO 100
      ENDIF
      FLAG = .FALSE.
      XX = -XX
      YY = -YY
      ZZ = -ZZ
      GOTO 300
```

```
01679   200 CONTINUE
01680 C     ******************** INPROC - PROCEDURE ************************
 1681       WRITE(*,*) ' ENTER :   RANGE VECTOR '
01682       READ(*,'(A)',END=100) LINE
01683       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01684       IF ( NARGS.NE.2 ) THEN
01685          WRITE(*,*) ' WRONG NUMBER OF ARGUMENTS.'
01686          GOTO 100
01687       ENDIF
01688       IF (TYPE(1).EQ.'RANGE') THEN
01689          N1 = INTARG(1,1)
01690          N2 = INTARG(1,2)
01691       ELSE
01692          WRITE(*,*) ' ERROR :  SECOND ARGUMENT MUST BE RANGE.'
01693          GOTO 100
01694       ENDIF
01695       IF (TYPE(2).EQ.'TEXT') THEN
01696
01697          CALL ATIND(TXTARG(2,1),N,*100)
01698          XX = -X(N)
01699          YY = -Y(N)
01700          ZZ = -Z(N)
 1701       ELSEIF (TYPE(2).EQ.'INTEGER') THEN
01702          N = INTARG(2,1)
01703          XX = -X(N)
01704          YY = -Y(N)
01705          ZZ = -Z(N)
01706       ELSEIF (TYPE(2).EQ.'VECTOR') THEN
01707          XX = -REARG(2,1)
01708          YY = -REARG(2,2)
01709          ZZ = -REARG(2,3)
01710       ELSE
01711          WRITE(*,'(A,I3,2A)') ' WRONG TYPE : ARGUMENT',2,'=',TYPE(2)
01712          GOTO 100
01713       ENDIF
01714 C     ******************** INPROC - PROCEDURE ************************
01715 C
01716       FLAG = .TRUE.
01717   300 CONTINUE
01718 C
01719       DO 500,I=N1,N2
01720          X(I) = X(I)+XX
 1721          Y(I) = Y(I)+YY
01722          Z(I) = Z(I)+ZZ
01723   500 CONTINUE
01724 C
01725       RETURN
01726 C
01727   100 CONTINUE
01728       RETURN 1
01729       END
01730 C
01731 C     ********************
01732 C
01733       SUBROUTINE MOVE(*)
01734 C
01735 C     ********************
01736 C
```

```
01737 C     *** MOVE A PART OF THE MOLECULE ALONG A LINE BY A GIVEN
01738 C         DISTANCE ***
01739 C
01740       PARAMETER ( NATOMS=100 )
 1741       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01742       CHARACTER*5 TEXT(NATOMS)
01743       COMMON /B/ TEXT
01744 C
01745       CHARACTER*5 TEXT1,TEXT2
01746       REAL SX,SY,SZ
01747       INTEGER N1,N2
01748       LOGICAL FLAG,FIRST
01749       SAVE N1,N2,SX,SY,SZ
01750 C     ****************** INPROC - DECLARATIONS **********************
01751       INTEGER ARGMAX
01752       PARAMETER ( ARGMAX=10 )
01753       CHARACTER LINE*60
01754       INTEGER INTARG(ARGMAX,2)
01755       CHARACTER*15 TXTARG(ARGMAX,2)
01756       REAL REARG(ARGMAX,6)
01757       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01758       LOGICAL CHEK
01759 C     ****************** INPROC - DECLARATIONS **********************
01760       DATA N1,N2 /2*0/
 1761       DATA FIRST /.FALSE./
01762 C
01763 C     ****************** INPROC - PROCEDURE **********************
01764       WRITE(*,*) ' ENTER :   RANGE LINE DISTANCE'
01765       READ(*,'(A)',END=100) LINE
01766       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01767       NARCHK = 3
01768       TYPCHK(1) = 'RANGE'
01769       TYPCHK(2) = 'LINE'
01770       TYPCHK(3) = 'REAL'
01771       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
01772       IF (.NOT.CHEK) THEN
01773          TYPCHK(3) = 'INTEGER'
01774          CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
01775       ENDIF
01776       N1 = INTARG(1,1)
01777       N2 = INTARG(1,2)
01778       X1 = REARG(2,1)
01779       Y1 = REARG(2,2)
01780       Z1 = REARG(2,3)
 1781       X2 = REARG(2,4)
01782       Y2 = REARG(2,5)
01783       Z2 = REARG(2,6)
01784       DIST = REARG(3,1)
01785
01786 C     ****************** INPROC - PROCEDURE **********************
01787 C
01788       X3 = X2-X1
01789       Y3 = Y2-Y1
01790       Z3 = Z2-Z1
01791       CALL ONE (X4,Y4,Z4,X3,Y3,Z3,*100)
01792       SX = X4*DIST
01793       SY = Y4*DIST
01794       SZ = Z4*DIST
```

```
01795 C
01796       FLAG = .FALSE.
01797       GOTO 600
01798       ENTRY MOCONT
01799       IF ( FIRST ) THEN
01800          WRITE(*,*) ' ERROR :  MOVECONT WITHOUT PRECEDING MOVE. '
 1801          GOTO 100
01802       ENDIF
01803       FIRST = .FALSE.
01804       FLAG = .TRUE.
01805   600 CONTINUE
01806 C
01807       DO 500,I=N1,N2
01808          X(I) = X(I)+SX
01809          Y(I) = Y(I)+SY
01810          Z(I) = Z(I)+SZ
01811   500 CONTINUE
01812       RETURN
01813 C
01814   100 CONTINUE
01815       RETURN 1
01816       END
01817 C
01818 C     ********************
01819 C
01820       SUBROUTINE TURN(*)
 1821 C
01822 C     ********************
01823 C
01824 C     *** TURN  ATOMA...ATOMB AROUND A AXIS GIVEN BY TWO POINTS ***
01825 C     *** ENTRY TBACK : MOVE THE WHOLE MOLECULE BACK
01826 C
01827 C     ******************** INPROC - DECLARATIONS ************************
01828       INTEGER ARGMAX
01829       PARAMETER ( ARGMAX=10 )
01830       CHARACTER LINE*60
01831       INTEGER INTARG(ARGMAX,2)
01832       CHARACTER*15 TXTARG(ARGMAX,2)
01833       REAL REARG(ARGMAX,6)
01834       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01835       LOGICAL CHEK
01836 C     ******************** INPROC - DECLARATIONS ************************
01837 C
01838       PARAMETER ( NATOMS=100 )
01839       PARAMETER ( PI = 3.141592654 )
01840       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
 1841       CHARACTER*5 TEXT(NATOMS)
01842       COMMON /B/ TEXT
01843       LOGICAL FLAG
01844       CHARACTER*5 TEXT1,TEXT2
01845       SAVE X1,Y1,Z1,X2,Y2,Z2,PHI,FLAG
01846       DATA FLAG /.FALSE./
01847 C
01848       GOTO 200
01849       ENTRY TBACK(*)
01850       IF (.NOT.FLAG) GOTO 100
01851       FLAG = .FALSE.
01852       PHI = -PHI
```

```
 1853       NSTART = 1
01854       NLAST = NUMBER
01855       GOTO 300
01856   200 CONTINUE
01857 C
01858 C     ****************** INPROC - PROCEDURE **********************
01859 C
01860       WRITE(*,*) ' ENTER :    RANGE LINE ANGLE :'
 1861        READ(*,'(A)',END=100) LINE
01862       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
01863       NARCHK = 3
01864 C     PRINT*,'TYPES = ',(TYPE(I),I=1,3)
01865       TYPCHK(1) = 'RANGE'
01866       TYPCHK(2) = 'LINE'
01867       TYPCHK(3) = 'REAL'
01868       CALL CHECK2 (NARGS,NARCHK,TYPE,TYPCHK,CHEK)
01869       IF (.NOT.CHEK) THEN
01870          TYPCHK(3) = 'INTEGER'
01871          CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
01872       ENDIF
01873       ANG = REARG(3,1)
01874       FLAG = .TRUE.
01875 C
01876       NSTART = INTARG(1,1)
01877       NLAST = INTARG(1,2)
01878       X1 = REARG(2,1)
01879       Y1 = REARG(2,2)
01880       Z1 = REARG(2,3)
 1881       X2 = REARG(2,4)
01882       Y2 = REARG(2,5)
01883       Z2 = REARG(2,6)
01884       PHI = ANG*PI/180.0
01885 C     PRINT*,' FIRST,LAST,X1,Y1,Z1,X2,Y2,Z2,ALPHA = ',NSTART,
01886 C    +         NLAST,X1,Y1,Z1,X2,Y2,Z2,ANG
01887   300 CONTINUE
01888 C
01889 C     ****************** INPROC - PROCEDURE **********************
01890 C
01891       X3 = X2-X1
01892       Y3 = Y2-Y1
01893       Z3 = Z2-Z1
01894       DO 400,I=NSTART,NLAST
01895          X(I) = X(I)-X1
01896          Y(I) = Y(I)-Y1
01897          Z(I) = Z(I)-Z1
01898   400 CONTINUE
01899 C     PRINT*,' BEFORE CALL TO TURMOL :'
01900 C     PRINT*,' NSTART,NLAST,X3,Y3,Z3,PHI= ',NSTART,NLAST,X3,Y3,Z3,PHI
 1901       CALL TURMOL (NSTART,NLAST,X3,Y3,Z3,PHI,*100)
01902       DO 500,I=NSTART,NLAST
01903          X(I) = X(I)+X1
01904          Y(I) = Y(I)+Y1
01905          Z(I) = Z(I)+Z1
01906   500 CONTINUE
01907 C
01908       RETURN
01909 C
01910   100 CONTINUE
```

```
01911       RETURN 1
01912       END
01913 C
01914 C     ***********************************
01915 C
01916       SUBROUTINE XTURN (*)
01917 C
01918 C     ***********************************
01919 C
01920 C     *** TURN NSTART...NLAST SO, THAT THE VECTOR COINCIDES WITH THE
 1921 C     *** X (Y OR Z) AXIS. ***
01922 C
01923       PARAMETER ( NATOMS=100 )
01924       PARAMETER ( PI = 3.141592654 )
01925       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
01926 C
01927       LOGICAL FLAG
01928       SAVE NSTART,NLAST,X2,Y2,Z2,PHI,FLAG
01929 C     ******************* INPROC - DECLARATIONS ************************
01930       INTEGER ARGMAX
01931       PARAMETER ( ARGMAX=10 )
01932       CHARACTER LINE*60
01933       INTEGER INTARG(ARGMAX,2)
01934       CHARACTER*15 TXTARG(ARGMAX,2)
01935       REAL REARG(ARGMAX,6)
01936       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
01937       LOGICAL CHEK
01938 C     ******************* INPROC - DECLARATIONS ************************
01939       DATA FLAG /.FALSE./
01940 C
 1941       XAX = 1.0
01942       YAX = 0.0
01943       ZAX = 0.0
01944       GOTO 200
01945       ENTRY YTURN (*)
01946       XAX = 0.0
01947       YAX = 1.0
01948       ZAX = 0.0
01949       GOTO 200
01950       ENTRY ZTURN (*)
01951       XAX = 0.0
01952       YAX = 0.0
01953       ZAX = 1.0
01954       GOTO 200
01955 C
01956       ENTRY AXBACK
01957       IF (.NOT.FLAG) THEN
01958          WRITE(*,*) ' ERROR : XYZTURNBACK WITHOUT PRECEDING XYZTURN.'
01959       ENDIF
01960       FLAG = .FALSE.
 1961       PHI = -PHI
01962       GOTO 300
01963 C
01964   200 CONTINUE
01965 C
01966 C     ******************* INPROC - PROCEDURE ************************
01967       WRITE(*,*) ' ENTER :   RANGE VECTOR'
01968       READ(*,'(A)',END=100) LINE
01969       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
```

```
 1970       IF (TYPE(1).EQ.'RANGE') THEN
01971         NSTART = INTARG(1,1)
01972         NLAST = INTARG(1,2)
01973       ELSE
01974         WRITE(*,*) ' ERROR :  ARGUMENT 1 MUST BE A RANGE.'
01975         GOTO 100
01976       ENDIF
01977       IF (TYPE(2).EQ.'VECTOR') THEN
01978         X1 = REARG(2,1)
01979         Y1 = REARG(2,2)
01980         Z1 = REARG(2,3)
 1981       ELSEIF (TYPE(2).EQ.'TEXT') THEN
01982         CALL ATIND(TXTARG(2,1),N,*100)
01983         X1 = X(N)
01984         Y1 = Y(N)
01985         Z1 = Z(N)
01986       ELSEIF (TYPE(2).EQ.'INTEGER') THEN
01987         N = INTARG(2,1)
01988         X1 = X(N)
01989         Y1 = Y(N)
01990         Z1 = Z(N)
01991       ELSE
01992         WRITE(*,*) ' THE SECOND ARGUMENT MUST BE A VECTOR.'
01993       ENDIF
01994 C     ****************** INPROC - PROCEDURE **********************
01995       FLAG = .TRUE.
01996 C
01997       CALL CROSS (X2,Y2,Z2,X1,Y1,Z1,XAX,YAX,ZAX)
01998       CALL ANGLE (ANG,XAX,YAX,ZAX,X1,Y1,Z1,*100)
01999       PHI = ANG*PI/180.0
02000   300 CONTINUE
 2001  C
02002       CALL TURMOL (NSTART,NLAST,X2,Y2,Z2,PHI,*100)
02003 C
02004       RETURN
02005 C
02006   100 CONTINUE
02007       RETURN 1
02008       END
02009 C
02010 C     ***********************************************
02011 C
02012       SUBROUTINE TURMOL (NSTART,NLAST,X1,Y1,Z1,PHI,*)
02013 C
02014 C     ***********************************************
02015 C
02016 C     *** TURN THE ATOMS N1...N2 AROUND A VECTOR ***
02017 C
02018       PARAMETER ( NATOMS=100 )
02019       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
02020 C
 2021       CALL ONE (CALP,CBET,CGAM,X1,Y1,Z1,*100)
02022       SALP2 = 1-CALP**2
02023       SBET2 = 1-CBET**2
02024       SGAM2 = 1-CGAM**2
02025       PHI1 = -PHI
02026       CPHI = COS(PHI1)
02027       SPHI = SIN(PHI1)
```

```
02028       CPHIC = 1-CPHI
02029       A11 = CALP**2+CPHI*SALP2
02030       A12 = CPHIC*CALP*CBET+SPHI*CGAM
02031       A13 = CPHIC*CALP*CGAM-SPHI*CBET
02032       A21 = CPHIC*CALP*CBET-SPHI*CGAM
02033       A22 = CBET**2+CPHI*SBET2
02034       A23 = CPHIC*CBET*CGAM+SPHI*CALP
02035       A31 = CPHIC*CALP*CGAM+SPHI*CBET
02036       A32 = CPHIC*CBET*CGAM-SPHI*CALP
02037       A33 = CGAM**2+CPHI*SGAM2
02038 C
02039       DO 400,I=NSTART,NLAST
02040       XX = X(I)
 2041       YY = Y(I)
02042       ZZ = Z(I)
02043       X(I) = XX*A11+YY*A12+ZZ*A13
02044       Y(I) = XX*A21+YY*A22+ZZ*A23
02045       Z(I) = XX*A31+YY*A32+ZZ*A33
02046   400 CONTINUE
02047 C
02048       RETURN
02049 C
02050   100 CONTINUE
02051       RETURN 1
02052 C
02053       END
02054 C
02055 C     ********************************************************
02056 C
02057       SUBROUTINE LINEUP(*)
02058 C
02059 C     ********************************************************
02060 C
 2061 C
02062 C     *** TURNS THE PART OF THE MOLECULE WHICH IS DEFINED BY RANGE
02063 C     *** SO, THAT LINE1 AND LINE2 BECOME ONE LINE AND VECTOR2 OF
02064 C     *** LINE1 AND VECTOR1 OF LINE2 COINCIDE. LINE2 MUST BE DEFINED
02065 C     *** BY ATOMS.
02066 C
02067       PARAMETER ( NATOMS=100 )
02068       COMMON /A/ X(NATOMS),Y(NATOMS),Z(NATOMS),NUMBER
02069       CHARACTER*5 TEXT(NATOMS)
02070       COMMON /B/ TEXT
02071 C
02072       PARAMETER ( PI = 3.141592654 )
02073       REAL XX1,YY1,ZZ1,XX2,YY2,ZZ2
02074 C     ****************** INPROC - DECLARATIONS **********************
02075       INTEGER ARGMAX
02076       PARAMETER ( ARGMAX=10 )
02077       CHARACTER LINE*60
02078       INTEGER INTARG(ARGMAX,2)
02079       CHARACTER*15 TXTARG(ARGMAX,2)
02080       REAL REARG(ARGMAX,6)
 2081       CHARACTER*7 TYPE(ARGMAX),TYPCHK(ARGMAX)
02082       LOGICAL CHEK
02083 C     ****************** INPROC - DECLARATIONS **********************
02084 C
02085 C     1) READ INPUT
02086 C
```

