Order Number 9108082

The iso-electronic method for locating transition states on the MNDO energy surface

Brennan, Peter Lester, Ph.D.

City University of New York, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

The Iso-Electronic Method For Locating Transition States On The MNDO Energy Surface

by

Peter Brennan

A dissertation submitted to the Graduate Faculty in Chemistry
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy.
The City University of New York.

1990

This manuscript has been read and accepted for the Graduate Faculty in Chemistry in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

May 18, 1990 — Thomas A. Halgren
Date — Chair of Examining Committee

5/31/90 — [signature]
Date — Executive Officer

Dr. D. Lindsay

Dr. N. McKelvie

Dr. J. Dannenberg
Supervisory Committee

The City University of New York

Abstract

The Iso-Electronic Method For Locating Transition States
On The MNDO Energy Surface
by
Peter Brennan

Adviser: Professor Tom A. Halgren

A new method of locating transition states on the MNDO energy surface is presented. It is an extension of Synchronous Transit, which was developed for PRDDO. Search direction vectors are generated from a hessian matrix which is updated after each search by one of three different methods. The Coupled-Perturbed-Hartree-Fock equations are solved in order to render each search direction iso-electronic, meaning that there is little or no net change in the bonding along the vector. Synchronous Transit is used to generated the starting point, and to find local maximums periodically throughout the optimization. For small molecules and/or on reasonably quadratic surfaces, the new approach works very well. On non-quadratic surfaces, convergence can still be achieved by subdividing the path through the use of flanking points, although far more computational effort may be required. Results are presented for a set of 15 simple reactions for which the Iso-Electronic Method has been applied.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

INTRODUCTION

The location of transition states by theoretical means has received a great deal of attention in recent years. There are several methods in current use. The primary ones will be outlined here.

GRID SEARCH

The grid search[1-3] is one of the earliest methods for locating transition states. The region of the transition state is mapped out with a grid of points. Each internal variable is incremented over a given range, and the energy is evaluated for each point. The transition state is found either by inspection or by polynomial fit. While this method can be used for very small molecules, the number of energy evaluations required grows very quickly with the size of the system. A system of n atoms contains 3n-6 internal variables, each of which would have to be incremented at least 10 times for reasonable precision, requiring 10^{3n-6} energy evaluations. For a system of 3 atoms, one thousand points would be required. For 4 atoms, one million points. Each additional atom increases the number of points by a factor of one thousand. It quickly becomes obvious that the grid search method cannot be used as a general technique.

REACTION COORDINATE

The reaction coordinate[4-7] approach is one of the most common methods for locating transition states. A particular internal variable is chosen as the reaction coordinate. This can be a bond length which breaks, or a bond angle or dihedral angle which changes significantly during, and is central to, the reaction. The reaction coordi-

nate is held fixed while all the other internal variables are freely optimized. The reaction coordinate is then incremented, and the other internal variables are again optimized. The maximum energy structure obtained from this method is taken as the transition state. This method can be quite tedious, as many different values of the reaction coordinate must be considered, each requiring the optimization of all other internal variables. This can also become expensive in terms of computing time, though not as costly as the grid search method. It might be thought that the other points calculated during the procedure would give information about the reaction pathway, but this information cannot be trusted. The path arrived at is often discontinuous[8] and can be very different if a different internal variable is chosen as the reaction coordinate.

GRADIENT NORM MINIMIZATION

Another method in widespread use is the minimization of the square of the gradient norm. McIver and Komornicki[9] proposed this method for locating transition states using the semiempirical program MINDO/2, but it is also currently used in ab initio programs as well. It consists of minimizing the square of the gradient vector by a standard least squares minimization routine, such as that of Peckham.[10] Again, this method can be very expensive in computer time, since the Hessian matrix must be evaluated at the start of the procedure, requiring either $(n+1)^2$ function or n gradient evaluations, and updated at least periodically during the optimization. Prior knowledge of the energy surface, or a close estimate of the structure of the transition state, is required for use as a starting point. The principle disadvantage of this method is that there is no guarantee that the transition state will be found. Any point of zero gradient

on the energy surface can be found, such as local maximums or local minimums. At the point of convergence, the Hessian must be reevaluated, and the eigenvalues must be computed. At a saddle point, there will be one and only one negative eigenvalue. If this is not the case, then the entire procedure will have to be repeated, starting from a different point. And there is still no guarantee that the transition state will be found.

QUASI-NEWTON METHODS

The quasi-Newton methods[11] form the most efficient class of minimization techniques used for finding stable geometries. An iteration consists of generating a search direction D according to $D=-HG$, where H is an approximation to the inverse Hessian matrix, and G is the gradient vector. The Hessian is initialized at the start of the calculation by taking a step in the direction of the gradient. A diagonal Hessian matrix is computed from the change in gradient. This Hessian is then updated after each iteration by one of several methods, such as the Fletcher,[12] the Powell,[13] or the Complimentary Davidon-Fletcher-Powell[14, 15] updating schemes. These methods perform a linear search for a minimum along the vector D according to $x^{k+1} = x^k + \alpha^k D^k$.

Pulay[16] and Schlegel[17] have had some success in adapting the quasi-Newton methods to finding saddle points. The primary difference is to search to minimize the gradient, rather than the functional value. The principle problems seem to be as follows. If a direction D^k is generated which has almost zero gradient, a very long steplength will result, giving a meaningless value for x^{k+1}. The major problem, however, is with the Hessian. If the Hessian becomes positive definite, the search will lead towards a minimum. In order to find a saddle point, there must be one negative

eigenvalue, but there is no way to prevent the Hessian from becoming positive definite. Pulay[16] has suggested starting the calculation with a crude grid search to locate the general area of the transition state, but this will greatly add to the cost of the calculation. Even with this, the Hessian could still become positive definite and the procedure could find a local minimum.