```
02087 C     ****************** INPROC - PROCEDURE **********************
02088       WRITE(*,*) ' ENTER :   RANGE LINE1 LINE2'
02089       READ(*,'(A)',END=100) LINE
02090       CALL INPROC(NARGS,TXTARG,INTARG,REARG,TYPE,LINE,*100)
02091       NARCHK = 3
02092       TYPCHK(1) = 'RANGE'
02093       TYPCHK(2) = 'LINE'
02094       TYPCHK(3) = 'LINE'
02095       CALL CHECK1 (NARGS,NARCHK,TYPE,TYPCHK,*100)
02096       NSTART = INTARG(1,1)
02097       NLAST = INTARG(1,2)
02098       XX1 = REARG(2,1)
02099       YY1 = REARG(2,2)
02100       ZZ1 = REARG(2,3)
 2101       XX2 = REARG(2,4)
02102       YY2 = REARG(2,5)
02103       ZZ2 = REARG(2,6)
02104       N3 = INTARG(3,1)
02105       N4 = INTARG(3,2)
02106       IF (N3.EQ.0 .OR. N4.EQ.0 ) THEN
02107          WRITE(*,*) ' ERROR : LINE2 MUST BE DEFINED WITH ATOMS.'
02108          GOTO 100
02109       ENDIF
02110 C     ****************** INPROC - PROCEDURE **********************
02111 C
02112 C     2) SHIFT ATOM1 INTO THE ORIGIN (SHIFT OVER WHOLE MOLECULE)
02113 C
02114       SX1 = XX1
02115       SY1 = YY1
02116       SZ1 = ZZ1
02117       DO 200,I=1,NUMBER
02118          X(I) = X(I)-SX1
02119          Y(I) = Y(I)-SY1
02120          Z(I) = Z(I)-SZ1
 2121   200 CONTINUE
02122       PRINT*,'AFTER 2) :'
02123       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02124 C
02125 C     3) TURN ATOM2 INTO THE Z - AXIS ( TURN OVER THE WHOLE MOLECULE )
02126 C
02127       X1 = XX2-XX1
02128       Y1 = YY2-YY1
02129       Z1 = ZZ2-ZZ1
02130       CALL CROSS (X2,Y2,Z2,X1,Y1,Z1,0.0,0.0,1.0)
02131       CALL ANGLE (ANG,0.0,0.0,1.0,X1,Y1,Z1,*100)
02132       PHI1 = ANG*PI/180.0
02133       CALL TURMOL (1,NUMBER,X2,Y2,Z2,PHI1,*100)
02134       BET = SQRT((XX1-XX2)**2+(YY1-YY2)**2+(ZZ1-ZZ2)**2)
02135       XX2 = 0.0
02136       YY2 = 0.0
02137       ZZ2 = BET
02138       PRINT*,'AFTER 3) :'
02139       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02140 C
 2141 C     4) SHIFT ATOM3 INTO ORIGIN ( SHIFT OVER ATOMA...ATOMB )
02142 C
02143       SX2 = X(N3)
02144       SY2 = Y(N3)
```

```
02145       SZ2 = Z(N3)
02146       DO 300,I=NSTART,NLAST
02147          X(I) = X(I)-SX2
02148          Y(I) = Y(I)-SY2
02149          Z(I) = Z(I)-SZ2
02150   300 CONTINUE
02151       PRINT*,'AFTER 4) :'
02152       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02153 C
02154 C     5) TURN ATOM4 INTO THE Z - AXIS ( TURN OVER ATOMA...ATOMB )
02155 C
02156       X3 = X(N4)
02157       Y3 = Y(N4)
02158       Z3 = Z(N4)
02159       CALL CROSS (X4,Y4,Z4,X3,Y3,Z3,0.0,0.0,1.0)
02160       CALL ANGLE (ANG,0.0,0.0,1.0,X3,Y3,Z3,*100)
 2161       PHI = ANG*PI/180.0
02162       CALL TURMOL (NSTART,NLAST,X4,Y4,Z4,PHI,*100)
02163       PRINT*,'AFTER 5) :'
02164       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02165 C
02166 C     6) SHIFT ATOM3 TO ATOM 2 ( SHIFT OVER ATOMA...ATOMB )
02167 C
02168       SX3 = XX2-X(N3)
02169       SY3 = YY2-Y(N3)
02170       SZ3 = ZZ2-Z(N3)
02171       DO 400,I=NSTART,NLAST
02172          X(I) = X(I)+SX3
02173          Y(I) = Y(I)+SY3
02174          Z(I) = Z(I)+SZ3
02175   400 CONTINUE
02176       PRINT*,'AFTER 6) :'
02177       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02178 C
02179 C     7) MAKE INVERTED TURN OF 3) ( TURN OVER WHOLE MOLECULE )
02180 C
 2181       PHI1 = -PHI1
02182       CALL TURMOL (1,NUMBER,X2,Y2,Z2,PHI1,*100)
02183 C
02184 C     8) MAKE INVERTED SHIFT OF 2) ( TURN OVER WHOLE MOLECULE )
02185 C
02186       DO 500,I=1,NUMBER
02187          X(I) = X(I)+SX1
02188          Y(I) = Y(I)+SY1
02189          Z(I) = Z(I)+SZ1
02190   500 CONTINUE
02191       PRINT*,'AFTER 7) :'
02192       WRITE(*,'(T3,3F10.5)') (X(I),Y(I),Z(I),I=1,NUMBER)
02193 C
02194       RETURN
02195 C
02196   100 CONTINUE
02197       RETURN 1
02198       END
02199 C
02200 C     ********************
 KOGEN    OLD     C1  FP 80  TRUNC=80 SIZE=2338 LINE=2200 COL=1 ALT=80
1 LINE(S) TRUNCATED.
```

```
02200 C     ********************
      1...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
>2201 C
02202       SUBROUTINE HELP
02203 C
02204 C     ********************
02205 C
02206 C     *** DISPLAY INFORMATION ABOUT THIS PROGRAM ***
02207 C
02208       WRITE(*,'(16(/T2,A))')
02209      +'HOW TO USE KOGEN :                                              ',
02210      +'                                                                ',
02211      +'1) INSERTING AN ATOM INTO THE MOLECULE :                        ',
02212      +'                                                                ',
02213      +' INPUT (TEXT) (X) (Y) (Z)                                       ',
02214      +' (TEXT) MUST BE A STRING WITH 1 TO 5 CHARACTERS.                ',
02215      +' IT MUST BE UNIQUE FOR EACH ATOM.                               ',
02216      +'                                                                ',
02217      +'2) READING A SET OF ATOMS FROM A FILE AND INSERTING IT INTO THE',
02218      +'   MOLECULE :                                                   ',
02219      +'                                                                ',
02220      +' READ (FILENAME)                                                ',
 2221      +'                                                                ',
02222      +' (FILENAME) IS THE NAME OF A FILE WHERE THE COORDINATES APPEAR ',
02223      +' IN THE FORMAT THAT IS USED FOR THE IBMOL PROGRAM.THE FILE MUST',
02224      +' ONLY CONTAIN THE COORDINATES.                                  ',
02225       WRITE(*,'(16(/T2,A))')
02226      +'                                                                ',
02227      +'3) WRITE (FILENAME)                                             ',
02228      +'                                                                ',
02229      +' (FILENAME) IS THE FILE TO WHICH THE COORDINATES ARE            ',
02230      +' WRITTEN IN THE IBMOL - FORMAT.IF THE FILENAME HAS ALREADY      ',
02231      +' APPEARED IN A FORMER WRITE STATEMENT, THE COORDINATES ARE      ',
02232      +' APPENDED TO THE END OF THE FILE.                               ',
02233      +'                                                                ',
02234      +'4) DISP                                                         ',
02235      +'                                                                ',
02236      +' THIS COMMAND DISPLAYS THE COORDINATES OF THE ATOMS.            ',
02237      +'                                                                ',
02238      +'5) DIST (TEXT1) (TEXT2)                                         ',
02239      +'                                                                ',
02240      +' DISPLAYS THE DISTANCE BETWEEN TWO ATOMS.                       ',
 2241      +'                                                                '
02242       WRITE(*,'(16(/T2,A))')
02243      +'6) ANGLE (TEXT1) (TEXT2) (TEXT3)                                ',
02244      +'                                                                ',
02245      +' DISPLAYS THE ANGLE (1-2-3) BETWEEN THE ATOMS WITH THE TEXTS.   ',
02246      +'                                                                ',
02247      +'7) DIANG (TEXT1) (TEXT2) (TEXT3) (TEXT4)                        ',
02248      +'                                                                ',
02249      +' DISPLAYS THE (DIHEDRAL) ANGLE BETWEEN THE 2 PLANES DEFINED     ',
02250      +' BY THE ATOMS 1,2,3 AND 2,3,4.                                  ',
02251      +'                                                                ',
02252      +'8) ATOM (TEXT1) (TEXT2) (TEXT3) (DIST) (ANG) (DIANG)            ',
02253      +'                                                                ',
02254      +' INSERTS A NEW ATOM INTO THE MOLECULE. THE NEW ATOM IS THE      ',
02255      +' 4 TH IN THE STRUCTURE 1-2-3-4. (DIST) IS THE DISTANCE BETWEEN ',
02256      +' 3 AND 4, (ANG) IS THE ANGLE 2-3-4 AND (DIANG) IS THE ANGLE     ',
```

```
02257     +' BETWEEN THE PLANES OF 1-2-3 AND 2-3-4.                    ',
02258     +'                                                           '
02259      WRITE(*,'(16(/T2,A))')
02260     +'9) SHIFT <X> <Y> <Z>                                       ',
 2261     +'                                                           ',
02262     +' SUBTRACTS THE VECTOR (X,Y,Z) TO THE COORDINATES OF ALL ATOMS. ',
02263     +'                                                           ',
02264     +'10) SHIFTBACK                                              ',
02265     +'                                                           ',
02266     +' ADDS THE VECTOR (X,Y,Z) OF THE LAST SHIFT TO ALL ATOMS.   ',
02267     +'                                                           ',
02268     +'11) TURN <X1> <Y1> <Z1> <X2> <Y2> <Z2> <ANGLE>             ',
02269     +'                                                           ',
02270     +' TURNS THE MOLECULE AROUND THE AXIS DEFINED BY THE TWO POINTS ',
02271     +' (X1,Y1,Z1) AND (X2,Y2,Z2) BY THE ANGLE (ANGLE).           ',
02272     +'                                                           ',
02273     +'12) TURNBACK                                               ',
02274     +'                                                           ',
02275     +' TURNS THE MOLECULE AROUND THE AXIS DEFINED BY THE LAST TURN '
02276      WRITE(*,'(16(/T2,A))')
02277     +' COMMAND BY THE OPPOSITE ANGLE OF THE LAST TURN COMMAND.   ',
02278     +'                                                           ',
02279     +'13) LIST <N>                                               ',
02280     +'                                                           ',
 2281     +' DISPLAYS FORMER INPUT AND OUTPUT ON THE TERMINAL.         ',
02282     +' <N> DETERMINES HOW MANY COMMANDS YOU WANT TO GO BACK.     ',
02283     +'                                                           ',
02284     +'14) HELP                                                   ',
02285     +'                                                           ',
02286     +' GIVES YOU THIS INFORMATION.                               ',
02287     +'                                                           ',
02288     +'15) DEL <TEXT>                                             ',
02289     +'                                                           ',
02290     +' REMOVES AN ATOM FROM THE MOLECULE                         ',
02291     +'                                                           ',
02292     +'15) I* <TEXT> <X> <Y> <Z> ...                              '
02293      WRITE(*,'(16(/T2,A))')
02294     +'                                                           ',
02295     +' ALLOWS INPUT OF MANY ATOMS (UNTIL TEXT=''END''IS ENCOUNTERED).',
02296     +' ',
02297     +'16) D*',
02298     +' ',
02299     +' DELETES ALL ATOMS.',
02300     +' ',
 2301     +'17) XTURN <X> <Y> <Z> , YTURN <X> <Y> <Z> , ZTURN <X> <Y> <Z>',
02302     +' ',
02303     +' TURN THE VECTOR (X,Y,Z) AROUND THE ORIGIN INTO THE X, Y, OR Z-',
02304     +' AXIS AND THE MOLECULE WITH HIM.',
02305     +' ',
02306     +'18) XTURNBACK , YTURNBACK , ZTURNBACK',
02307     +' ',
02308     +' PERFORM THE INVERSE OPERATION AS THE LAST XTURN, YTURN OR',
02309     +' ZTURN - COMMANDS.'
02310      WRITE(*,'(16(/T2,A))')
02311     +' ',
02312     +'19) TABLE <MINDIST>',
02313     +' ',
02314     +' GIVES YOU A TABLE OF THE DISTANCES AND ANGLES BETWEEN ATOMS,',
02315     +' WHOSE DISTANCES ARE SMALLER THAN MINDIST.',
```