SYNCHRONOUS TRANSIT

This method[8] was first proposed by Halgren and Lipscomb for use in the approximate ab initio program PRDDO.[18] First, the reactant and product structures are optimized, and a constraint axis vector is formed by subtracting the cartesian coordinates of the reactant from those of the product. A more recent version[19] forms this vector from the difference of the full (overcomplete) sets of internal variables, comprised of all bond lengths, bond angles, and dihedral angles in the reactant and product structures. The Linear Synchronous Transit (LST) energy maximum is found along this vector, and it is used as the starting point for the search. The set of optimization modes and the order in which they will be varied is determined and input by the user. In order to prevent the search from finding either the reactant or product, the search directions to minimize the energy are first orthogonalized to the constraint axis. Following this, the Quadratic Synchronous Transit (QST) energy maximum is found along the path connecting the reactant, the product, and the current estimate for the transition state. This procedure of alternating maximum and minimum searches is repeated until no further progress can be made. Since the resultant structure is simultaneously a maximum and a minimum, it should be the transition state of the reaction.

A major problem of this method is that successive path maximums can oscillate from one side of the transition state to the other, producing an infinite loop. Another problem is that it is difficult to determine exactly what structure the procedure has converged on. Since the gradient is not computed, the criterion of zero gradient cannot be used. Nor is there a Hessian matrix whose eigenvalues can be examined. Also, it is difficult and tedious to determine an appropriate set of optimization modes, except for very small molecules. Many decisions on the part of the user must be made for each computer run concerning the subset of the optimization modes which will be submitted and the order in which they will be varied.

NEGATIVE EIGENVECTOR

Baker has devised a method[20] which he has encorporated into Gaussian 82. An initial estimate is made for the transition state. The gradient vector and hessian matrix are then computed. Gaussian 82 can compute analytical gradients for its SCF, MP2, CID and CISD wave functions, and analytical second derivatives for its SCF wave function. The hessian would have to be computed by taking finite differences of analytical first derivatives for other than the SCF wave functions. Alternatively, the SCF hessian could be used for the starting point. The hessian is then diagonalized, and the eigenvalues are examined. If the hessian has more than one negative eigenvalue, the lowest one can be followed, or the corresponding eigenvectors can be combined into a composite vector. If the hessian has the correct one negative eigenvalue, then either the Newton-Raphson or the Partitioned-Rational-Function-Approximation step can be taken. The vector is then scaled down, if needed, to a user-supplied maximum

length, with a default value of 0.3. The vector is used to update the current estimate, and the gradient is computed for the new point. If the RMS values and maximum components of both the gradient and updating vectors are below the user-supplied tolerances, then the optimization is completed.

CONJUGATE GRADIENTS IN CONJUGATE SPACE

Bell and Crighton [21] have devised this method and incorporated it into a number of different computer programs. As with Synchronous Transit,[8] a maximum is found along a line $p^{(1)}$ known to have negative curvature. A quasi-Newton minimization is then made in the space of (n-1) linearly independent directions which are conjugate to $p^{(1)}$. A test is made of the gradient along $p^{(1)}$, and, if necessary a new maximum is found. The method has been modified somewhat[22] by adjusting the vector $p^{(1)}$ to a parabolic path, including the two endpoints and the current estimate of the saddle point, similar to the QST path in Synchronous Transit.[8]

THE ISO-ELECTRONIC METHOD

A new method of locating transition states on theoretical energy surfaces has been developed as part of this research project. It is based primarily on Synchronous Transit[8] and the Quasi-Newton[11] methods. The MNDO[23] program has been used for the energy and gradient calculations, but it could also be used in any of the other methods for predicting molecular structures, provided an analytical gradient can be computed, and the CPHF equations can be solved to determine the gradient of the wavefunction. On nearly quadratic surfaces, convergence is remarkably quick. On non-

quadratic surfaces, convergence-can still be achieved, although sometimes the partial optimization of other path defining points is required.

Synchronous Transit is used to generate the structure of maximum energy along the constraint axis. In this work, the constraint axis is defined by the difference between the complete sets (normally, 3n-6 in number) of bond lengths, bond angles, and dihedral angles in the reactant and product structures. All intermediate structures are generated by varying this complete set of internal variables. The LST maximum is taken as the initial estimate for the transition state. The analytical gradient is computed, and a step is taken in the direction of the gradient. The change in gradient is used to initialize the inverse Hessian matrix, which is used to generate subsequent optimization vectors. Each optimization vector is adjusted to hold constant position on the constraint axis under the iso-electronic constraint, to minimize drift towards either the reactant or product structures. This change in position is determined by computing the change to the electron distribution in the structure, as reflected in the change to the bond orders between the atoms, rather than the change to the internal variables, as in standard Synchronous Transit. The minimum in energy is found along the new vector by taking a step along the vector and examining the change in energy and gradient. If needed, a longer step is taken, or, if too long a step was taken, an interpolation is made to find the minimum between the endpoints. A limit is placed on the steplength times the individual components of the vector, so that vectors of very low gradient will not generate very large steplengths, and thus move away from the region of the saddle point. The change in gradient is used to update the Hessian, using either the

Fletcher,[12] the Powell,[13] or the Complimentary Davidon-Fletcher-Powell[14, 15] updating scheme. After a set number of these line searches, or if the gradient increases sufficiently, or if there is drift towards either the reactant or product, the QST path maximum is found, which is then further optimized. When both the gradient and the length of the optimization step approach zero, the optimization is completed. The structure converged upon should be the transition state of the reaction, since it has a gradient of essentially zero, and it is, or is very close to, a QST path maximum. However, this must be verified by mapping out the reaction path, to show that the maximum along the path has been found.

THE ANALYTICAL GRADIENT

The gradient is most easily computed with respect to the XYZ coordinates. Dewar and Yamaguchi[24] have derived equations for the XYZ gradient, which will not be repeated here. This must then be transformed into the gradient with respect to the internal variables, as required for the optimization. This requires the gradient of the XYZ coordinates with respect to the internal variables. The method employed by Gaussian 82 was adapted for this. Equations for the transformation are as follows.

Given a matrix q, of the internal variables, and a matrix x, of the XYZ coordinates, there exists a matrix B such that

$$q = x\ B \tag{1}$$

Therefore, there exists a matrix B^{-1} such that

$$x = q\ B^{-1} \tag{2}$$

B^{-1} can be used to transform the XYZ coordinates into the internal variables, or their

respective gradients.

The matrix B consists of the derivatives of the internal variables with respect to the XYZ coordinates of the defining atoms. For bond distance I-J, there are 6 elements. The unit vector $\vec{e}_{IJ}$ comprises the first 3 elements, while the unit vector $\vec{e}_{JI}$ comprises the last 3 elements.

For bond angle I-J-K (θ), there are 9 elements.

$$B_{123} = \frac{COS\,\theta\ \vec{e}_{JI} - \vec{e}_{JK}}{|\vec{JI}|\ SIN\,\theta} \tag{3}$$

$$B_{789} = \frac{COS\,\theta\ \vec{e}_{JK} - \vec{e}_{JI}}{|\vec{JK}|\ SIN\,\theta} \tag{4}$$

Each of the middle 3 elements is equal to minus the sum of the corresponding elements from the first 3 and the last 3 elements.

For dihedral angle I-J-K-L, there are 12 elements. Angle I-J-K will be referred to as θ, while angle J-K-L will be referred to as ω.

$$B_{123} = \frac{\vec{e}_{JI} \times \vec{e}_{JK}}{|\vec{JI}|\ SIN^2\theta} \tag{5}$$

$$B_{10\,11\,12} = \frac{\vec{e}_{KL} \times \vec{e}_{KJ}}{|\vec{KL}|\ SIN^2\omega} \tag{6}$$

$$B_{456} = B_{123}\left(\frac{|\vec{JI}|}{|\vec{JK}|}\ COS\,\theta - 1\right) + B_{10\,11\,12}\frac{|\vec{KL}|}{|\vec{JK}|}\ COS\,\omega \tag{7}$$

Elements 7, 8 and 9 each are equal to minus the sum of the appropriate three of the other 9 elements.

Since B is not a square matrix, a special procedure is used to calculate its inverse. B^{-1} can be determined from the following equation

$$B^{-1} = (B^T M B)^{-1} B^T M \tag{8}$$

where M is the unit matrix of dimension three times the number of atoms, with six zero's on the diagonal. Multiplication by M removes the six degrees of freedom corresponding to translation and rotation of the rigid molecule. The net result is to place the first atom at the origin, the second along the X axis, the third in the XY plane, and the rest as defined by the internal variables.

THE ISO-ELECTRONIC CONSTRAINT

The electronic path coordinate is a new way to evaluate the position on the constraint axis. It is computed from the changes in the bond orders between the atoms, rather than from the changes in the internal variables, as for a geometric path coordinate. This iso-electronic constraint requires the gradient of the wavefunction, computed from the solution of the Coupled-Perturbed-Hartree-Fock (CPHF)[25] equations.

The concept of the electronic path coordinate, P_e, is patterned after that of the geometric path coordinate, P_g, which shall be discussed first. P_g is defined from the RMS change in the internal variables which define the molecules,[19] according to the equation

$$P_g = \frac{P_g(r) + P_g(p)}{2} + \frac{P_g(p) - P_g(r)}{2} \frac{\sum(VR_i - V_i)^2 - \sum(VP_i - V_i)^2}{\sum(VP_i - VR_i)^2} \qquad (9)$$

where VR_i, VP_i, and V_i are the internal variables of the reactants, the products, and the structure for which the path coordinate is being computed. $P_g(r)$ is the path coordinate of the reactant, usually 0.0, and $P_g(p)$ is the path coordinate of the product, usually 1.0. Conformational changes, such as the rotation of a sigma bond, change the dihedral angles (and therefore P_g), but may contribute little to the actual

progress into the chemical reaction. The electronic path coordinate, P_e, is computed from the Armstrong-Perkins-Stewart (APS)[26] bond orders, thus giving a closer measure of chemical changes in the system. P_e is defined in a similar way to P_R, but using the RMS change in the APS bond orders, according to

$$P_e = \frac{P_e(r) + P_e(p)}{2} + \frac{P_e(p) - P_e(r)}{2} \frac{\sum (BR_i - B_i)^2 - \sum (BP_i - B_i)^2}{\sum (BP_i - BR_i)^2} \quad (10)$$

where $P_e(r)$ and $P_e(p)$ are the electronic path coordinates of the reactants and products, and BR_i, BP_i, and B_i are the APS bond orders of the three structures, defined by

$$B_i = \sum_{\mu}^{A} \sum_{\nu}^{B} P_{\mu\nu}^2 \quad . \quad (11)$$

where μ and ν are the atomic orbitals for the A, B atom pair. Note that the summations over i in eq 9 are over the complete set of internal variables which define the structures, while the summations over i in eq 10 are over the bond orders, B_i, for atom pairs $A > B$.