```
02316      +' ',
02317      +'19) MOVE (TEXT1) (TEXT2) (X1) (Y1) (Z1) (X2) (Y2) (Z2) (DIST)',
02318      +' ',
02319      +' MOVES THE PART OF THE MOLECULE FROM TEXT1 UNTIL TEXT2 BY THE',
02320      +' SPECIFIED DISTANCE ALONG THE VECTOR X,Y,Z..',
 2321      +' ',
02322      +'20) MOVCONT',
02323      +' ',
02324      +' REPEATES THE LAST MOVE - COMMAND ( IF THERE WAS ANY)',
02325      +' ',
02326      +'FORMATS OF THE TERMINAL INPUT :'
02327       WRITE(*,'(9(/T2,A))')
02328      +' ',
02329      +' ALL COMMAND NAMES HAVE TO START IN THE FIRST COLUMS OF A LINE.',
02330      +' AFTER ENTERING THE COMMAND NAME, THE PROGRAM ASKS FOR ENTERING',
02331      +' THE OTHER PARAMETERS, IF THERE ARE ANY.                        ',
02332      +' FILENAMES ARE FORTRAN FILE NAMES AND HAVE TO BE ENCLOSED      ',
02333      +' IN '' CHARACTERS.                                             ',
02334      +' THE PARAMETERS ARE READ IN FREE FORMAT.                       ',
02335      +' THE ANGLES APPEAR IN DEGREES.                                 ',
02336      +' IF TWO OR MORE ATOMS HAVE THE SAME TEXT, THE FIRST IS TAKEN.'
02337       RETURN
02338       END
02339 * * * END OF FILE * * *
```

APPENDIX II

CNDO Program

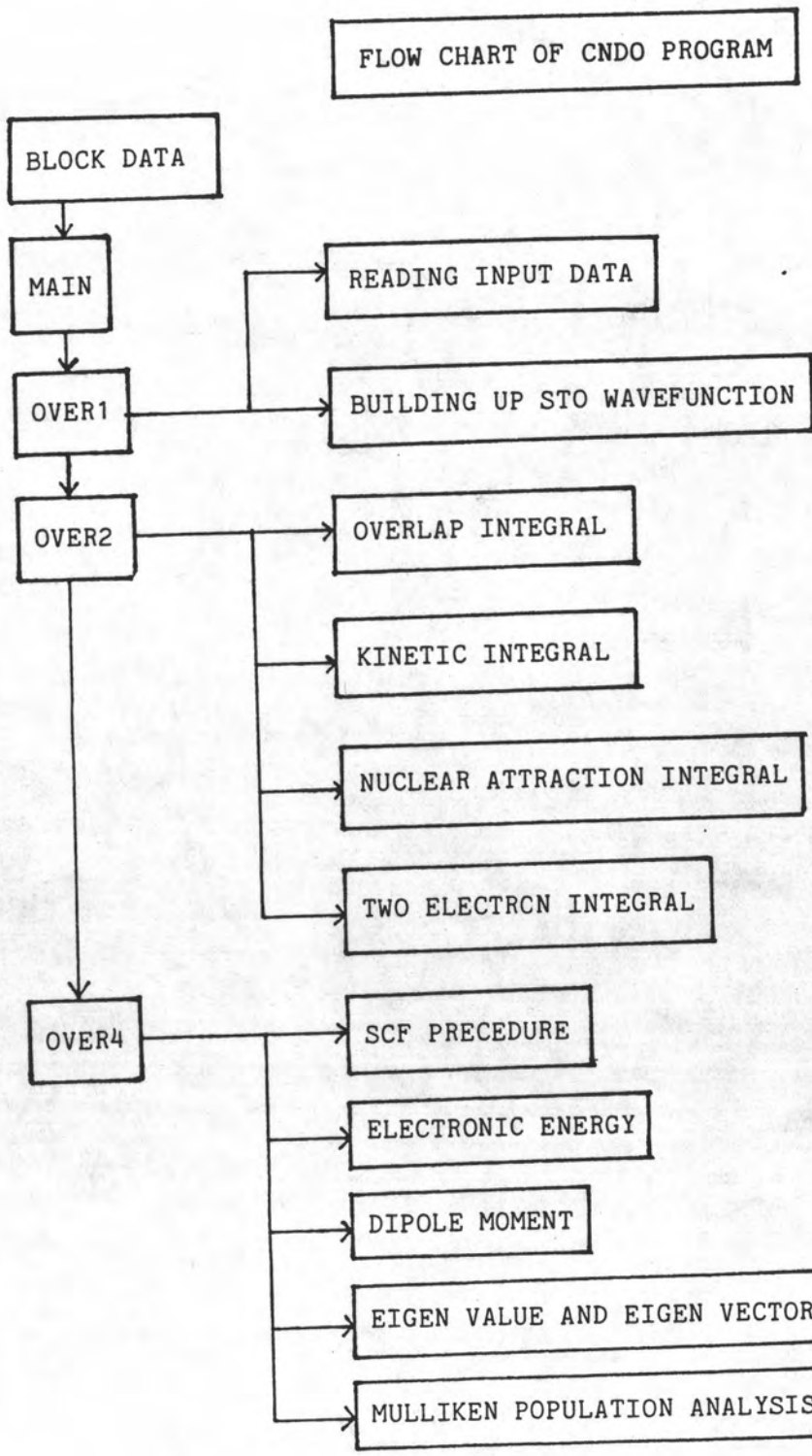
The appendix gives the Fortran listing and a sample input for mefloquine drug molecule. This program was supported by the Quantum Chemistry Group of the Innsbruck University which used a CDC computer and a FORTRAN V computer. However, the programs has been corrected for using on IBM 3031/08 and a VS FORTRAN compiler.

The program will calculate CNDO wavefunctions for any closed shell molecules containning the elements H to Cl. The program has been extended the matrices up to 140 atoms or 270 basis funtions. The operation of each mainly subroutien are described as followed:

Subroutine OVER1 is required for reading the coordinate input and assigns the coefficients used in the calculation of overlap and coulomb integrals. The overlap matrix, stored in the first array of COMMON /C1/, and the coulomb integral matrix, stored in COMMON /C6/, are computed in subroutine OVER2. Integrals are calculated for pairs of atoms using a local diatomic coordinates system. Then the subroutine HARMTR will convert the diatomic overlap integrals to the molecular coordinate system.

The SCF begin in subroutine OVER4 that have two main original subroutine, HUCKCL and SCFCLO. The initial guess density matrix are prepared by the former and the latter lakes as input initial density matrixs and CNDO hamiltonian. The new Fock matrix is formed by first adding the integrals and diagonalized so a new density matrix is formed which is used to construct a new Fock matrix. The procidure is repeated until the electronic energy coverges to 10^{-6} and computed after each new Fock matrix is formed

and before it is diagonalized. Iteration limit is allowed to 100.



```
 0001 ************************************************************************
00002 ***                                                                  ***
00003 ***    COMPLETE NEGLECT OF DIFERENTIAL OVERLAP (CNDO/2) PROGRAM      ***
00004 ***                                                                  ***
00005 ************************************************************************
00006       BLOCK DATA
00007       REAL STARTX,ENDX
00008       INTEGER DIMA,DIMB,NN1,INK
00009       PARAMETER(NN1=0,INK=500)
00010       PARAMETER(STARTX=1.5,ENDX=1.5)
00011       PARAMETER(DIMA=140,DIMB=270)
00012       COMMON /C1/V(13),IFALL,KFALL,NV,NVAR,VAR(140),ESV(140),IVAR(140)
00013       COMMON/C2/C(140,3)
00014       COMMON/C4/EL(18)
00015       COMMON/C5/ORB(9)
00016       COMMON/C8/AA(3,4),ES,XX(3)
00017       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
00018       CHARACTER*4 EL,ORB,OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80
00019       DATA (V(I),I=1,13)/13*0.0/
00020       DATA IFALL,KFALL,NV,NVAR/0,0,1,0/
 0021       DATA (VAR(I),I=1,140)/140*0.0/
00022       DATA (ESV(I),I=1,140)/140*0.0/
00023       DATA (IVAR(I),I=1,140)/140*0.0/
00024       DATA OPTION,CNDO,INDO/'CNDO','CNDO','INDO'/
00025       DATA OPNCLO,CLOSED,OPEN/'CLSD','CLSD','OPEN'/
00026       DATA ORB(1),ORB(2)/'   S',' PX'/
00027       DATA ORB(3),ORB(4)/' PY',' PZ'/
00028       DATA ORB(5),ORB(6)/' DZ2',' DXZ'/
00029       DATA ORB(7),ORB(8)/' DYZ','DX-Y'/
00030       DATA ORB(9)/' DXY'/
00031       DATA EL(1),EL(2),EL(3)/'  H',' HE',' LI'/
00032       DATA EL(4),EL(5),EL(6)/' BE','  B','  C'/
00033       DATA EL(7),EL(8),EL(9)/'  N','  O','  F'/
00034       DATA EL(10),EL(11),EL(12)/' NE',' NA',' MG'/
00035       DATA EL(13),EL(14),EL(15)/' AL',' SI','  P'/
00036       DATA EL(16),EL(17),EL(18)/'  S',' CL',' AR'/
00037       END
00038 C
00039 C*****SUBROUTINE READ THE COORDINATE INPUT*****
00040       SUBROUTINE KOORD(X,IST)
 0041       INTEGER DIMA,PRTOPT,AN,CHARGE
00042       PARAMETER(DIMA=140)
00043       COMMON/C1/V(13),IFALL,KFALL,NV,NVAR,VAR(140),ESV(140),IVAR(140)
00044       COMMON/C2/C(DIMA,3)
00045       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,ITER,EPS,LIMEX
00046       DATA FAKT/0.01745329252/
00047       NV=1
00048       V(IST)=X
00049       DO1 I=1,NATOMS
00050       READ2,C(I,1),C(I,2),C(I,3)
00051     1 CONTINUE
00052     2 FORMAT(2X,3F10.6)
00053       RETURN
00054       END
```

```
00055 C
00056 C*****PROGRAM CNINDO(INPUT,OUTPUT,TAPE7=/80)*****
00057       PROGRAM CNINDO
00058       INTEGER DIMA,DIMB
00059       PARAMETER(DIMA=140,DIMB=270)
00060       COMMON    ABC(DIMB,DIMB,3)
 0061       COMMON/C1/V(13),IFALL,KFALL,NV,NVAR,VAR(140),ESV(140),IVAR(140)
00062       COMMON/C2/C(DIMA,3)
00063       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
00064       COMMON/C4/EL(18)
00065       COMMON/C5/ORB(9)
00066       COMMON/C6/G(DIMA,DIMA),XXX(DIMB,7),ENERGY,ENERG
00067       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,ITER,EPS,LIMEX
00068      1,NROWLY,X(12),XS,STARTX,ENDX,E(12),IST,L1,INK,INKMIN,NST,SCHR,EAL
00069      2,NN1
00070       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
00071       CHARACTER*4 EL,ORB,OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80
00072       COMMON/C8/AA(3,4),ES,XX(3)
00073       INTEGER PRTOPT
00074       INTEGER AN,CHARGE,CZ,U,ULIM,OCCA,OCCB
00075 C**************************************************************************
00076 C     INPUT IS READ IN THE FOLLOWING ORDER
00077 C  (1)AN IDENTIFICATION CARD WHICH IS PRINTED AT THE BEGINNING OF THE RU
00078 C  (2)OPTION(WAVE FUNCTION OPTION) AND OPNCLO(OPEN OR CLOSED SHELL)
00079 C         THE FORMAT IS A4,1X,A4  AND THE KEY WORDS ARE-
00080 C            FOR THE WAVEFUNCTION(A4)    CNDO    INDO
 0081 C            FOR THE OPEN-CLOSED OPTION(A6)    OPEN   CLSD
00082 C  (3)NATOMS,CHARGE,MULTIP   FORMAT(3I4)
00083 C  (4)ATOMIC NUMBER, X COORDINATE, Y COORDINATE, Z COORDINATE - 1 CARD/A
00084 C        FORMAT(I4,3(3X,F12.7))
00085 C**************************************************************************
00086       DO 60 I=1,3
00087       DO 60 J=1,4
00088       AA(I,J)=1.0
00089    60 CONTINUE
00090       DO 65 I=1,DIMA
00091       DO 65 J=1,3
00092       C(I,J)=0.0
00093    65 CONTINUE
00094       DO 70 I=1,DIMB
00095       DO 70 J=1,7
00096       XXX(I,J)=0.0
00097    70 CONTINUE
00098   100 NST=1
00099   200 CALL OVER1
00100       CALL OVER2
 0101 C     IF(OPNCLO.EQ.OPEN) GOTO 90
00102       CALL OVER4
00103 C     GOTO 95
00104 C 90  CALL OVER3
00105    95 CONTINUE
00106       STOP
00107       END
00108 C
00109 C*****SUBROUTINE SCFOUT(OP,MOP)
00110 C     THIS ROUTINE PRINTS THE ARRAY IN COMMON/ARRAYS/ WHICH IS DESIGNATE
00111 C     MOP. IF OP = 1 THE EIGENVALUES CONTAINED IN COMMON/C6/ ARE ALSO
00112 C     PRINTED. IF OP= 0 THE EIGENVALUES ARE NOT PRINTED
00113       SUBROUTINE SCFOUT(OP,MOP)
```

```
00114       INTEGER DIMA,DIMB
00115       PARAMETER(DIMA=140,DIMB=270)
00116       COMMON    A(DIMB,DIMB,3)
00117       COMMON/C6/G(DIMA,DIMA),XXX(DIMB,3),EPSILN(DIMB)
00118       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N
00119       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
00120       COMMON/C5/ORB(9)
 0121       COMMON/C4/EL(18)
00122       CHARACTER*4 EL,ORB
00123       INTEGER PRTOPT,OP,AN,ANII,CZ,U,ULIM,CHARGE,OCCA,OCCB
00124       DO 120 M=1,N,12
00125       K=M+11
00126       IF (K.GT.N) K=N
00127       PRINT100
00128       IF (OP.EQ.0) GO TO 50
00129       PRINT40,  (EPSILN(I) ,I=M,K)
00130    40 FORMAT(/'0EIGENVALUES---',12F9.4//)
00131    50 CONTINUE
00132       PRINT60,  (I,I=M,K)
00133    60 FORMAT(14X,12I9)
00134       DO 120 I=1,N
00135       II=U(I)
00136       ANII=AN(II)
00137       L=I-LLIM(II)+1
00138       PRINT80,  I,II,EL(ANII),ORB(L),(A(I,J,MOP),J=M,K)
00139    80 FORMAT(1X,2I3,A,1X,A,12(F9.4))
00140       IF (I.EQ.ULIM(II)) PRINT100
 0141   100 FORMAT(1X)
00142   120 CONTINUE
00143       PRINT200
00144   200 FORMAT(//)
00145       RETURN
00146       END
00147 C
00148 C*****SUBROUTINE CALCULATE ATOMIC INTEGRALS FOR CNDO CALCULATIONS
00149       SUBROUTINE OVER2
00150       INTEGER Y,Z,PRTOPT
00151       INTEGER DIMA,DIMB
00152       PARAMETER(DIMA=140,DIMB=270)
00153       COMMON    S(DIMB,DIMB),Y(9,5,203),Z(17,45)
00154       COMMON/C2/C(DIMA,3)
00155       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
00156       COMMON/C6/GAMMA(DIMA,DIMA),T(9,9),PAIRS(9,9),TEMP(9,9)
00157       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,ITER,EPS,LIMEX
00158       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
00159       CHARACTER*4 EL,ORB,OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80
00160       DIMENSION MU(18),NC(18),LC(9),MC(9),E(3)
 0161       DIMENSION P(DIMB,DIMB)
00162       EQUIVALENCE (P(1,1),Y(1,1,1))
00163       REAL MU,NUM,K1,K2
00164       CHARACTER ERRMSG*60
00165       INTEGER AN,ULIM,ULK,ULL,CZ,U,CHARGE,ANL,ANK,OCCA,OCCB
00166 C     ASSIGNMENT OF ANGULAR MOMENTUM QUANTUM NOS. TO ATOMIC ORBITALS
00167       DATA LC/0,1,1,1,2,2,2,2,2/
00168       DATA MC/0,1,-1,0,0,1,-1,2,-2/
00169 C     DETERMINATION OF SIZE OF AO BASIS IN AND CORE CHARGE CZ
00170       N=0
00171       DO 60 I=1,NATOMS
00172       LLIM(I)=N+1
```

```
00173       K=1
00174       IF (AN(I).LT.11) GO TO 20
00175       N=N+9
00176       CZ(I)=AN(I)-10
00177       GO TO 50
00178    20 IF (AN(I).LT.3) GO TO 40
00179       N=N+4
00180       CZ(I) = AN(I)-2
 0181       GO TO 50
00182    40 N=N+1
00183       CZ(I)= AN(I)
00184    50 CONTINUE
00185       ULIM(I) = N
00186    60 CONTINUE
00187       IF (N.LE.(DIMB)) GO TO 65
00188       WRITE(ERRMSG,666)N,DIMB
00189   666 FORMAT('0 **** SIZE OF AO BASIS =',I4,9X,'(LIMIT=',I3,') ****')
00190 C     CALL SYSTEM(52,ERRMSG)
00191    65 CONTINUE
00192 C     FILL U ARRAY---U(J) IDENTIFIES THE ATOM TO WHICH ORBITAL J IS
00193 C     ATTACHED E.G. ORBITAL 32 ATTACHED TO  ATOM 7, ETC.
00194       DO 70 K=1,NATOMS
00195       LLK = LLIM(K)
00196       ULK = ULIM(K)
00197       LIM = ULK+1-LLK
00198       DO 70 I=1,LIM
00199       J = LLK+I-1
00200    70 U(J) = K
 0201 C     ASSIGNMENT OF ORBITAL EXPONENTS TO ATOMS BY SLATERS RULES
00202       MU(2)=1.7E0
00203       MU(1)=1.2E0
00204       NC(1)=1
00205       NC(2)=1
00206       DO 80 I=3,10
00207       NC(I)=2
00208    80 MU(I)=.325E0*(I-1)
00209       DO 90 I=11,18
00210       NC(I)=3
00211    90 MU(I)=(.65E0*(I)-4.95E0)/3.E0
00212 C     STEP THRU PAIRS OF ATOMS
00213       DO 320 K=1,NATOMS
00214       DO 320 L=K,NATOMS
00215 C     CALCULATE UNIT VECTOR ALONG INTERATOM AXIS,E
00216       R=0.
00217       DO 101 I=1,3
00218       E(I) = C(L,I)-C(K,I)
00219       R = R+E(I)**2
00220   101 CONTINUE
 0221       IF (K.EQ.L) GO TO 105
00222       R=SQRT(R)
00223       IF (R.GE.1E-6) GO TO 103
00224 C     CALL SYSTEM(52,'0*** FAULTY COORDINATES')
00225   103 DO 102 I=1,3
00226   102 E(I)=E(I)/R
00227   105 LLK = LLIM(K)
00228       LLL = LLIM(L)
00229       ULK = ULIM(K)
00230       ULL = ULIM(L)
00231       NORBK=ULK-LLK+1
```