The optimization vector, $\vec{y}$, generated by the hessian should produce a structure of lower energy, but will most likely point in the general direction of either the reactant or the product of the reaction. $\vec{y}$ is orthogonalized to the constraint axis vector, $\vec{z}$, which is formed by subtracting the set of $3n-6$ internal variables of the reactant from those of the product, according to

$$\vec{y}'_i = \vec{y}_i - \vec{y} \frac{(\partial P_e / \partial \vec{y})}{(\partial P_e / \partial \vec{z})} \vec{e}_{z_i} \quad . \quad (12)$$

producing the new search direction $\vec{y}'$, along which P_e should not change. This requires the partial derivatives $(\partial P_e / \partial \vec{y})$ and $(\partial P_e / \partial \vec{z})$. Here we will develop

equations for the $\vec{y}$ derivative. Equations for the $\vec{z}$ derivatives are identical in form. Taking the derivative of eq (10) with respect to $\vec{y}$, we have

$$\frac{\partial P_e}{\partial \vec{y}} = \frac{P_e(p) - P_e(r)}{\sum (BP_i - BR_i)^2} \sum (BP_i - BR_i) \frac{\partial B_i}{\partial \vec{y}} \quad (13)$$

where

$$\frac{\partial B_i}{\partial \vec{y}} = 2 \sum_{\mu}^{A} \sum_{\nu}^{H} P_{\mu\nu} \frac{\partial P_{\mu\nu}}{\partial \vec{y}} \quad (14)$$

The CPHF equations allow us to compute the derivatives $(\partial P_{\mu\nu} / \partial \vec{y})$ and $(\partial P_{\mu\nu} / \partial \vec{z})$ without having to compute the energies of the points resulting from displacements along the vectors $\vec{y}$ and $\vec{z}$.

THE COUPLED PERTURBED HARTREE-FOCK EQUATIONS

Pople, Krishnan, Schlegel and Binkley[25] solved these equations to compute the gradient of the energy for the full ab initio wavefunction as used by Gaussian 80. Here, we will consider a perturbation, such as the displacement of the nuclei in an optimization step, and present the equations for the gradient of the wavefunction, in the specific form of the MNDO approximation.

The following convention will be used to index array elements. Greek letters, μ, ν, λ and σ will indicate atomic orbitals, i and j will represent occupied molecular orbitals, a and b will represent virtual molecular orbitals, and p, q, r and s will represent the full basis set, both occupied and virtual orbitals. Capital letters A and B will represent the atoms in the molecule.

The perturbation to the wavefunction can be expressed as a matrix $U(\vec{y})$, with elements u_{rs}, such that

$$C(\vec{y}) = C(0)\, U(\vec{y}) \tag{15}$$

where $\vec{y}$ represents the perturbation, $C(0)$ is the original coefficient matrix, and $C(\vec{y})$ is the perturbed coefficient matrix, with elements $c_{\mu i}(\vec{y})$. Note that $U(0)$ is the identity matrix. The perturbed density matrix elements can be expressed as

$$P_{\mu\nu}(\vec{y}) = \sum_{i}^{2n} c^{*}_{\mu i}(\vec{y})\, c_{\nu i}(\vec{y}) \tag{16}$$

where $2n$ is the number of occupied spin orbitals, including both α and β spin. Expanding eqs (15) and (16) to first order in $\vec{y}$, substituting into (16) and equating first order terms in $\vec{y}$ gives

$$P^{(1)}_{\mu\nu} = \sum_{i}^{2n}\sum_{r}^{2N} [\, c_{\mu i}(0)\, c_{\nu r}(0) + c_{\mu r}(0)\, c_{\nu i}(0)\,]\, u^{(1)}_{ri} \,. \tag{17}$$

where $P^{(1)}_{\mu\nu}$ represents the derivative of density matrix element $\mu\nu$ with respect to a displacement along the vector $\vec{y}$, and $2N$ is the total number of spin orbitals. Pople, et al. have derived the following equation

$$u^{(1)}_{qp} = \frac{F^{(1)}_{qp} - S^{(1)}_{qp}\,\epsilon_p(0)}{\epsilon_p(0) - \epsilon_q(0)} \tag{18}$$

where $F^{(1)}_{qp}$ is an element of the first order expansion of the Fock matrix, $S^{(1)}_{qp}$ is the first order expansion of the overlap integral of spin orbitals χ_q and χ_p, and ϵ_p is the eigenvalue of the spin orbital χ_p. In MNDO, the overlap integrals in the atomic orbital basis set are equal to the kronical δ, and so $S^{(1)} = 0$. The $\Delta\epsilon$ term makes $U^{(1)}$ antisymmetric, and so, in eq (17), the occupied-occupied block sums to zero. The occupied-virtual elements of the Fock matrix can be expressed

$$F_{ai} = H_{ai} + \sum_{j}^{2n}\sum_{rs}^{2N} u^{*}_{rj}\, u_{sj}\, (a\ r \mid\mid i\ s) \,. \tag{19}$$

where $(a\ r \mid\mid i\ s)$ is the notation for the electron repulsion integrals

$$(a\,r \quad i\,s) = \int\int \chi_a^*(1)\,\chi_r^*(2)\,\frac{1}{r_{12}}\,[\,\chi_i(1)\,\chi_s(2) - (\,\chi_i(2)\,\chi_s(1)\,]\;d\tau_1\,d\tau_2$$

$$= (a\,i \quad r\,s) - (a\,s \quad r\,i) \tag{20}$$

and $H_{\mu\mu}$ represents the two-center one-electron core resonance integrals and nuclear attraction integrals.

$$H_{\mu\mu} = \sum_A \sum_\mu^A c_{\mu a} \left\{ \sum_\nu^A c_{\nu i}\, U_{\mu\mu}\,\delta_{\mu\nu} + \right. \tag{21}$$

$$\left. \sum_{B \neq A} \left[\sum_\lambda^B c_{\lambda i}\,(\beta_\mu + \beta_\lambda)\,S_{\mu\lambda} + \sum_\nu^A Z^B(\mu\nu \mid S^B S^B) + Z^A Z^B(S^A S^A \mid S^B S^B)\,g(R_{AB}) \right] \right\}$$

where $U_{\mu\mu}$ is the sum of the kinetic energy and potential energy of an electron in atomic orbital μ. β is an atomic parameter in the term for the core resonance integral. $S_{\mu\lambda}$ is the overlap integral of the Slater orbitals, and S^A, S^B, Z^A and Z^B are the S orbitals and core charges of atoms A and B. Note that the Slater orbital integrals are merely used as parameters in forming H and do not figure in the orthonormality of the MNDO MO's. The formal overlap integrals which appear there are taken to be mutually orthogonal, as stated above in connection with eq 18.

Collecting the first order terms in the expansion of eq (19), we have

$$\left| \frac{\partial F_{\mu i}}{\partial \bar{y}} \right|_{\bar{y}=0} = F_{\mu i}^{(1)} = H_{\mu i}^{(1)} + Q_{\mu i}^{(1)} + W_{\mu i}^{(1)} \tag{22}$$

where

$$H_{\mu i}^{(1)} = \sum_A \sum_{B \neq A} \sum_\mu^A c_{\mu a} \left[\sum_\lambda^B c_{\lambda i}\,[\,\beta_\mu + \beta_\lambda\,]\,S_{\mu\lambda}^{(1)} + \sum_\nu^A c_{\nu i}\,Z^B(\mu\nu \mid S^B S^B)^{(1)} \right] \tag{23}$$

and

$$Q_{\mu i}^{(1)} = \sum_j^{2n} \sum_s^{2N} u_{rj}^*\,u_{si}\,(a\,r \mid\mid i\,s)^{(1)} = \sum_j^{2n} (a\,j \mid\mid i\,j)^{(1)} \tag{24}$$

and

$$W_{ai}^{(1)} = \sum_{j}^{2n} \sum_{r}^{2N} \left\{ u_{rj}^{*(1)} u_{ri} \; (a\ r \quad i\ s) + u_{rj}^{*} u_{ri}^{(1)} (a\ r \quad i\ s) \right\}$$

$$= \sum_{j}^{2n} \sum_{r}^{2N} u_{rj}^{(1)} \left\{ (a\ r \quad i\ j) + (a\ j \quad i\ r) \right\} \tag{25}$$

In eqs (24) and (25), the subscripts a, i, j, r and s include both the α and β spin cases. Since, for a closed-shell SCF wavefunction, the $a_\beta i_\beta$ pairs are equal to the $a_\alpha i_\alpha$ pairs, and the $a_\alpha i_\beta$ pairs vanish, only the $a_\alpha i_\alpha$ pairs need to be considered here. Applying eq (20) and the antisymmetry of $U^{(1)}$, eqs (24) and (25) can be written

$$Q_{ai}^{(1)} = \sum_{j}^{n} \left\{ (a_\alpha i_\alpha \quad j_\alpha j_\alpha)^{(1)} + (a_\alpha i_\alpha \quad j_\beta j_\beta)^{(1)} - (a_\alpha j_\alpha \quad j_\alpha i_\alpha)^{(1)} \right\}$$

$$= \sum_{j}^{n} \left\{ 2\,(a\ i \quad j\ j)^{(1)} - (a\ j \mid j\ i)^{(1)} \right\}$$

$$= \sum_{\mu\nu\lambda\sigma} c_{\mu a} \left\{ 2\, c_{\nu i} P_{\lambda\sigma} - c_{\sigma i} P_{\nu\lambda} \right\} (\mu\nu \mid \lambda\sigma)^{(1)} \tag{26}$$

and

$$W_{ai}^{(1)} = \sum_{j}^{n} \sum_{b}^{N} u_{bj}^{(1)} \left\{ \begin{array}{l} (a_\alpha i_\alpha \mid b_\alpha j_\alpha) + (a_\alpha i_\alpha \mid b_\beta j_\beta) - (a_\alpha j_\alpha \mid b_\alpha i_\alpha) + \\ (a_\alpha i_\alpha \mid j_\alpha b_\alpha) + (a_\alpha i_\alpha \mid j_\beta b_\beta) - (a_\alpha b_\alpha \mid j_\alpha i_\alpha) \end{array} \right\}$$

$$= \sum_{j}^{n} \sum_{b}^{N} u_{bj}^{(1)} \left\{ 4\,(a\ i \mid b\ j) - (a\ j \mid b\ i) - (a\ b \mid j\ i) \right\} \tag{27}$$

$$= \sum_{j}^{n} \sum_{b}^{N} u_{bj}^{(1)} \sum_{\mu\nu\lambda\sigma} c_{\mu a} \left\{ 4\, c_{\nu i} c_{\lambda b} c_{\sigma j} - c_{\nu j} c_{\lambda b} c_{\sigma i} - c_{\nu b} c_{\lambda j} c_{\sigma i} \right\} (\mu\nu \mid \lambda\sigma)$$

Eq (18) can now be written

$$\vec{B}_\alpha + \vec{B}_\beta = \vec{B}_{0\alpha} + \vec{B}_{0\beta} + A_\alpha \vec{B}_\alpha + A_\beta \vec{B}_\beta \tag{28}$$

or

$$(1 - A)\,\vec{B} - \vec{B}_0 = 0 \tag{29}$$

where $\vec{B} = \vec{B}_\alpha + \vec{B}_\beta$ is the vector of dimension $n(N-n)$ comprised of the doubly occupied-virtual elements of $U^{(1)}$. $\vec{B}_0$ is the vector of the same dimension with ele-

ments

$$\vec{B}_{0\,(ai)} = \frac{H_{ai}^{(1)} + Q_{ai}^{(1)}}{\epsilon_i(0) - \epsilon_a(0)} \tag{30}$$

and A is the square matrix of the same dimension defined by $W = \Delta\vec{\epsilon}\ A\ \vec{B}$, with elements

$$A_{(ai\ bj)} = \frac{(ab\ \ ij) + (aj\ ||\ ib)}{\epsilon_i(0) - \epsilon_a(0)} \tag{31}$$

In the MNDO approximation, the three- and four- center integrals are set to zero, as well as many of the two-center integrals. The one-center integrals are constants taken from empirical data, and so all of these derivatives are zero. Derivatives are needed only for the portion of the two-center electron repulsion integrals which restrict each electron to only one atom $(\mu^A \nu^A \mid \lambda^B \sigma^B)$, where $A \neq B$, and for the overlap integral of the Slater orbitals $S_{\mu\lambda}$. $S^{(1)}$ is symmetric, so only its upper triangle need be evaluated. Equations for $S^{(1)}$ and for the derivatives of the two-center repulsion integrals have been worked out by Dewar and Yamaguchi.[24] and will not be repeated here.