```
00232       NORBL=ULL-LLL+1
00233       ANK=AN(K)
00234       ANL=AN(L)
00235 C     LOOP THRU PAIRS OF BASIS FUNCTIONS, ONE ON EACH ATOM
00236       DO 200 I=1,NORBK
00237       DO 200 J=1,NORBL
00238       IF(K.EQ.L) GO TO 160
00239       IF(MC( I).NE.MC(J)) GO TO 180
00240       IF(MC(I).LT.0) GO TO 140
 0241       PAIRS(I,J)=SQRT((MU(ANK)*R)**(2*NC(ANK)+1)*(MU(ANL)*R)**(2*NC(ANL
00242      1)+1)/(FACT(2*NC(ANK))*FACT(2*NC(ANL))))*(-1.E0)**(LC(J)+MC(J))
00243      2*SS(NC(ANK),LC(I),MC(I),NC(ANL),LC(J),MU(ANK)*R,MU(ANL)*R)
00244       GO TO 190
00245   140 PAIRS(I,J)=PAIRS(I-1,J-1)
00246       GO TO 190
00247   160 IF (I.EQ.J) GO TO 170
00248   180 PAIRS(I,J)=0.0E0
00249       GO TO 190
00250   170 PAIRS(I,J)=1.0E0
00251   190 CONTINUE
00252   200 CONTINUE
00253       LCULK=LC(NORBK)
00254       LCULL=LC(NORBL)
00255       MAXL=MAX0(LCULK,LCULL)
00256       IF(R.LE.0.000001E0) GO TO 250
00257 C     ROTATE INTEGRALS FROM DIATOMIC BASIS TO MOLECULAR BASIS
00258       CALL HARMTR(MAXL,E)
00259       DO 230 I=1,NORBK
00260       DO 230 J=1,NORBL
 0261       TEMP(I,J) = 0.E0
00262       DO 230 KK=1,NORBL
00263       TEMP(I,J) = TEMP(I,J)+T(J,KK)*PAIRS(I,KK)
00264   230 CONTINUE
00265       DO 240 I=1,NORBK
00266       DO 240 J=1,NORBL
00267       PAIRS(I,J) = 0.E0
00268       DO 240 KK=1,NORBK
00269       PAIRS(I,J) = PAIRS(I,J)+T(I,KK)*TEMP(KK,J)
00270   240 CONTINUE
00271 C     FILL S MATRIX
00272   250 CONTINUE
00273       DO 260 I=1,NORBK
00274       LLKP=LLK+I-1
00275       DO 260 J=1,NORBL
00276       LLLP=LLL+J-1
00277   260 S(LLKP,LLLP)=PAIRS(I,J)
00278 C     COMPUTATION OF 1-CENTER COULOMB INTEGRALS OVER SLATER S FUNCTIONS
00279       N1=NC(ANK)
00280       N2=NC(ANL)
 0281       K1=MU(ANK)
00282       K2=MU(ANL)
00283       IF(K.NE.L) GO TO 290
00284       TERM1 = FACT(2*N1-1)/((2.E0*K2)**(2*N1))
00285       TERM2 = 0.E0
00286       LIM = 2*N1
00287       DO 280 J=1,LIM
00288       NUM =(J)*(2.E0*K1)**(2*N1-J)*FACT(4*N1-J-1)
00289       DEN = FACT(2*N1-J)*2.E0*(N1)*(2.E0*(K1+K2))**(4*N1-J)
00290       TERM2 = TERM2 + NUM/DEN
```

```
00291   280 CONTINUE
00292       GO TO 310
00293 C     COMPUTATION OF 2-CENTER COULOMB INTEGRALS OVER SLATER S FUNCTIONS
00294   290 IF (OPNCLO.EQ.OPEN) GO TO 295
00295       IF (R.LT.4.0) GO TO 295
00296       GAMMA(K,L)=1.0/R
00297       GO TO 320
00298   295 TERM1=(R/2.E0)**(2*N2)*SS(0,0,0,2*N2-1,0,0.E0,2.E0*K(2*R)
00299       TERM2 = 0.E0
00300       LIM = 2*N1
 0301       DO 300 J=1,LIM
00302   300 TERM2 = TERM2+((J)*(2.E0*K(1)**(2*N1-J)*(R/2.E0)**(2*
00303      1N1-J+2*N2))/ (FACT(2*N1-J)*2.E0*(N1))*SS(2*N1-J,0,0,2*N2-1,0
00304      2,2.E0*K(1*R,2.E0*K(2*R)
00305   310 GAMMA(K,L) = ((2.E0*K2)**(2*N2+1)/FACT(2*N2))*(TERM1-TERM2)
00306   320 CONTINUE
00307 C     SYMMETRIZATION OF OVERLAP AND COULOMB INTEGRAL MATRICES
00308       DO 330 I=1,N
00309       DO 330 J=I,N
00310   330 S(J,I) = S(I,J)
00311       DO 340 I=1,NATOMS
00312       DO 340 J=I,NATOMS
00313   340 GAMMA(J,I) = GAMMA(I,J)
00314       IF(PRTOPT.LE.1)GO TO 380
00315       PRINT350
00316   350 FORMAT('0'/4(/'+OVERLAP INTEGRAL MATRIX'))
00317       CALL MATOUT(N,1)
00318 C     TRANSFER GAMMA TO 80X80 MATRIX P FOR PRINTING
00319       DO 360 I=1,NATOMS
00320       DO 360 J=1,NATOMS
 0321   360 P(I,J)=GAMMA(I,J)
00322       PRINT370
00323   370 FORMAT(/4(/'+COULOMB INTEGRAL MATRIX'))
00324       CALL MATOUT(NATOMS,2)
00325   380 CONTINUE
00326       RETURN
00327       END
00328 C
00329 C*****SUBROUTINE CONVERT INTEGRAL FROM DIATOMIC BASIS TO MOL BASIS**
00330       SUBROUTINE HARMTR(MAXL,E)
00331       INTEGER DIMA
00332       PARAMETER(DIMA=140)
00333       DIMENSION T(9,9),E(3)
00334       COMMON/C6/YYY(DIMA,DIMA),T
00335       DATA SQRT3/1.732050808/
00336       COST = E(3)
00337       IF((1.-COST**2).GT.0.0000000001) GO TO 20
00338       SINT = 0.E0
00339       GO TO 30
00340    20 SINT=SQRT(1.E0-COST**2)
 0341    30 CONTINUE
00342       IF(SINT.GT.0.000001E0)  GO TO 50
00343       COSP = 1.E0
00344       SINP = 0.E0
00345       GO TO 70
00346    50 COSP = E(1)/SINT
00347       SINP = E(2)/SINT
00348    70 CONTINUE
00349       DO 80 I=1,9
```

```
00350       DO 80 J=1,9
00351    80 T(I,J) = 0.E0
00352       T(1,1) =1.E0
00353       IF(MAXL-1)120,110,100
00354   100 COS2T = COST**2-SINT**2
00355       SIN2T = 2.E0*SINT*COST
00356       COS2P = COSP**2-SINP**2
00357       SIN2P = 2.E0*SINP*COSP
00358 C     TRANSFORMATION MATRIX ELEMENTS FOR D FUNCTIONS
00359       T(5,5) = (3.E0*COST**2-1.E0)/2.E0
00360       T(5,6) = -SQRT3   *SIN2T/2.E0
 0361       T(5,8) = SQRT3   *SINT**2/2.E0
00362       T(6,5) = SQRT3   *SIN2T*COSP/2.E0
00363       T(6,6) = COS2T*COSP
00364       T(6,7) = -COST*SINP
00365       T(6,8)=-T(6,5)/SQRT3
00366       T(6,9)=SINT*SINP
00367       T(7,5)=SQRT3  *SIN2T*SINP/2.E0
00368       T(7,6)=COS2T*SINP
00369       T(7,7)=COST*COSP
00370       T(7,8)=-T(7,5)/SQRT3
00371       T(7,9) = -SINT*COSP
00372       T(8,5) = SQRT3   *SINT**2*COS2P/2.E0
00373       T(8,6) = SIN2T*COS2P/2.E0
00374       T(8,7) = -SINT*SIN2P
00375        T(8,8) = (1.E0+COST**2)*COS2P/2.E0
00376       T(8,9) = -COST*SIN2P
00377       T(9,5) = SQRT3   *SINT**2*SIN2P/2.E0
00378       T(9,6) = SIN2T*SIN2P/2.E0
00379       T(9,7) = SINT*COS2P
00380       T(9,8) = (1.E0+COST**2)*SIN2P/2.E0
 0381       T(9,9) = COST*COS2P
00382   110 CONTINUE
00383 C     TRANSFORMATION MATRIX ELEMENTS FOR P FUNCTIONS
00384       T(2,2) = COST*COSP
00385       T(2,3) = -SINP
00386       T(2,4) = SINT*COSP
00387       T(3,2) = COST*SINP
00388       T(3,3) = COSP
00389       T(3,4) = SINT*SINP
00390       T(4,2) = -SINT
00391       T(4,4) = COST
00392   120 CONTINUE
00393       DO 130 I=1,9
00394       DO 130 J=1,9
00395       IF(ABS(T(I,J)).LE.1.E-8) T(I,J)=0.
00396   130 CONTINUE
00397       RETURN
00398       END
00399 C
00400 C*****FUNCT SUBROUTINE
 0401       FUNCTION FACT(N)
00402       PRODT = 1.E0
00403       DO 30 I=2,N
00404    30 PRODT=PRODT*(I)
00405       FACT=PRODT
00406       RETURN
00407       END
00408 C
```

```
00409 C     FUNCT SUBROUTINE FOR CALCULATE THE REDUCED OVERLAP INTEGRALS*****
00410       FUNCTION SS(NN1,LL1,MM,NN2,LL2,ALPHA,BETA)
00411       INTEGER DIMA,DIMB
00412       PARAMETER(DIMA=140,DIMB=270)
00413       INTEGER Y,Z
00414       COMMON    S(DIMB,DIMB),Y(9,5,203),Z(17,45)
00415       COMMON/C6/G(DIMA,DIMA),XXX(9,9,3),A(18),B(18)
00416       INTEGER ULIM
00417       N1=NN1
00418       L1=LL1
00419       M=MM
00420       N2=NN2
 0421       L2=LL2
00422       P =(ALPHA + BETA)/2.E0
00423       PT=(ALPHA - BETA)/2.E0
00424       X = 0.E0
00425       M=IABS(M)
00426 C     REVERSE QUANTUM NUMBERS IF NECESSARY
00427       IF(L2-L1)20,10,30
00428    10 IF(N2.GE.N1)GO TO 30
00429    20 K=N1
00430       N1= N2
00431       N2= K
00432       K= L1
00433       L1= L2
00434       L2= K
00435       PT=-PT
00436    30 CONTINUE
00437       K = MOD((N1+N2-L1-L2),2)
00438 C     FIND A AND B INTEGRALS
00439       CALL AINTGS(P,N1+N2)
00440       CALL BINTGS(PT,N1+N2)
 0441       IF((L1.GT.0).OR.(L2.GT.0)) GO TO 60
00442 C     BEGIN SECTION USED FOR OVERLAP INTEGRALS INVOLVING S FUNCTIONS
00443 C     FIND Z TABLE NUMBER L
00444       L = (90-17*N1+N1**2-2*N2)/2
00445       ULIM = N1+N2
00446       LLIM = 0
00447       DO 50 I=LLIM,ULIM
00448       NNI1=N1+N2-I+1
00449    50 X=X+Z(I+1,L)*A(I+1)*B(NNI1)/2.E0
00450       SS=X
00451       GO TO 80
00452 C     BEGIN SECTION USED FOR OVERLAPS INVOLVING NON-S FUNCTIONS
00453 C     FIND Y TABLE NUMBER L
00454    60 L=(5-M)*(24-10*M+M**2)*(83-30*M+3*M**2)/120+
00455      1  (30-9*L1+L1**2-2*N1)*(28-9*L1+L1**2-2*N1)/8+
00456      2  (30-9*L2+L2**2-2*N2)/2
00457       LLIM = 0
00458       DO 70 I=LLIM,8
00459       ULIM=4 - MOD(K+I,2)
00460       DO 70 J=LLIM,ULIM
 0461       IIII=2*J+MOD(K+I,2)+1
00462    70 X=X+Y(I+1,J+1,L)*A(I+1)*B(IIII)
00463       SS = X*(FACT(M+1)/8.E0)**2*SQRT((2*L1+1)*FACT(L1-M)*
00464      1 (2*L2+1)*FACT(L2-M)/(4.E0*FACT(L1+M)*FACT(L2+M)))
00465    80 CONTINUE
00466       RETURN
00467       END
```

```
00468 C
00469 C*****SUBROUTINE
00470       SUBROUTINE AINTGS(X,K)
00471       INTEGER DIMA
00472       PARAMETER(DIMA=140)
00473       COMMON/C6/G(DIMA,DIMA),XXX(9,9,3),A(18),B(18)
00474       A(1) =EXP(-X)/X
00475       DO 10  I=1,K
00476    10 A(I+1) =(A(I)*(I)+EXP(-X))/X
00477       RETURN
00478       END
00479 C
00480 C*****SUBROUTINE
 0481       SUBROUTINE BINTGS(XX,KK)
00482 C     FILLS ARRAY OF B-INTEGRALS. NOTE THAT B(I) IS B(I-1) IN THE
00483 C     USUAL NOTATION
00484 C     FOR X.GT.3                    EXPONENTIAL FORMULA IS USED
00485 C     FOR 2.LT.X.LE.3 AND K.LE.10   EXPONENTIAL FORMULA IS USED
00486 C     FOR 2.LT.X.LE.3 AND K.GT.10   15 TERM SERIES IS USED
00487 C     FOR 1.LT.X .E.2 AND K.LE.7    EXPONENTIAL FORMULA IS USED
00488 C     FOR 1.LT.X.LE.2 AND K.GT.7    12 TERM SERIES IS USED
00489 C     FOR .5.LT.X.LE.1 AND K.LE.5   EXPONENTIAL FORMULA IS USED
00490 C     FOR .5.LT.X.LE.1 AND K.GT.5    7 TERM SERIES IS USED
00491 C     FOR X.LE..5                    6 TERM SERIES IS USED
00492 C     ********************************************************************
00493       INTEGER DIMA
00494       PARAMETER(DIMA=140)
00495       COMMON/C6/G(DIMA,DIMA),XXX(9,9,3),A(18),B(18)
00496       DATA I0/0/
00497       X=XX
00498       K=KK
00499       ABSX=ABS(X)
00500       IF(ABSX.GT.3.E0) GO TO 120
 0501       IF(ABSX.GT.2.E0) GO TO 20
00502       IF(ABSX.GT.1.E0) GO TO 50
00503       IF(ABSX.GT..5E0) GO TO 80
00504       IF(ABSX.GT..000001E0) GO TO 110
00505       DO 180 I=I0,K,2
00506       B(I+2)=0.0
00507   180 B(I+1)=2.0/(I+1)
00508       RETURN
00509   110 LAST=6
00510       GO TO 140
00511    80 IF(K.LE.5) GO TO 120
00512       LAST=7
00513       GO TO 140
00514    50 IF(K.LE.7) GO TO 120
00515       LAST=12
00516       GO TO 140
00517    20 IF(K.LE.10) GO TO 120
00518       LAST=15
00519       GO TO 140
00520   120 EXPX=EXP(X)
 0521       EXPMX=1.E0/EXPX
00522       B(1)=(EXPX-EXPMX)/X
00523       DO 130 I=1,K
00524       EXPX=-EXPX
00525   130 B(I+1)=(I*B(I)+EXPX-EXPMX)/X
00526       RETURN
```