The order of operations for the simultaneous transformation $(\mu\nu \mid \lambda\sigma) \rightarrow A_{(ai\ bj)}$ is $(n(N-n))^2 M$, where M is the total number of integrals. For an ab-initio study, there are N^4 integrals. In MNDO, only a portion of the one- and two- center integrals are considered, but it is still a time consuming process. This can be reduced considerably by making use of permutational symmetry and by transforming in four separate steps. In the first step, $(\mu\nu \mid \lambda\sigma) \rightarrow A_{(ai\ \lambda\sigma)}$, the two-center integrals are transformed according to

$$A_{(\mu\nu\,|\,\lambda\sigma)} = \sum_{\mu}^{M}\sum_{\nu}^{M} c_{\mu\mu}\ (\mu\nu\ \ \lambda\sigma) \tag{32}$$

In the second step, the one center integrals are mixed in with the matrix $A_{(\mu\nu\,|\,\lambda\sigma)}$ according to

$$\begin{aligned} A_{(\mu i\,|\,\lambda\sigma)} = \sum_{i}\sum_{A}\sum_{\nu}^{A}\Bigg\{ \sum_{\mu}\Big[& 2\,c_{\nu\mu}\,c_{\nu i}\,(\nu\nu\ |\ \nu\nu) \\ & + \sum_{\mu}^{\nu}\Big\{ \begin{array}{l} (c_{\nu\mu}c_{\nu i} + c_{\mu\mu}c_{\mu i})\,[\,4\,(\nu\nu\ \ \mu\mu) - 2\,(\nu\mu\ |\ \nu\mu)\,] + \\ (c_{\nu\mu}c_{\mu i} + c_{\nu\mu}c_{\mu i})\,[\,3\,(\nu\mu\ \ \nu\mu) - (\nu\nu\ |\ \mu\mu)\,] \end{array} \Big\} \\ & + \sum_{B}\sum_{\lambda}^{B}\sum_{\sigma}^{\lambda}\,(\,4\,c_{\nu i} - c_{\lambda i} - c_{\sigma i}\,)\,A_{(\mu\nu\,|\,\lambda\sigma)} \Big]\Bigg\} \end{aligned} \tag{33}$$

For the third step, $A_{(\mu i\,|\,\lambda\sigma)} \rightarrow A_{(\mu i\,|\,b\sigma)}$.

$$A_{(\mu i\,|\,b\sigma)} = \sum_{b}\sum_{\lambda} c_{\lambda b}\ A_{(\mu i\,|\,\lambda\sigma)} \tag{34}$$

The final step, $A_{(\mu i\,|\,b\sigma)} \rightarrow A_{(\mu i\,|\,bj)}$, proceeds as

$$A_{(\mu i\,|\,bj)} = \sum_{j}\sum_{\sigma} \frac{c_{\sigma j}\,A_{(\mu i\,|\,b\sigma)}}{\epsilon_i(0) - \epsilon_\mu(0)} \tag{35}$$

$\vec{B}_0$ can be transformed along with A according to

$$\vec{B}_{0\,(\mu\nu)} = \sum_{\mu}^{vir}\sum_{A}\sum_{B\neq A}\sum_{\mu}^{A} c_{\mu\mu}\left\{\beta_\mu\,S^{(1)}_{\mu\nu} + Z^B\,(\,\mu\ \nu\ \ S^B S^B\,)^{(1)} + \sum_{\lambda\sigma}^{B} P_{\lambda\sigma}\,(\mu\nu\ |\ \lambda\sigma)^{(1)}\right\} \tag{36}$$

followed by

$$\vec{B}_{0\,(\mu i)} = \sum_{i}^{n}\sum_{\nu}^{M} \frac{c_{\nu i}\ \vec{B}_{0\,(\mu\nu)}}{\epsilon_i(0) - \epsilon_\mu(0)} \tag{37}$$

In MNDO, the number of integrals over AO's is small enough for us to store them in random access memory, or real storage. Each of the arrays in eqs (32) to (34) used for the transformation is smaller than the number of integrals, so these also are held in real storage. However, using double precision variables, for a moderately sized system of 100 valence orbitals, the A matrix would require 50 million bytes of real

storage, and in our implementation it is stored on disk.

The set of simultaneous equations in eq (29) are solved without having to invert the matrix $(1-A)$, using the Direct Inversion in Iterative Subspace method from Gaussian 82.[25] Define a set of orthogonal vectors $\vec{B}_n$, such that

$$\vec{B}_{k+1} = A\vec{B}_k - \sum_{l=0}^{\infty} \frac{<\vec{B}_l \mid A \mid \vec{B}_k>}{<\vec{B}_l \mid \vec{B}_l>} \vec{B}_l \quad (38)$$

or

$$\vec{B}_{k+1} = A\vec{B}_k - (\text{projection of } A\vec{B}_k \text{ on } \vec{B}_k, \vec{B}_{k-1}, \cdots \vec{B}_0) \quad (39)$$

The solution set $\vec{B}$ can be approximated from the vectors $\vec{B}_n$ by defining a set of coefficients α_n such that

$$\vec{B} = \alpha_0\vec{B}_0 + \alpha_1\vec{B}_1 + \cdots + \alpha_n\vec{B}_n \quad (40)$$

The coefficients α are obtained by requiring that the projections of the left side of eq (39) on the $\vec{B}_n$'s vanish. The procedure converges quickly, normally within about 10 iterations, giving the required elements of $U^{(1)}$.