```
                URNED EIGENVALUES IN ALGEBRAIC ASCENDING ORDER
 0527   140 DO 160 I=I0,K
00528        Y=0.E0
00529        II=MOD(I,2)
00530        DO 150 M=II,LAST,2
00531   150 Y=Y+(-X)**M*2.0/(FACT(M)*(M+I+1))
00532   160 B(I+1)=Y
00533        RETURN
00534        END
00535 C
00536 C*****SUBROUTINE
00537        SUBROUTINE MATOUT(NN,MATOP)
00538        INTEGER DIMB
00539        PARAMETER(DIMB=270)
00540        COMMON     A(DIMB,DIMB,3)
 0541         N=NN
00542        DO 60 M=1,N,12
00543        K=M+11
00544        IF (K.GT.N) K=N
00545        PRINT40,  (J,J=M,K)
00546    40 FORMAT('0     ',12I9)
00547        DO 60 I=1,N
00548        PRINT50,  I,(A(I,J,MATOP),J=M,K)
00549    50 FORMAT(I4,4X,12F9.4)
00550    60 CONTINUE
00551        PRINT70
00552    70 FORMAT(//)
00553        RETURN
00554        END
00555 C
00556 C*****SUBROUTINE EIGNOP(NN,RHO)
00557 C     RHO= UPPER LIMIT FOR OFF-DIAGONAL ELEMENT
00558 C     NN= SIZE OF MATRIX
00559 C     A = F MATRIX (ONLY LOWER TRIANGLE IS USED + THIS IS DESTROYED)
00560 C     EIG = RETURNED EIGENVALUES IN ALGEBRAIC ASCENDING ORDER
 0561 C     VEC = RETURNED EIGENVECTORS IN COLUMNS
00562        SUBROUTINE EIGNOP(NN,RHO)
00563        INTEGER DIMA,DIMB
00564        PARAMETER(DIMA=140,DIMB=270)
00565        COMMON     A(DIMB,DIMB),VEC(DIMB,DIMB)
00566        COMMON/C6/XXX(DIMA,DIMA),GAMMA(DIMB),BETA(DIMB),BETASQ(DIMB),
00567       ,EIG(DIMB),W(DIMB)
00568 C     THE FOLLOWING DIMENSIONED VARIABLES ARE EQUIVALENCED
00569        DIMENSION P(DIMB),Q(DIMB)
00570        EQUIVALENCE (P(1),BETA(1)),(Q(1),BETA(1))
00571        DIMENSION IPOSV(DIMB),IVPOS(DIMB),IORD(DIMB)
00572        EQUIVALENCE (IPOSV(1),GAMMA(1)),(IVPOS(1),BETA(1)),
00573       1(IORD(1),BETASQ(1))
00574        RHOSQ=RHO*RHO
00575        N=NN
00576        IF (N .EQ. 0) GO TO 640
00577        N1=N-1
00578        N2=N-2
00579        GAMMA(1)=A(1,1)
00580        IF(N2) 200,190,40
 0581    40 DO 180 NR=1,N2
00582        B=A(NR+1,NR)
00583        S=0.E0
00584        DO 50 I=NR,N2
```

```
00585    50 S=S+A(I+2,NR)**2
00586 C     PREPARE FOR POSSIBLE BYPASS OF TRANSFORMATION
00587       A(NR+1,NR)=0.E0
00588       IF (S) 170,170,60
00589    60 S=S+B*B
00590       SGN=+1.E0
00591       IF (B) 70,80,80
00592    70 SGN=-1.E0
00593    80 SQRTS=SQRT(S)
00594       D=SGN/(SQRTS+SQRTS)
00595       TEMP=SQRT(.5E0+B*D)
00596       W(NR)=TEMP
00597       A(NR+1,NR)=TEMP
00598       D=D/TEMP
00599       B=-SGN*SQRTS
00600 C     D IS FACTOR OF PROPORTIONALITY. NOW COMPUTE AND SAVE W VECTOR.
 0601 C     EXTRA SINGLY SUBSCRIPTED W VECTOR USED FOR SPEED.
00602       DO 90 I=NR,N2
00603       TEMP=D*A(I+2,NR)
00604       W(I+1)=TEMP
00605    90 A(I+2,NR)=TEMP
00606 C     PREMULTIPLY VECTOR W BY MATRIX A TO OBTAIN P VECTOR.
00607 C     SIMULTANEOUSLY ACCUMULATE DOT PRODUCT WP,(THE SCALAR K)
00608       WTAW=0.E0
00609       DO 140 I=NR,N1
00610       SUM=0.E0
00611       DO 100 J=NR,I
00612   100 SUM=SUM+A(I+1,J+1)*W(J)
00613       I1=I+1
00614       IF(N1-I1) 130,110,110
00615   110 DO 120 J=I1,N1
00616   120 SUM=SUM+A(J+1,I+1)*W(J)
00617   130 P(I)=SUM
00618   140 WTAW=WTAW+SUM*W(I)
00619 C     P VECTOR AND SCALAR K  NOW STORED. NEXT COMPUTE Q VECTOR
00620       DO 150 I=NR,N1
 0621   150 Q(I)=P(I)-WTAW*W(I)
00622 C     NOW FORM PAP MATRIX, REQUIRED PART
00623       DO 160 J=NR,N1
00624       QJ=Q(J)
00625       WJ=W(J)
00626       DO 160 I=J,N1
00627   160 A(I+1,J+1)=A(I+1,J+1)-2.E0*(W(I)*QJ+WJ*Q(I))
00628   170 BETA(NR)=B
00629       BETASQ(NR)=B*B
00630   180 GAMMA(NR+1)=A(NR+1,NR+1)
00631   190 B=A(N,N-1)
00632       BETA(N-1)=B
00633       BETASQ(N-1)=B*B
00634       GAMMA(N)=A(N,N)
00635   200 BETASQ(N)=0.E0
00636 C     ADJOIN AN IDENTITY MATRIX TO BE POSTMULTIPLIED BY ROTATIONS.
00637       DO 220 I=1,N
00638       DO 210 J=1,N
00639   210 VEC(I,J)=0.E0
00640   220 VEC(I,I)=1.E0
 0641       M=N
00642       SUM=0.E0
00643       NPAS=1
```

```
00644       GO TO 350
00645   230 SUM=SUM+SHIFT
00646       COSA=1.E0
00647       G=GAMMA(1)-SHIFT
00648       PP=G
00649       PPBS=PP*PP+BETASQ(1)
00650       PPBR=SQRT(PPBS)
00651       DO 320 J=1,M
00652       COSAP=COSA
00653       IF (PPBS.GT.1.E-12) GO TO 250
00654       SINA=0.E0
00655       SINA2=0.E0
00656       COSA=1.E0
00657       GO TO 290
00658   250 SINA=BETA(J)/PPBR
00659       SINA2=BETASQ(J)/PPBS
00660       COSA=PP/PPBR
 0661 C     POSTMULTIPLY IDENTITY BY P-TRANSPOSE MATRIX
00662       NT=J+NPAS
00663       IF(NT .LT. N) GO TO 270
00664       NT=N
00665   270 DO 280 I=1,NT
00666       TEMP=COSA*VEC(I,J)+SINA*VEC(I,J+1)
00667       VEC(I,J+1)=-SINA*VEC(I,J)+COSA*VEC(I,J+1)
00668   280 VEC(I,J)=TEMP
00669   290 DIA=GAMMA(J+1)-SHIFT
00670       U=SINA2*(G+DIA)
00671       GAMMA(J)=G+U
00672       G=DIA-U
00673       PP=DIA*COSA-SINA*COSAP*BETA(J)
00674       IF(J .NE. M) GO TO 310
00675       BETA(J)=SINA*PP
00676       BETASQ(J)=SINA2*PP*PP
00677       GO TO 330
00678   310 PPBS=PP*PP+BETASQ(J+1)
00679       PPBR=SQRT(PPBS)
00680       BETA(J)=SINA*PPBR
 0681   320 BETASQ(J)=SINA2*PPBS
00682   330 GAMMA(M+1)=G
00683 C     TEST FOR CONVERGENCE OF LAST DIAGONAL ELEMENT
00684       NPAS=NPAS+1
00685       IF(BETASQ(M) .GT. RHOSQ) GO TO 370
00686   340 EIG(M+1)=GAMMA(M+1)+SUM
00687   350 BETA(M)=0.E0
00688       BETASQ(M)=0.E0
00689       M=M-1
00690       IF(M .EQ. 0) GO TO 400
00691       IF(BETASQ(M) .LE. RHOSQ) GO TO 340
00692 C     TAKE ROOT OF CORNER 2 BY 2 NEAREST TO LOWER DIAGONAL IN VALUE
00693 C     AS ESTIMATE OF EIGENVALUE TO USE FOR SHIFT
00694   370 A2=GAMMA(M+1)
00695       R2=0.5E0*A2
00696       R1=0.5E0*GAMMA(M)
00697       R12=R1+R2
00698       DIF=R1-R2
00699       TEMP=SQRT(DIF*DIF+BETASQ(M))
00700       R1=R12+TEMP
 0701       R2=R12-TEMP
00702       DIF=ABS(A2-R1)-ABS(A2-R2)
```

```
00703       IF(DIF .LT. 0.E0) GO TO 390
00704       SHIFT=R2
00705       GO TO 230
00706   390 SHIFT=R1
00707       GO TO 230
00708   400 EIG(1)=GAMMA(1)+SUM
00709 C     INITIALIZE AUXILIARY TABLES REQUIRED FOR REARRANGING THE VECTORS
00710       DO 410 J=1,N
00711       IPOSV(J)=J
00712       IVPOS(J)=J
00713   410 IORD(J)=J
00714 C     USE A TRANSPOSITION SORT TO ORDER THE EIGENVALUES
00715       M=N
00716       GO TO 450
00717   420 DO 440 J=1,M
00718       IF (EIG(J) .LE. EIG(J+1)) GO TO 440
00719       TEMP=EIG(J)
00720       EIG(J)=EIG(J+1)
 0721       EIG(J+1)=TEMP
00722       ITEMP=IORD(J)
00723       IORD(J)=IORD(J+1)
00724       IORD(J+1)=ITEMP
00725   440 CONTINUE
00726   450 M=M-1
00727       IF(M .NE. 0) GO TO 420
00728       IF(N1 .EQ. 0) GO TO 510
00729       DO 500 L=1,N1
00730       NV=IORD(L)
00731       NP=IPOSV(NV)
00732       IF(NP .EQ. L) GO TO 500
00733       LV=IVPOS(L)
00734       IVPOS(NP)=LV
00735       IPOSV(LV)=NP
00736       DO 490 I=1,N
00737       TEMP=VEC(I,L)
00738       VEC(I,L)=VEC(I,NP)
00739   490 VEC(I,NP)=TEMP
00740   500 CONTINUE
 0741   510 CONTINUE
00742 C     BACK TRANSFORM THE VECTORS OF THE TRIPLE DIAGONAL MATRIX
00743       DO 570 NRR=1,N
00744       K=N1
00745   520 K=K-1
00746       IF(K .LE. 0) GO TO 560
00747       SUM=0.E0
00748       DO 540 I=K,N1
00749   540 SUM=SUM+VEC(I+1,NRR)*A(I+1,K)
00750       SUM=SUM+SUM
00751       DO 550 I=K,N1
00752   550 VEC(I+1,NRR)=VEC(I+1,NRR)-SUM*A(I+1,K)
00753       GO TO 520
00754   560 CONTINUE
00755   570 CONTINUE
00756   640 RETURN
00757       END
00758 C
00759 C*****SUBROUTINE OVER4   EXTENDED HUCKL FOR CLOSED SHELLS (HUCKCL)****
00760 C     OVERLAPS ARE IN MATX A, COULOMB INTEGRALS (GAMMA) ARE IN MATX G
 0761       SUBROUTINE OVER4
```

```
00762       INTEGER DIMA,DIMB
00763       PARAMETER(DIMA=140,DIMB=270)
00764       INTEGER PRTOPT
00765       COMMON      A(DIMB,DIMB),B(DIMB,DIMB)
00766       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
00767       COMMON/C6/G(DIMA,DIMA),Q(DIMB),XXX(DIMB),YYY(DIMB,5),ENERGY,ENERG
00768       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,IT,EPS,LIMEX
00769       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
00770       CHARACTER*4 OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80
00771       DIMENSION ENEG(18,3),BETA0(18)
00772       DIMENSION G1(18),F2(18)
00773       INTEGER CHARGE,OCCA,OCCB,UL,AN,CZ,U,ULIM,ANI
00774       INTEGER Z
00775       REAL YY(10)
00776       EQUIVALENCE(BETA0,YY)
00777       DATA IZ,Z,SUMYY,OLSUMY/2*0,2*0.0/
00778       DATA G1(3) /.092012/
00779       DATA G1(4) /.1407  /
00780       DATA G1(5) /.199265/
 0781       DATA G1(6) /.267708/
00782       DATA G1(7) /.346029/
00783       DATA G1(8) /.43423 /
00784       DATA G1(9) /.532305/
00785       DATA F2(3) /.049865     /
00786       DATA F2(4) /.089125     /
00787       DATA F2(5) /.13041      /
00788       DATA F2(6) /.17372      /
00789       DATA F2(7) /.219055     /
00790       DATA F2(8) /.266415     /
00791       DATA F2(9) /.31580      /
00792       DATA ENEG(1,1) /7.1761  /
00793       DATA ENEG(3,1) /3.1055  /
00794       DATA ENEG(3,2) /1.258   /
00795       DATA ENEG(4,1) /5.94557 /
00796       DATA ENEG(4,2) /2.563   /
00797       DATA ENEG(5,1) /9.59407 /
00798       DATA ENEG(5,2) /4.001   /
00799       DATA ENEG(6,1) /14.051  /
00800       DATA ENEG(6,2) /5.572   /
 0801       DATA ENEG(7,1) /19.31637/
00802       DATA ENEG(7,2) /7.275   /
00803       DATA ENEG(8,1) /25.39017/
00804       DATA ENEG(8,2) /9.111   /
00805       DATA ENEG(9,1) /32.2724 /
00806       DATA ENEG(9,2) /11.08   /
00807       DATA ENEG(11,1) /2.804  /
00808       DATA ENEG(11,2) /1.302  /
00809       DATA ENEG(11,3) /0.150  /
00810       DATA ENEG(12,1) /5.1254 /
00811       DATA ENEG(12,2) /2.0516 /
00812       DATA ENEG(12,3) /0.16195/
00813       DATA ENEG(13,1) /7.7706 /
00814       DATA ENEG(13,2) /2.9951 /
00815       DATA ENEG(13,3) /0.22425/
00816       DATA ENEG(14,1) /10.0327/
00817       DATA ENEG(14,2) /4.1325 /
00818       DATA ENEG(14,3) /0.337  /
00819       DATA ENEG(15,1) /14.0327/
00820       DATA ENEG(15,2) /5.4638 /
```

```
 NEWCNDO  FORTRAN  B1  F 80  TRUNC=72 SIZE=1823 LINE=820 COL=1 ALT=58
====>
00820       DATA ENEG(15,2) /5.4638 /
     |...+....1....+....2....+....3....+....4....+....5....+....6....+....7.)
00821       DATA ENEG(15,3) /0.500  /
00822       DATA ENEG(16,1) /17.6496/
00823       DATA ENEG(16,2) /6.989  /
00824       DATA ENEG(16,3) /0.71325/
00825       DATA ENEG(17,1) /21.5906/
00826       DATA ENEG(17,2) /8.7081 /
00827       DATA ENEG(17,3) /0.97695/
00828       DATA BETA0(1) / -9.     /
00829       DATA BETA0(3) / -9.     /
00830       DATA BETA0(4) / -13.    /
00831       DATA BETA0(5) / -17.    /
00832       DATA BETA0(6) / -21.    /
00833       DATA BETA0(7) / -25.    /
00834       DATA BETA0(8) / -31.    /
00835       DATA BETA0(9) / -39.    /
00836       DATA BETA0(11) /-7.7203 /
00837       DATA BETA0(12) /-9.4471 /
00838       DATA BETA0(13) /-11.3011/
00839       DATA BETA0(14) /-13.065 /
00840       DATA BETA0(15) /-15.070 /
 0841       DATA BETA0(16) /-18.150 /
00842       DATA BETA0(17) /-22.330 /
00843       DATA OLDENG,ELDENG/2*1E10/
00844       DATA RHO /1E-6/
00845 C     FIND NELECS AND FILL H CORE(DIAGONAL) WITH (I+A)/2
00846       NELECS=0
00847       DO 60 I=1,NATOMS
00848       NELECS=NELECS+CZ(I)
00849       LL =LLIM(I)
00850       UL =ULIM(I)
00851       ANI=AN(I)
00852       L=0
00853       DO 50 J=LL,UL
00854       L=L+1
00855       IF (L.EQ.1) GO TO 10
00856       IF (L.LT.5) GO TO 40
00857       A(J,J)=-ENEG(ANI,3)/27.21E0
00858       GO TO 50
00859    40 A(J,J)=-ENEG(ANI,2)/27.21E0
00860       GO TO 50
 0861    10 A(J,J)  =-ENEG(ANI,1)/27.21E0
00862    50 CONTINUE
00863    60 CONTINUE
00864       NELECS=NELECS-CHARGE
00865       OCCA=NELECS/2
00866 C     FORM HUCKEL HAMILTONIAN IN A (OFF DIAGONAL TWO CENTER TERMS)
00867       DO 90 I=2,N
00868       K=U(I)
00869       L=AN(K)
00870       UL=I-1
00871       DO 90 J=1,UL
00872       KK=U(J)
00873       LL=AN(KK)
00874       IF ((L.GT.9).OR.(LL.GT.9)) GO TO 70
00875       A(I,J)=A(I,J)*(BETA0(L)+BETA0(LL))/54.42E0
```