THE TRANSITION STATE LOCATION STRATEGY

In most cases, the reactions studied will involve breaking bonds between certain atoms and forming bonds between others, which can present a problem during the optimization. For a simple bond disassociation, the energy, as a function of bond distance, climbs quickly, then levels off into a flat region, with large changes in the curvature throughout the region. However, the energy as a function of bond order behaves much better. The flat region virtually disappears, as the bond order is almost zero here. As the bond forms, changes to the bond order produce smaller and smaller changes to

the bond distance. The orthogonality adjustment is automatically scaled, so that for a given change in bond order, a small change is made to a bond which is almost fully formed, but a much larger change is made to a bond which is almost fully broken. For these reasons, when a bond breaks or forms during a reaction, the corresponding bond lengths are converted to bond orders, using the Pauling relationship[27]

$$R = R_0 + A \; LOG \; (B) . \tag{41}$$

where R is the bond length, B is the bond order, and R_0 and A are constants. The APS bond orders of the reactant and product are used to determine the values of R_0 and A. The bond order is substituted for the bond length during the path maximum searches as well as the optimization. The corresponding components of the constraint vector are formed by subtracting the bond orders of the reactant and product structures. For the optimization, the derivative of the bond order is computed by applying the chain rule,

$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial R} \frac{\partial R}{\partial B} = \frac{LN\ (10)}{B} \frac{\partial E}{\partial R} \tag{42}$$

If the energy and P_e change discontinuously for a small change in P_g, this usually means that there has been a qualitative change to the wavefunction, such as an inversion in the order of the HOMO and LUMO. Perhaps the first structure is strained, but reactant-like, but the small change in geometry allows a shift to a still strained, but more product-like wavefunction. The reaction might be formally disallowed, and require a more sophisticated wavefunction than the simple SCF expression. However, changes to the geometry might help matters. By breaking symmetry, a mechanism close to a two step procedure can be followed more easily. There could also be prob-

lems with the reaction pathway, as defined by the internal variable set.

The first step in using the program is to optimize the reactants and products of the reaction, and then examine their internal variables. They should be as similar as possible, so that the constraint vector reflects chemical changes, rather than unimportant conformational ones. The orientation of the molecules will greatly effect the ease of convergence. Atoms between which a bond is forming might have to be brought closer together or further apart, or fragments might have to be moved in order to approach each other from a different orientation. The fragments should be optimized separately, and then together, with the forming bonds fixed at about 0.75 Å to 1.0 Å longer than the optimized bond length. Certain angles will also have to be fixed, since the gradient will strongly favor separating all non-bonded atoms. When these structures are submitted in a transition state search, the APS bond orders are printed for each bond which is forming or breaking, and a warning message is given if the bond order in the reactant or product is less than 0.01.

Consideration must be given as to how a reaction is formally allowed. Depending on whether a rotation will be con or dis, or a sigmatropic shift will be suprafacial or antarafacial, etc., a different product will form, and the correct product must be given.

Linear bond angles present a problem, which becomes critical if a bond angle is linear in either the reactant or product, but not both. The dihedral angles through these atoms become undefined for the linear structure, making it impossible to defi e the corresponding components of the constraint vector. The reactions involving acetylene fell into this category. The sigmatropic shift reaction of acetylene giving

vinylidene could not be studied using linear acetylene. The hydrogens had to be bent slightly into a 'trans' structure, so that the dihedral angle would be well defined. Similarly, for the addition reaction of acetylene plus hydrogen giving ethylene, the hydrogens had to be bent slightly into a 'cis' structure. As a general rule, bond angles which do not remain constant should be no larger than 175° in either structure.

If a bond angle will increase beyond 180°, it also presents a problem. The dihedral angles through this angle will instantaneously change by 180° as it rotates through the linear bond angle. As an example, in the decomposition reaction of ethylene giving vinylidene plus hydrogen (see fig B), ^4H increases its bond length and rotates to the other side of the $C=C$ bond, while ^3H increases its bond length and rotates towards the center of the $C=C$ bond. The transition state is planar, with both of the leaving hydrogens on the same side of the $C=C$ bond. If the internal variables are defined using the bonds of ethylene, the path is discontinuous at the point of the linear $C=C-H$ structure. This problem was solved by defining the dihedral angle for ^4H as $^4H-^3H-^2C=^1C$. In this way, all of the angles remained well below 180°. A similar problem arose for Reaction G, and was solved in the same manner.

In general, symmetry should be broken as much as possible. A symmetrical structure could have a symmetrical gradient, and thus force symmetry to be maintained, even though the true transition state is asymmetrical. The search strategy will reestablish symmetry for truly symmetrical transition states.

The next step is to map out the constraint vector in steps of 0.2. The energy and electronic path coordinate are computed for each point. If the energy increases

smoothly, passes through a maximum, then descends towards the product's value, while P_e also increases smoothly, with no sudden jumps, then the transition state should be located quickly. If P_e remains effectively zero while P_g indicates there is substantial progress into the reaction, and/or the energy climbs to a very high value, conformational changes to the reactant and/or product should help. The internal variables of each point should be examined, especially if any bond angles are nearly linear, or if any fragments are formed. The orientation of any fragments will greatly effect the ease of convergence. They can potentially approach each other in an infinite number of ways. Chemical intuition or knowledge of the structure of the transition state as found from a different method, or obtained during the course of the search, must be used to construct a suitable structure. Different orientations can be evaluated by examining the reaction paths they would form. The orientation with the lowest energy maximum along the path, and the smoothest change in P_e, should lead to the transition state in the minimum amount of time. If P_e changes very quickly for small changes in P_g, this could indicate that the reaction is formally disallowed. Even so, changing the orientation of the fragments might still facilitate the search. A certain orientation might favor a two step mechanism, rather than the forbidden concerted reaction.