```
00876       A(J,I)=A(I,J)
00877       GO TO 90
00878    70 A(I,J)=0.75E0*A(I,J)*(BETA0(L)+BETA0(LL))/54.42E0
00879       A(J,I)=A(I,J)
00880    90 CONTINUE
 0881       DO 100 I=1,N
00882   100 Q(I)=A(I,I)
00883       NN=N
00884       CALL EIGNCL(NN,RHO)
00885 C     EIGENVECTORS (IN B) ARE CONVERTED INTO DENSITY MATRIX (IN B)
00886       DO 140 I=1,N
00887       DO 120 J=I,N
00888       XXX(J)=0.0E0
00889       DO 110 K=1,OCCA
00890   110 XXX(J)= XXX(J)+2.E0*B(I,K)*B(J,K)
00891   120 CONTINUE
00892       DO 130 J=I,N
00893   130 B(I,J)= XXX(J)
00894   140 CONTINUE
00895       DO 150 I=1,N
00896       DO 150 J=I,N
00897   150 B(J,I)=B(I,J)
00898 C     ADD V(AB) TO HCORE--CNDO
00899       DO 170 I=1,N
00900       J=U(I)
 0901       Q(I)=Q(I)  +0.5E0*G(J,J)
00902       DO 160 K=1,NATOMS
00903   160 Q(I)=Q(I)-(CZ(K)*G(J,K))
00904   170 CONTINUE
00905 C     EXIT SEGMENT IF ONLY CNDO APPROXIMATIONS ARE DESIRED
00906       IF (OPTION.EQ.CNDO) GO TO 290
00907 C     INDO MODIFICATION (CORRECTION TO U(I,I) )
00908       DO 280 I=1,NATOMS
00909       K=AN(I)
00910       J=LLIM(I)
00911       IF ((K.GT.1).AND.(K.LT.10)) GO TO 190
00912       GO TO 280
00913   190 IF (K.LE.3) GO TO 210
00914       Q(J)=Q(J)+(CZ(I)-1.5)*G1(K)/6.
00915   210 IF(K.EQ.3) GO TO 220
00916       IF(K.EQ.4) GO TO 240
00917       TEMP=G1(K)/3.+(CZ(I)-2.5)*2.*F2(K)/25.
00918       GO TO 260
00919   240 TEMP=G1(K)/4.E0
00920       GO TO 260
 0921   220 TEMP=G1(K)/12.E0
00922   260 CONTINUE
00923       DO 270 L=1,3
00924       JPL=J+L
00925   270 Q(JPL)=Q(JPL)+TEMP
00926   280 CONTINUE
00927   290 CONTINUE
00928       DO 310 I=1,N
00929       DO 300 J=I,N
00930   300 A(J,I)=A(I,J)
00931   310 A(I,I)=Q(I)
00932       IF(PRTOPT.LE.1)GO TO 380
00933       PRINT320
00934   320 FORMAT(/4(/'+ CORE HAMILTONIAN')/)
```

```
00935       CALL SCFOUT(0,1)
00936   380 CONTINUE
00937 C*****************************************************************
00938 C     CNDO/INDO CLOSED SHELL SCF SEGMENT
00939 C     GAMMA MATRIX CONTAINED IN G, CORE HAMILTONIAN CONTAINED IN Q AND
00940 C     UPPER TRIANGLE OF A, AND INITIAL DENSITY MATRIX CONTAINED IN B
 0941 C     OPTIONS   CNDO OR INDO
00942   410 CONTINUE
00943       Z = Z+1
00944       ENERGY = 0.E0
00945 C     TRANSFER CORE HAMILTONIAN TO LOWER TRIANGLE OF A
00946       DO 420 I=1,N
00947       A(I,I)=Q(I)
00948       DO 420 J=I,N
00949   420 A(J,I)=A(I,J)
00950       DO 430 I=1,N
00951       II=U(I)
00952       A(I,I)=A(I,I)-B(I,I)*G(II,II)*0.5E0
00953       DO 430 K=1,N
00954       JJ=U(K)
00955   430 A(I,I)=A(I,I)+B(K,K)*G(II,JJ)
00956       NM=N-1
00957       DO 440 I=1,NM
00958       II=U(I)
00959       LL=I+1
00960       DO 440 J=LL,N
 0961       JJ=U(J)
00962   440 A(J,I)=A(J,I)-B(J,I)*G(II,JJ)*0.5E0
00963 C     INDO MODIFICATION
00964       IF (OPTION.EQ.CNDO) GO TO 490
00965       DO 480 II=1,NATOMS
00966       K=AN(II)
00967       I=LLIM(II)
00968       IF (K.EQ.1) GO TO 480
00969       PAA=B(I,I)+B(I+1,I+1)+B(I+2,I+2)+B(I+3,I+3)
00970       A(I,I)=A(I,I)-(PAA-B(I,I)) *G1(K)/6.E0
00971       DO 470 J=1,3
00972       IPJ=I+J
00973       A(IPJ,IPJ)=A(IPJ,IPJ)-B(I,I)*G1(K)/6.E0-(PAA-B(I,I))*7.E0*
00974      1F2(K)/50.E0+B(IPJ,IPJ)*11.E0*F2(K)/50.E0
00975   470  A(IPJ,I)=A(IPJ,I)+B(I,IPJ)*G1(K)/2.E0
00976       I1=I+1
00977       I2=I+2
00978       I3=I+3
00979       A(I2,I1)=A(I2,I1)+B(I2,I1)*11.E0*F2(K)/50.E0
00980       A(I3,I1)=A(I3,I1)+B(I3,I1)*11.E0*F2(K)/50.E0
 0981       A(I3,I2)=A(I3,I2)+B(I3,I2)*11.E0*F2(K)/50.E0
00982   480 CONTINUE
00983   490 CONTINUE
00984       DO 500 I=1,N
00985   500 ENERGY=ENERGY+0.5E0*B(I,I)*(A(I,I)+Q(I))
00986       DO 505 I=1,NM
00987       LL=I+1
00988       DO 505 J=LL,N
00989   505 ENERGY=ENERGY+B(I,J)*(A(I,J)+A(J,I))
00990       IF(ABS(ENERGY-OLDENG).GE.EPS)GOTO 550
00991       IF(ABS(ENERGY-ELDENG).GE.EPS)GOTO 550
00992   530 PRINT540, ENERGY,Z
00993   540 FORMAT(//10X,'ELECTRONIC ENERGY =',F17.10,' A.U.',T70,'ENERGY SATI
```

```
 0994       GO TO 1570
00995   550 IZ=IZ+1
00996       YY(IZ)=ENERGY
00997       SUMYY=ENERGY*0.1+SUMYY
00998       IF (IZ.NE.10) GO TO 1560
00999 C           KONVERGIEREN MITTELWERTE ?
01000       IF (ABS(SUMYY-OLSUMY).GT.EPS) GO TO 1530
01001       ENERGY=SUMYY
01002       PRINT1520, YY
01003  1520 FORMAT('0 LAST ITERATION VALUES'//2X,10F13.5)
01004       GO TO 530
01005  1530 OLSUMY=SUMYY
01006       SUMYY=0.0
01007       IZ=0
01008  1560 ELDENG=OLDENG
01009       OLDENG=ENERGY
01010       IF (Z.LE.IT) GOTO 610
01011       IF (Z.EQ.IT+1) PRINT541, ENERGY
01012   541 FORMAT(//10X,'ELECTRONIC ENERGY =',F17.5,' A.U.',4(/'+',T70,'++++
01013      ++ ENERGY NOT SATISFIED +++++')/)
 1014 C    SYMMETRIZE F FOR PRINTING (MATRIX A)
01015  1570 DO 590 I=1,N
01016       DO 590 J=I,N
01017   590 A(I,J)=A(J,I)
01018   610 CONTINUE
01019       NN=N
01020       CALL EIGNCL(NN,RHO)
01021       IF (Z.LE.IT) GO TO 640
01022       IF(PRTOPT.LT.1)GO TO 640
01023       PRINT630
01024   630 FORMAT(/4(/'+ EIGENVALUES AND EIGENVECTORS'))
01025       CALL SCFOUT(1,2)
01026   640 CONTINUE
01027 C     EIGENVECTORS (IN B) ARE CONVERTED INTO DENSITY MATRIX (IN B)
01028       DO 680 I=1,N
01029       DO 660 J=I,N
01030       XXX(J)=0.0E0
01031       DO 650 K=1,OCCA
01032   650 XXX(J)= XXX(J)+B(I,K)*B(J,K)*2.0E0
01033   660 CONTINUE
 1034       DO 670 J=I,N
01035   670 B(I,J)= XXX(J)
01036   680 CONTINUE
01037       DO 690 I=1,N
01038       DO 690 J=I,N
01039   690 B(J,I)=B(I,J)
01040       IF (Z.LE.IT) GO TO 410
01041       CALL CPRINT
01042       RETURN
01043       END
01044 C
01045 C*****SUBROUTINE EIGNCL
01046       SUBROUTINE EIGNCL(P,THR)
01047       INTEGER P,P1
01048       LOGICAL TRANSF
01049       INTEGER DIMA,DIMB
01050       PARAMETER(DIMA=140,DIMB=270)
01051       COMMON  B(DIMB,DIMB),X(DIMB,DIMB)
01052       COMMON/C6/XXX(DIMA,DIMA),GAMMA(DIMB),BETA(DIMB),BETASQ(DIMB),
```

```
 1053      ,G(DIMB)
01054        DO 20 I=1,P
01055    20 G(I)=B(I,I)
01056       SR=.5
01057       DO1I=1,P
01058       DO2J=1,P
01059     2 X(I,J)=0.
01060     1 X(I,I)=1.E0
01061       DO4M=1,9
01062     5 TRANSF=.FALSE.
01063       DO6I=2,P
01064       I1=I-1
01065       I2=I+1
01066       DO6J=1,I1
01067       IF(ABS(B(I,J)).LT.SR)GOTO6
01068       TRANSF=.TRUE.
01069       Z=2.E0*B(I,J)
01070       R=G(J)-G(I)
01071       S2=Z/SIGN(SQRT(Z*Z+R*R),R)
01072       S1=S2/SQRT(2.*(1.+SQRT(ABS(1.-S2*S2))))
 1073       S=S1
01074       C=SQRT(1.E0-S1*S1)
01075       W=(Z*C-R*S)*S
01076       G(J)=G(J)+W
01077       G(I)=G(I)-W
01078       B(I,J)=0.
01079       IF(J.EQ.1)GOTO7
01080       J1=J-1
01081       DO8K=1,J1
01082       W=B(J,K)
01083       B(J,K)=W*C+B(I,K)*S
01084     8 B(I,K)=-W*S+B(I,K)*C
01085     7 IF(J.EQ.I1)GOTO9
01086       J1=J+1
01087       DO10K=J1,I1
01088       W=B(K,J)
01089       B(K,J)=W*C+B(I,K)*S
01090    10 B(I,K)=-W*S+B(I,K)*C
01091     9 IF(I.EQ.P)GOTO11
01092       DO12K=I2,P
 1093       W=B(K,J)
01094       B(K,J)=W*C+B(K,I)*S
01095    12 B(K,I)=-W*S+B(K,I)*C
01096    11 DO13K=1,P
01097       W=X(K,J)
01098       X(K,J)=W*C+X(K,I)*S
01099    13 X(K,I)=-W*S+X(K,I)*C
01100     6 CONTINUE
01101       IF(TRANSF)GOTO5
01102     4 SR=SR/10.E0
01103       P1=P-1
01104       DO15I=1,P1
01105       GMAX= 1E10
01106       DO16J=I,P
01107       IF(G(J).GE.GMAX)GOTO16
01108       GMAX=G(J)
01109       L=J
01110    16 CONTINUE
```

```
01111       G(L)=G(I)
01112       G(I)=GMAX
 1113       DO15K=1,P
01114       W=X(K,I)
01115       X(K,I)=X(K,L)
01116    15 X(K,L)=W
01117       RETURN
01118       END
01119 C
01120 C*****SUBROUTINE CPRINT CNDO-INDO SCF CLOSED SHELL- PRINTOUT SEGMENT
01121       SUBROUTINE CPRINT
01122       INTEGER PRTOPT
01123       INTEGER DIMA,DIMB
01124       PARAMETER(DIMA=140,DIMB=270)
01125       COMMON      A(DIMB,DIMB),B(DIMB,DIMB)
01126       COMMON/C6/G(DIMA,DIMA),G(DIMB),XXX(DIMB),YYY(DIMB,5),ENERGY,ENERG
01127       COMMON/C2/C(DIMA,3)
01128       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
01129       COMMON/C4/EL(18)
01130       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,IT,EPS,LIMEX
01131       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
01132       CHARACTER*4 EL,OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80
 1133       INTEGER CHARGE,AN,U,ULIM,OCCA,OCCB,UL,CZ,ANI
01134       DIMENSION DPM(3),DM(3),DMSP(3),DMPD(3)
01135       DIMENSION ATENG(18)
01136       DATA ATENG(1)/-0.6387302462 /
01137       DATA ATENG(3)/-.2321972405  /
01138       DATA ATENG(4)/-1.1454120355 /
01139       DATA ATENG(5)/-2.9774239048 /
01140       DATA ATENG(6)/-6.1649936261 /
01141       DATA ATENG(7)/-11.076874625 /
01142       DATA ATENG(8)/-18.081965865 /
01143       DATA ATENG(9)/-27.549130288 /
01144       DATA ATENG(11)/-.1977009568 /
01145       DATA ATENG(12)/-.8671913833 /
01146       DATA ATENG(13)/-2.0364557744/
01147       DATA ATENG(14)/-3.8979034686/
01148       DATA ATENG(15)/-6.7966009163/
01149       DATA ATENG(16)/-10.765817434/
01150       DATA ATENG(17)/-16.046701794/
01151       IF (OPTION.EQ.CNDO) GO TO 30
01152       ATENG(4)=-1.1219620354
 1153       ATENG(5)=-2.8725750048
01154       ATENG(6)=-5.9349548261
01155       ATENG(7)=-10.673174125
01156       ATENG(8)=-17.292085065
01157       ATENG(9)=-26.257437787
01158    30 ENERG=ENERGY
01159       K=NATOMS-1
01160       IF(PRTOPT.LT.1)GO TO 290
01161       PRINT40
01162    40 FORMAT(/4(/'+ DENSITY MATRIX'))
01163       CALL SCFOUT(0,2)
01164   290 DO 50 I=1,K
01165       L=I+1
01166       DO 50 J=L,NATOMS
01167       RAD=SQRT((C(I,1)-C(J,1))**2+(C(I,2)-C(J,2))**2
01168      1         +(C(I,3)-C(J,3))**2)
01169    50 ENERGY=ENERGY+(CZ(I)*CZ(J))/RAD
```

```
01170       ENERG=ENERGY-ENERG
01171       PRINT60, ENERG,ENERGY
01172    60 FORMAT(/'0  NUCLEAR REPULSION ENERGY =',F12.4,' A.U.',
 1173      */13X,' TOTTAL ENERGY =',F12.4,' A.U.')
01174       ENERG=ENERGY
01175       DO 70 I=1,NATOMS
01176       ANI=AN(I)
01177    70 ENERGY=ENERGY-ATENG(ANI)
01178       PRINT80,  ENERGY
01179    80 FORMAT(11X,'  BINDING ENERGY =',F12.4,' A.U.')
01180       IF(PRTOPT.EQ.0)GOTO300
01181       PRINT85
01182    85 FORMAT(//16X,'*VALENCE ELECTRON DENSITY*')
01183       DO 110 I=1,NATOMS
01184       TCHG = 0.E0
01185       LL=LLIM(I)
01186       UL=ULIM(I)
01187       DO 90 J=LL,UL
01188    90 TCHG = TCHG+B(J,J)
01189       ANI=AN(I)
01190       PRINT100,  I,EL(ANI),TCHG
01191   100 FORMAT(I20,A,F15.4)
01192       XXX(I)=TCHG
 1193   110 CONTINUE
01194       DO 120 I=1,3
01195       DM(I)=0.0E0
01196       DMSP(I)=0.0E0
01197   120 DMPD(I)=0.0E0
01198       DO 200 J=1,NATOMS
01199       IF (AN(J).LT.3) GO TO 180
01200       IF (AN(J).LT.11) GO TO 140
01201       SLTR1=(.65*AN(J)-4.95)/3.
01202       FACTOR=2.5416E0*7.E0/(2.2360679775*SLTR1)
01203       INDEX=LLIM(J)
01204       DO 170 K=1,3
01205       INDEXK=INDEX+K
01206   170 DMSP(K)=DMSP(K)-B(INDEX,INDEXK )*10.27175E0/SLTR1
01207       DMPD(1)=DMPD(1)-FACTOR*(B(INDEX+2,INDEX+8)+B(INDEX+3,INDEX+5)
01208      1  +B(INDEX+1,INDEX+7)-1.E0/1.7320508076*B(INDEX+1,INDEX+4))
01209       DMPD(2)=DMPD(2)-FACTOR*(B(INDEX+1,INDEX+8)+B(INDEX+3,INDEX+6)
01210      1  -B(INDEX+2,INDEX+7)-1.E0/1.7320508076*B(INDEX+2,INDEX+4))
01211       DMPD(3)=DMPD(3)-FACTOR*(B(INDEX+1,INDEX+5)+B(INDEX+2,INDEX+6)
01212      1  +2.E0/1.7320508076*B(INDEX+3,INDEX+4))
 1213       GO TO 180
01214   140 INDEX=LLIM(J)
01215       DO 150 K=1,3
01216       INDEXK=INDEX+K
01217   150 DMSP(K)=DMSP(K)-B(INDEX,INDEXK )*7.33697E0/
01218      1(.325*(AN(J)-1))
01219   180 DO 190 I=1,3
01220   190 DM(I)=DM(I)+(CZ(J)-XXX(J))*C(J,I)*2.5416
01221   200 CONTINUE
01222       DP=0.
01223       DO 210 I=1,3
01224       DPM(I)=DM(I)+DMSP(I)+DMPD(I)
01225   210 DP=DP+DPM(I)**2
01226       DP=SQRT(DP)
01227       PRINT220, DM,DMSP,DMPD,DPM,DP
```

```
01228   220 FORMAT(//22X,'DIPOLE MOMENTS'//6X,'COMPONENTS',5X,'X',9X,'Y',9X,'Z
01229      Z'/6X,'DENSITIES',3F10.5/6X,'S.P',6X,3F10.5/6X,'P.D',6X,3F10.5/6X,'
01230       'TOTAL',4X,3F10.5/'0  DIPOLE MOMENT=',F10.5,' DEBYES'//)
01231   300 PRINT9999
01232  9999 FORMAT('0END OF SCF-PROCEDURE'/1X,20('*')/)
 1233       RETURN
01234       END
01235 C
01236 C*****SUBROUTINE OVER1 READ MOLECULAR DATA &
01237       SUBROUTINE OVER1
01238       INTEGER Y,Z
01239       INTEGER DIMA,DIMB
01240       PARAMETER(DIMA=140,DIMB=270)
01241       COMMON    S(DIMB,DIMB),Y(9135),Z(765)
01242       COMMON/C1/V(13),IFALL,KFALL,NV,NVAR,VAR(140),ESV(140),IVAR(140)
01243       COMMON/C7/PRTOPT,AN(DIMA),NATOMS,CHARGE,MULTIP,N,ITER,EPS,LIMEX
01244      1,NROVLY,X(12),XS,STARTX,ENDX,E(12),IST,L1,INK,INKMIN,NST,SCHR,EALT
01245      2,NN1
01246       COMMON/C9/OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN
01247       CHARACTER*4 EL,OPTION,OPNCLO,CNDO,INDO,CLOSED,OPEN,IAN*80,CARD*80
01248       CHARACTER*4 END7
01249       COMMON/C2/C(DIMA,3)
01250       COMMON/C4/EL(18)
01251       COMMON/C3/CZ(DIMA),U(DIMB),ULIM(DIMA),LLIM(DIMA),NELECS,OCCA,OCCB
01252       INTEGER PRTOPT,AN,CHARGE,CZ,U,ULIM,OCCA,OCCB
 1253       REAL YZ(80000)
01254       EQUIVALENCE (YZ,Y)
01255 C
01256 C          * * * * * * * *
01257 C
01258       GOTO(101,100,513,104),NST
01259 C
01260 C**********  NST=1
01261 C
01262 C          1.KARTE: UEBERSCHRIFT  (ODER READ-,WRITE-,END-ANWEISUNG)
01263 C
01264   101 READ(*,20,END=111)  IAN
01265       IF(IAN(1:4).EQ.'READ')GOTO600
01266       IF(IAN(1:5).EQ.'WRITE')GOTO700
01267       IF(IAN(1:2).NE.'*/') GOTO25
01268       IF (STARTX.NE.ENDX) GO TO 111
01269       STOP 10001
01270    25 EALT=1E10
01271 C
01272 C          2.KARTE: SP. 1- 4  OPTION (INDO,CNDO)
 1273 C                   SP. 6- 9  OPNCLO (OPEN,CLSD)
01274 C                   SP.11-14  ANZAHL DER ATOME
01275 C                   SP.15-18  GESAMTLADUNG
01276 C                   SP.22     NEG.EXPONENT VON EPS (GRUNDWERT=1)
01277 C                   SP.23-26  ITERATIONSLIMIT  (GRUNDWERT=60)
01278 C                   SP.27-30  MULTIPLIZITAET  (GRUNDWERT=1)
01279 C
01280       READ20, CARD
01281    20 FORMAT(A)
01282       IF(CARD(1:4).NE.'    ')THEN
01283       IF(CARD(1:3).EQ.'VAR')GOTO 205
01284       OPTION=CARD(1:4)
01285       ENDIF
```

```
01286       IF(CARD(6:9).NE.'    ')OPNCLO=CARD(6:9)
01287       READ(CARD(11:),'(5I4)')NATOMS,CHARGE,LIMEX,ITER,MULTIP
01288 C     IF (NATOMS.GT.DIMA) CALL SYSTEM(52,'0*** ILLEGAL INPUT (NATOMS) **
01289 C    **')
01290       IF(LIMEX.EQ.0) LIMEX=5
01291       EPS=10.**(-LIMEX)
01292       IF(ITER.EQ.0) ITER=65
 1293       NROVLY=3
01294       IF (OPNCLO.EQ.OPEN) GO TO 120
01295       NROVLY=4
01296       MULTIP=1
01297 C     IF (OPNCLO.NE.CLOSED) CALL SYSTEM(52,'0*** ILLEGAL INPUT (OPEN/CLS
01298 C    SD) ***')
01299   120 IF(MULTIP.EQ.0) MULTIP=1
01300 C
01301 C          3.KARTE: LADUNGEN DER ATOME
01302 C
01303       READ40,  (AN(I),I=1,NATOMS)
01304    40 FORMAT(36I2/36I2)
01305       J=9
01306       IF (OPTION.EQ.INDO) GO TO 41
01307       J=18
01308 C     IF (OPTION.NE.CNDO) CALL SYSTEM(52,'0*** ILLEGAL INPUT (CNDO/INDO)
01309 C     ) ***')
01310    41 DO 4 I=1,NATOMS
01311       IF (AN(I).LE.0) AN(I)=1
01312       IF (AN(I).LE.J) GO TO 4
 1313       PRINT3,  OPTION,EL(J)
01314     3 FORMAT('0 **** THIS PROGRAM DOES NOT DO ',A,' CALCULATIONS FOR MOL
01315      LECULES CONTAINING ELEMENTS HIGHER THAN',A)
01316 C     CALL SYSTEM(52,'0 **** EXIT')
01317     4 CONTINUE
01318 C
01319 C          4.KARTE: SP. 1-10  STARTX
01320 C                   SP.11-20  ENDX
01321 C                   SP.21-24  INKREMENT
01322 C                   SP.25-28  MINIMALES INKREMENT  (GRUNDWERT=1)
01323 C                   SP.29-30  STEUERVARIABLE  (GRUNDWERT=1)
01324 C                   SP.31-40  ENERGIESCHRANKE FUER VARIATION
01325 C
01326       READ20, CARD
01327       IF(CARD(1:3).EQ.'VAR')GOTO205
01328       READ(CARD,50) STARTX,ENDX,INK,INKMIN,IST,SCHR
01329       GOTO 210
01330    50 FORMAT(2F10.5,2I4,I2,F10.7)
01331 C
01332 C        ALTERNATIVE PARAMETERKARTE:
 1333 C          SP. 1- 3  'VAR'
01334 C          SP. 4- 5  STEUERVARIABLE
01335 C          SP. 6-15  STARTX
01336 C          SP.16-25  ENDX
01337 C          SP.26-29  INKREMENT
01338 C          SP.30-33  MINIMALES INKREMENT
01339 C          SP.34-43  ENERGIESCHRANKE
01340 C
01341   205 READ(CARD(4:),206)IST,STARTX,ENDX,INK,INKMIN,SCHR
01342   206 FORMAT(I2,2F10.5,2I4,F10.2)
01343   210 IF (IST.LE.0)IST=1
```

```
01344 C     IF (IST.GT.NV) CALL SYSTEM(52,'0*** NR. OF VARIABLE !')
01345       NVAR=NVAR+1
01346       IF (NVAR.LE.100) GO TO 215
01347       NVAR=0
01348       REWIND 7
01349       WRITE(7,*) '''',OPTION,''' ''',OPNCLO,''' ',NATOMS,CHARGE,LIMEX,
01350      ,ITER,MULTIP,NV,NVAR,IFALL,KFALL,(AN(I), I=1,NATOMS),
01351      ,(V(I), I=1,NV),(IVAR(I),VAR(I),ESV(I), I=1,NVAR),' ''*END'''
01352       REWIND 7
 1353       PRINT214
01354   214 FORMAT('0'/'0 *** MORE THAN 30 VARIATIONS ***')
01355       NVAR=100
01356   111 PRINT112,  (I,V(I), I=1,NV)
01357   112 FORMAT('1'/'0+++ LIST OF VARIABLES +++'/'0  VARIABLE    VALUE'//(I
01358      I8,F14.5))
01359       PRINT113,  (I,IVAR(I),VAR(I),ESV(I), I=1,NVAR)
01360   113 FORMAT('0'/'0+++ VALUES OF VARIATIONS +++'/'0  NR      VARIABLE
01361      1VALUE       ENERGY'//(I5,I11,F13.5,F14.7))
01362       PRINT114
01363   114 FORMAT('0',60('='))
01364       STOP
01365   215 IVAR(NVAR)=IST
01366       IF (STARTX+0.0019999999*INK.GT.ENDX) GOTO53
01367       IF(ABS(ENDX-STARTX).LE.INK*0.0101)GOTO55
01368 C  53 CALL SYSTEM(52,'0*** ILLEGAL INPUT (STARTX,ENDX,INK) ***')
01369    53 CONTINUE
01370 C
01371 C          READ-ANWEISUNG:
01372 C            SP. 1- 4  'READ'
 1373 C            SP. 5- 6  ANZAHL DER VARIABLEN  (WENN WERT 0 DANN WERDEN
01374 C                      DIE WERTE VOM FILE GELESEN, SONST MUESSEN SIE
01375 C                      UNMITTELBAR AUF DEN NAECHSTEN KARTEN FOLGEN)
01376 C            SP. 7- 8  IFALL
01377 C            SP. 9-10  KFALL
01378 C
01379   600 READ(IAN(5:),'(3I2)') NV,IFALL,KFALL
01380       IF (NV.EQ.0) GO TO 630
01381 C     IF (NV.GT.(13)) CALL SYSTEM(52,'0*** NV.GT.13 !')
01382       READ620,  (V(I),I=1,NV)
01383   620 FORMAT(F20.17)
01384       GOTO 650
01385   630 REWIND 7
01386       READ(7,*) OPTION,OPNCLO,NATOMS,CHARGE,LIMEX,ITER,MULTIP,NV,NVAR,
01387      ,IFALL,KFALL,(AN(I), I=1,NATOMS),(V(I), I=1,NV),
01388      ,(IVAR(I),VAR(I),ESV(I), I=1,NVAR),END7
01389       REWIND 7
01390 C     IF(END7.NE.'*END') CALL SYSTEM(52,'0*** WRONG FORMAT ON TAPE7')
01391       EPS=10.**(-LIMEX)
01392       NROVLY=3
 1393       IF (OPNCLO.EQ.CLOSED) NROVLY=4
01394   650 PRINT112,  (I,V(I), I=1,NV)
01395       GOTO101
01396 C
01397 C          WRITE-ANWEISUNG:
01398 C            SP.1-5  'WRITE'
01399 C
01400   700 REWIND 7
01401       WRITE(7,*) '''',OPTION,''' ''',OPNCLO,''' ',NATOMS,CHARGE,LIMEX,
```

```
01402       ,ITER,MULTIP,NV,NVAR,IFALL,KFALL,(AN(I), I=1,NATOMS),
01403       ,(V(I), I=1,NV),(IVAR(I),VAR(I),ESV(I), I=1,NVAR),' ''*END'''
01404       REWIND 7
01405       GOTO 101
01406 C
01407 C          FORTSETUNG DER EINGABE
01408 C
01409    55 IF (SCHR.LT.EPS) SCHR=EPS
01410       IF(INKMIN.EQ.0)INKMIN=1
01411 C
01412 C********** NST=2
 1413 C
01414   100 X(1)=STARTX
01415       L1=1
01416       PRTOPT=0
01417       IF (STARTX.NE.ENDX) GO TO 103
01418       IF (STARTX.EQ.0.0) X(1)=V(1)
01419 C
01420 C********** NST=4
01421 C
01422   104 PRTOPT=2
01423       NST=4
01424       VAR(NVAR)=X(1)
01425   103 IF (NST.NE.2)  PRINT7
01426     7 FORMAT('1')
01427       PRINT60, (IAN,I=1,4)
01428    60 FORMAT('0'/'0'/4(/'+',35X,A))
01429       PRINT80,  OPTION,OPNCLO,NATOMS,CHARGE,LIMEX,ITER,MULTIP
01430    80 FORMAT('0',A,1X,A,I4,' ATOMS    CHARGE=',I2,4X,'ENERGY CONSISTENCY
01431      Y LIMIT=1.0E-',I2.2,4X,'ITERATION LIMIT=',I4,4X,'MULT=',I2)
01432       IF (NST.GT.3) GO TO 513
 1433       PRINT512,  STARTX,ENDX,INK,INKMIN,SCHR
01434   512 FORMAT('0STARTX=',F9.4,4X,'ENDX=',F9.4,5X,'INK=',I4,5X,'INKMIN=',I
01435      I4,5X,'DIFFLIM=',1PE7.1)
01436 C
01437 C********** NST=3
01438 C
01439   513 XS=X(L1)
01440       CALL KOORD(XS,IST)
01441       PRINT514, IST,XS
01442   514 FORMAT('0'/'0'/I10,'. VARIABLE',7X,2HX=,F10.5/'0',20X,'COORDINATES
01443      S'/10X,3H*X*,12X,3H*Y*,12X,3H*Z*)
01444       DO 10 I=1,NATOMS
01445       J=AN(I)
01446       PRINT70, EL(J),C(I,1),C(I,2),C(I,3)
01447    70 FORMAT(A,3F15.7)
01448 C      CONVERSION OF COORDINATES FROM ANGSTROM TO AT.
01449       DO 9 J=1,3
01450     9 C(I,J)=C(I,J)/.529167E0
01451    10 CONTINUE
01452 C     Y(I)=0                 I=1,9135
 1453 C     Z(I)=0                 I=1,765
01454       DO 2 I=1,80000
01455     2 YZ(I)=0.0
01456 C     LOAD NON-ZERO Y COEFFICIENTS
01457       Y(7039)=    64
01458       Y(7040)=    64
01459       Y(7049)=   -64
```

```
01460      Y(7032)= -128
01461      Y(7041)=  -64
01462      Y(7033)= -128
01463      Y(7042)=  128
01464      Y(7025)=   64
01465      Y(7034)=  128
01466      Y(7026)=   64
01467      Y(7035)=  -64
01468      Y(7027)=  -64
01469      Y(6904)=  -96
01470      Y(6913)=   32
01471      Y(6896)= -192
01472      Y(6905)=  192
 1473      Y(6906)=  288
01474      Y(6915)=  -96
01475      Y(6889)=  192
01476      Y(6907)= -192
01477      Y(6890)=   96
01478      Y(6899)= -288
01479      Y(6891)= -192
01480      Y(6900)=  192
01481      Y(6892)=  -32
01482      Y(6901)=   96
01483      Y(2854)=  -16
01484      Y(2863)=   16
01485      Y(2847)=   32
01486      Y(2856)=  -16
01487      Y(2865)=  -16
01488      Y(2840)=  -16
01489      Y(2849)=  -16
01490      Y(2858)=   32
01491      Y(2842)=   16
01492      Y(2851)=  -16
 1493      Y(2710)=   48
01494      Y(2719)=  -48
01495      Y(2711)=   48
01496      Y(2720)=  -96
01497      Y(2729)=   48
01498      Y(2703)=  -48
01499      Y(2712)=  -48
01500      Y(2721)=   96
01501      Y(2704)=  -48
01502      Y(2713)=   48
01503      Y(2722)=   48
01504      Y(2731)=  -48
01505      Y(2705)=   96
01506      Y(2714)=  -48
01507      Y(2723)=  -48
01508      Y(2706)=   48
01509      Y(2715)=  -96
01510      Y(2724)=   48
01511      Y(2707)=  -48
01512      Y(2716)=   48
 1513      Y(5329)=   64
01514      Y(5322)= -128
01515      Y(5340)=  -64
01516      Y(5315)=   64
01517      Y(5333)=  128
```

```
01518      Y(5326)=  -64
01519      Y(5185)=  -96
01520      Y(5194)=   32
01521      Y(5186)=  -96
01522      Y(5195)=   64
01523      Y(5204)=   32
01524      Y(5178)=   96
01525      Y(5187)=   32
01526      Y(5196)=   64
01527      Y(5179)=   96
01528      Y(5188)=  -32
01529      Y(5197)=   32
01530      Y(5206)=  -96
01531      Y(5180)=  -64
01532      Y(5189)=  -32
 1533      Y(5198)=  -96
01534      Y(5181)=  -32
01535      Y(5190)=  -64
01536      Y(5199)=   96
01537      Y(5182)=  -32
01538      Y(5191)=   96
01539      Y(4375)= -144
01540      Y(4384)=   96
01541      Y(4393)=  -16
01542      Y(4368)=  144
01543      Y(4386)=  -48
01544      Y(4395)=   96
01545      Y(4370)=  -96
01546      Y(4379)=   48
01547      Y(4397)= -144
01548      Y(4372)=   16
01549      Y(4381)=  -96
01550      Y(4390)=  144
01551      Y(1900)=  144
01552      Y(1909)= -144
 1553      Y(1893)= -144
01554      Y(1920)=  144
01555      Y(1895)=  144
01556      Y(1922)= -144
01557      Y(1906)= -144
01558      Y(1915)=  144
01559      Y( 955)=  -16
01560      Y( 964)=   32
01561      Y( 973)=  -16
01562      Y( 948)=   16
01563      Y( 966)=  -48
01564      Y( 975)=   32
01565      Y( 950)=  -32
01566      Y( 959)=   48
01567      Y( 977)=  -16
01568      Y( 952)=   16
01569      Y( 961)=  -32
01570      Y( 970)=   16
01571      Y(8155)=   64
01572      Y(8156)=  -64
 1573      Y(8165)=  -64
01574      Y(8148)=  -64
01575      Y(8157)=   64
```

```
01576      Y(8149)=   64
01577      Y(8158)=   64
01578      Y(8150)=  -64
01579      Y(8020)=  -96
01580      Y(8029)=   32
01581      Y(8021)=  128
01582      Y(8013)=   96
01583      Y(8031)=  -96
01584      Y(8014)= -128
01585      Y(8015)=  -32
01586      Y(8024)=   96
01587      Y(7084)=  -64
01588      Y(7076)= -128
01589      Y(7085)=   64
01590      Y(7086)=  128
01591      Y(7069)=  128
01592      Y(7070)=   64
 1593      Y(7079)= -128
01594      Y(7071)=  -64
01595      Y(3205)=  -16
01596      Y(3214)=   16
01597      Y(3206)=   16
01598      Y(3215)=  -16
01599      Y(3198)=   16
01600      Y(3216)=  -16
01601      Y(3199)=  -16
01602      Y(3217)=   16
01603      Y(3200)=  -16
01604      Y(3209)=   16
01605      Y(3201)=   16
01606      Y(3210)=  -16
01607      Y(7579)=   64
01608      Y(7580)=  -64
01609      Y(7572)= -128
01610      Y(7573)=  128
01611      Y(7565)=   64
01612      Y(7566)=  -64
 1613      Y(5680)=   64
01614      Y(5681)=  -64
01615      Y(5673)=  -64
01616      Y(5691)=  -64
01617      Y(5674)=   64
01618      Y(5692)=   64
01619      Y(5684)=   64
01620      Y(5685)=  -64
01621      Y(7435)=  -96
01622      Y(7444)=   32
01623      Y(7436)=  -96
01624      Y(7445)=  160
01625      Y(7428)=   96
01626      Y(7437)=  128
01627      Y(7446)=  -96
01628      Y(7429)=   96
01629      Y(7438)= -128
01630      Y(7447)=  -96
01631      Y(7430)= -160
01632      Y(7439)=   96
 1633      Y(7431)=  -32
```

```
01634      Y(7440)=   96
01635      Y(5545)=  -96
01636      Y(5554)=   32
01637      Y(5546)=   32
01638      Y(5555)=   32
01639      Y(5538)=   96
01640      Y(5556)=   32
01641      Y(5539)=  -32
01642      Y(5557)=  -96
01643      Y(5540)=  -32
01644      Y(5549)=  -32
01645      Y(5541)=  -32
01646      Y(5550)=   96
01647      Y(3070)=   48
01648      Y(3079)=  -48
01649      Y(3071)=  -48
01650      Y(3080)=   48
01651      Y(3063)=  -48
01652      Y(3081)=   48
 1653      Y(3064)=   48
01654      Y(3082)=  -48
01655      Y(3065)=   48
01656      Y(3074)=  -48
01657      Y(3066)=  -48
01658      Y(3075)=   48
01659      Y(8200)=  -64
01660      Y(8201)=   64
01661      Y(8193)=   64
01662      Y(8194)=  -64
01663      Y(7615)=  -64
01664      Y(7616)=  -64
01665      Y(7625)=   64
01666      Y(7608)=   64
01667      Y(7617)=   64
01668      Y(7609)=   64
01669      Y(7618)=  -64
01670      Y(7610)=  -64
01671      Y(3250)=   16
01672      Y(3259)=  -16
 1673      Y(3243)=  -16
01674      Y(3261)=   16
01675      Y(3245)=   16
01676      Y(3254)=  -16
01677      Y(5725)=  -64
01678      Y(5718)=   64
01679      Y(5736)=   64
01680      Y(5729)=  -64
01681      Z(341)=    -1
01682      Z(343)=     3
01683      Z(345)=    -3
01684      Z(347)=     1
01685      Z(664)=    -1
01686      Z(665)=     5
01687      Z(666)=   -10
01688      Z(667)=    10
01689      Z(668)=    -5
01690      Z(669)=     1
01691      Z(154)=    -1
```

```
01692      Z(156)=    5
 1693      Z(158)=   -10
01694      Z(160)=   10
01695      Z(162)=   -5
01696      Z(164)=    1
01697      Z(222)=   -1
01698      Z(223)=    1
01699      Z(224)=    4
01700      Z(225)=   -4
01701      Z(226)=   -6
01702      Z(227)=    6
01703      Z(228)=    4
01704      Z(229)=   -4
01705      Z(230)=   -1
01706      Z(231)=    1
01707      Z(307)=   -1
01708      Z(308)=    2
01709      Z(309)=    2
01710      Z(310)=   -6
01711      Z(312)=    6
01712      Z(313)=   -2
 1713      Z(314)=   -2
01714      Z(315)=    1
01715      Z(409)=   -1
01716      Z(410)=    3
01717      Z(411)=   -1
01718      Z(412)=   -5
01719      Z(413)=    5
01720      Z(414)=    1
01721      Z(415)=   -3
01722      Z(416)=    1
01723      Z(528)=   -1
01724      Z(529)=    4
01725      Z(530)=   -5
01726      Z(532)=    5
01727      Z(533)=   -4
01728      Z(534)=    1
01729      Z(562)=   -1
01730      Z(563)=    2
01731      Z(565)=   -2
01732      Z(566)=    1
 1733      Z(732)=   -1
01734      Z(733)=    1
01735      Z(545)=    1
01736      Z(546)=   -3
01737      Z(547)=    2
01738      Z(548)=    2
01739      Z(549)=   -3
01740      Z(550)=    1
01741      Z(579)=    1
01742      Z(580)=   -1
01743      Z(581)=   -1
01744      Z(582)=    1
01745      Z(596)=   -1
01746      Z(598)=    1
01747      Z(443)=   -1
01748      Z(444)=    1
01749      Z(445)=    2
```

```
 1750       Z(446)=    -2
01751       Z(447)=    -1
01752       Z(448)=     1
01753        Z(698)=    -1
01754       Z(699)=     3
01755       Z(700)=    -3
01756       Z(701)=     1
01757       Z(324)=     1
01758       Z(325)=    -1
01759       Z(326)=    -3
01760       Z(327)=     3
01761       Z(328)=     3
01762       Z(329)=    -3
01763       Z(330)=    -1
01764       Z(331)=     1
01765       Z(460)=     1
01766       Z(462)=    -2
01767       Z(464)=     1
01768       RETURN
01769       END
 1770 /*
01771 //GO.FT07F001  DD
01772 //GO.SYSIN DD *
01773  MEFLOQUINE***MEF120*****
01774 CNDO CLSD   42         100
01775 07060606060606060606060601010101060909090909090608010106010701060101060101
01776 060101060101
01777 1.200        1.200   000
01778     4.87784  -2.63411   0.00000
01779     5.06297  -1.31705   0.00000
01780     3.99051  -0.44857   0.00000
01781     2.68349  -0.95028   0.00000
01782     1.24185  -2.90004   0.00000
01783     1.11085  -4.36747   0.00000
01784     2.22178  -5.11681   0.00000
01785     3.47333  -4.55957   0.00000
01786     3.59884  -3.12506   0.00000
01787     2.43819  -2.34218   0.00000
01788     6.48200  -0.77231  -0.00001
01789     4.16008   0.62209   0.00000
 1790     0.35390  -2.27827   0.00000
01791     0.13657  -4.84268   0.00000
01792     2.14615  -6.19818   0.00000
01793     4.76237  -5.36508   0.00000
01794     5.13548  -5.59824   1.24453
01795     4.56433  -6.51223  -0.62224
01796     5.70662  -4.68424  -0.62224
01797     6.89275  -0.61464   1.24452
01798     7.27898  -1.62082  -0.62225
01799     6.50651   0.39154  -0.62225
01800     1.51000  -0.00001   0.00000
01801     0.71488  -0.22339   1.16738
01802    -0.04668   0.37698   1.18937
01803     0.91505  -0.16715  -0.87350
01804     2.02407   1.45165   0.00000
01805     2.61868   1.62000   0.87349
01806     2.84652   1.68450  -1.20820
01807     3.63207   1.04967  -1.20878
```

```
01808     3.34056   3.07961  -1.20819
01809     3.93517   3.24795  -2.08169
 1810     3.93517   3.24795  -0.33470
01811     2.14376   4.04877  -1.20820
01812     1.54882   3.88162  -2.08169
01813     2.50094   5.05739  -1.20820
01814     1.28748   3.80820   0.04898
01815     1.88142   3.97570   0.92309
01816     0.45594   4.48157   0.04898
01817     0.77196   2.35704   0.04987
01818     0.17735   2.18870  -0.82362
01819     0.17701   2.18989   0.92336
01820 /*
01821 //
01822 * * * END OF FILE * * *
```

APPENDIX III

Linear Regression Procedure

For the statistical precedure, we are mainly work on linear regression fitting. The REGRESSION procedure can be used to study the relationship between a dependent variable and a set of independent variables and regression coefficients evaluate how well the model fits.

We used the Multiple Regression Analysis of SPSS batch system that obtained by Computor Center of Chulalongkorn University. Procedure REGRESSION is driven by a series of subcommand sets. Each subcommand set groups together logically related operations or definitions and is relatively independent of the specifications used on another subcommand.

To work with SPSS, we must set the input data file to construct our own procedure by any COMMAND within the SPSS system. Some important example are brifly described as followed:

INPUT FILE FOR SPSS: Have some important parts that are,

JCL COMMAND depends on the system we used. Define also EXEC=SPSS.

FILE HANDLE wording after FILE HANDLE will be written in the output files for any comments.

DATA LIST do in the same way as FORMAT in FORTRAN. We can start our data by the first card (/1) followed by name of data and the field of data which defined the exact format in the blanket. Each card has only 72 columns and we can continue the data by second card (/2) and so on. Number of cards must be specified by RECORD, that is RECORD=2 for using two cards to fill input data.

BEGIN DATA fill data according to the format in data list and follow END DATA after the last value.

COMMANDS AND SUBCOMMANS

the Regression procedure in SPSS is operated by using the REGRESSION command and its subcommans. Each subcommand must have slash (/) between two of them. Many subcommands in SPSS allows to write cur own procedure within the SPSS system.

FINISH end of input file.

EXAMPLE OF SPSS INPUT DATA FILE

```
 0001 //BIAJ0053     JOB  CLASS=I,MSGCLASS=R,MSGLEVEL=(1,1)
00002 //             EXEC SPSSX,A=R
00003 FILE HANDLE ABINITIOALLDATAMODEL53
00004 DATA LIST FIXED RECORDS=2
00005     /1 A 1-5(3) N1 7-13(4) N2 15-21(4) C2 23-29(4) C3 31-36(4)
00006        C4 38-44(4) C9 46-52(4) C10 54-60(4) ORING 62-68(4)
00007     /2 U 1-5(4) H22 7-13(4) Q23 15-21(4) Q58 23-29(4) Q910 31-37(4)
00008 BEGIN DATA
00009 4.605  0.3909  0.4446 -0.0675  0.4414 -0.2347 -0.2389  0.6533  1.7998
00010 3.881 -0.3276  0.3739  0.8552  0.4144
00011 4.605  0.3909  0.4449 -0.0673  0.4412 -0.2346 -0.2391  0.6531  1.7991
00012 3.995 -0.3286  0.3739  0.8550  0.4140
00013 3.912  0.3907  0.4432 -0.0669  0.4406 -0.2336 -0.2385  0.6523  1.7988
00014 4.333 -0.3284  0.3737  0.8543  0.4138
00015 4.605  0.4351  0.5440 -0.1277  0.4955 -0.3669 -0.2359  0.6576  1.6940
00016 5.094 -0.3594  0.3678  0.8363  0.4217
00017 3.912  0.4349  0.5440 -0.1279  0.4958 -0.3669 -0.2360  0.6577  1.6937
00018 5.086 -0.3597  0.3679  0.8362  0.4217
00019 4.382  0.6589  0.6033  0.2565  0.7802 -0.1500  0.0109  0.7108  4.4500
00020 1.593 -0.2267  1.0385  2.1758  0.6999
 0021 2.079  0.4352  0.5445 -0.1283  0.4959 -0.3668 -0.2361  0.6578  1.6941
00022 5.694 -0.3601  0.3676  0.8363  0.4217
00023 END DATA
00024 REGRESSION DESCRIPTIVES=DEFAULTS VARIANCE SIG COV XPROD/
00025     VARIABLES=A N1 C2 C3 C4 C9 C10 N2
00026    /STATISTICS=DEFAULTS ZPP F
00027    /DEPENDENT=A
00028    /METHOD=ENTER
00029    /RESID=DEFAULT
00030    /CASEWISE=DEFAULT ALL SRE MAH COOK SDR
00031    /SCATTERPLOT (A,*PRED)
00032 FINISH
00033 /*
00034 //
00035 * * * END OF FILE * * *
```

VITA

Miss Supa Polman was born on April 9, 1964. She received her B.Sc. in Chemistry from Chulalongkorn University in 1986. She further studied in the Graduate School of Chulalongkorn University, with the major in Physical Chemistry. One publication and two submitted for publication have come out of her master's degree work. That is Quantum Pharmacological Studies on Primaquine Antimalarial Activity, Analytical Science, forthcoming. The two submitted for publication are Quantum Pharmacological Studies on Antimalarial Drugs and Quantum Pharmacological Studies on Mefloquine Antimalarial Drugs.

During this study, she received a Teacher Assistantship from Graduate School in 1986, Somdetpra Mahitalathibeth Praboorommarachanok Funds between 1987-1988 and Dr. Buaret Khamthong Funds for study support in 1987. She also received the Phumipol Funds awards from the King in 1987 and 1988 for essay competitions of the university.