In the initial line search, the LST path maximum is found along the constraint vector, and is then optimized at constant electronic path coordinate. After four optimization line searches, or if the gradient rises significantly, or if P_e has drifted, a line search along the QST pathway is carried out to locate the QST path maximum, which is then further optimized. Successive QST maximums should be of lower energy and

gradient, finally converging on the transition state of the reaction.

This simple strategy will work for some reactions, but usually locating the transition state is more complicated. In some cases, successive path maximums will tend to oscillate, one being reactant-like, the next product-like. This produces large changes in P_e, and often increases the values of the energy and gradient above the values of the previous path maximum. In other cases, the optimization will stagnate, in that both the optimization vector and the step taken along the vector approach zero, and virtually no progress is made. This can sometimes be helped by substituting one of the other updating schemes to the hessian. Or, the optimization can get well below the value of the transition state. This can cause the path maximums to oscillate, as previously discussed, or to locate a local maximum, perhaps only created by an inappropriate reaction path. In these cases, flanking points will have to be utilized to segment the pathway.

These flanking points can be found on either an LST or QST path. The best guess for the transition state is used as the third point on the QST path. A structure of a particular P_g is computed, which is then optimized at constant P_e. As the optimization proceeds for these points, the energy and gradient should decrease. There is no need for the flanking points to be fully optimized. In fact, too much optimization could actually find an unrealistic structure. Five to ten line searches should be sufficient. The flanking points are then used to construct a new constraint vector, and the optimization continues.

If fragments are formed, optimization of the fragment-side flanking point will often be difficult. Since there is virtually no bond, a small change to the bond order will cause a large change in the bond length, and the gradient will strongly favor separating the fragments. For this reason, for flanking points, the optimized variable should be the bond length, rather than the bond order. As a general rule, no more than 1.0 to 1.5 Å should be added to a bond which is breaking. These structures must be examined to ensure that the optimization is not merely separating the fragments, or essentially finding the reactant or product structure. Often, the optimization goes wrong after a few line searches, but one of the early structures is suitable as a flanking point. The structure should be examined, as it might suggest a better orientation for any fragments.

In troublesome cases, it might be necessary to alternate many times between searching for the TS and finding new flanking points. In this way, we can 'walk' the endpoints of the reaction path in from the reactant and product, until a vector is found which will cross through the general region of the transition state. In cases of extreme difficulty, points very close to the reactant and product should be used, such as 0.1 and 0.9, and then closer and closer to the middle, until reasonable structures are found. If the current estimate seems to be fairly close to the true TS, then the flanking points can be close to the current estimate. The flanking points must not be too close to the candidate TS, however. If the optimization of the flanking points were to bring one of them to the other side of the TS, then the TS would be excluded from the search. The criteria which can be used to judge the 'quality' of a structure are its energy, gradient,

P_R, P_e, and the internal variables which define it.

A very effective technique to locate flanking points was used once the optimization had found a reasonable estimate for the TS. In all the reactions studied, the results of the first search were used, starting from the initial LST maximum, and allowing 25 to 50 line searches, or when successive QST maximums had approximately the same or even higher energies and gradient. If no previous information as to the structure of the TS is available, then the technique could be implemented once the RMS gradient has been lowered to below 10.0, preferably around 1.0, Kcals/coordinate unit. The coordinate units are Å for bond lengths, and radians for bond angles and dihedral angles. When a bond order is used in place of the bond length for a forming or breaking bond, the coordinate unit is the bond order. In general, one of the flanking points will be of 'better quality' than the other. An LST path is constructed between this point and the current estimate for the TS, and a point beyond this best guess is found. This is accomplished by assigning the flanking point and the best guess both geometric and electronic path coordinates of 0.0 and 0.5, respectively. A structure of P_R= 0.75 to 1.0 on this LST path can then be found and optimized at constant P_e. The original and newly located flanking points are then given path coordinates of 0.0 and 1.0 (or 1.0 and 0.0), respectively, and the search is resumed along the QST path through the best guess structure. Convergence was achieved quickly using this technique. In only one case did an additional flanking point have to be found after applying this technique.

There are several input options which help to further constrain the optimization

in troublesome cases. The default is to perform a linear search for the minimum in energy along the optimization vector. The first three options are used to control this search. If too large a change is made, the search could leave the region of the TS, but if the linear search is severely restricted, then much time can be wasted by taking trivial steps along the vectors which point towards the TS. The best way to overcome this problem seems to be to allow a large number of steps (3 to 5) along the vectors, but halt the line search when the gradient along the vector has been reduced to 30% to 70% of its initial value, rather than to 20%, as for an unconstrained optimization. Finally, when the initial step has not sufficiently reduced the gradient along the vector, the step length is multiplied by 5 for an unconstrained optimization, which might be too much for a transition state search. In this study, a value of 2 was used, but a value of 3 might be more appropriate.

The fourth option concerns the approximation to the Hessian, which is updated at the end of each line search. On quadratic surfaces, the true Hessian is identical at all points. On more realistic surfaces, the true Hessian will change as the curvature of the region changes. The program tests the approximation to the Hessian after each update, and will reinitialize it if it becomes positive definite. The initialization can be made by taking a step in the direction of the gradient and computing the change in gradient, or, if the option is called, by using the change in gradient and position which was used in the last update of the Hessian.

RESULTS

Schroder and Thiel[28, 29] have located many transition states for small molecules using the MNDO program. 13 of these reactions, plus 2 ring closure reactions, were used to test the Iso-Electronic Constraint method. The convergence criterion used in this study were 0.02 Kcals/coordinate unit for the RMS gradient, and 0.0001 coordinate unit for the length of the optimization vector. Table 1 lists the internal variables and their derivatives for each transition state as found using the Iso-Electronic method, plus the values reported by Schroder and Thiel. A graph of geometric path coordinate vs. heat of formation is also presented for each reaction (see figs A-P). On each graph is a computer drawn picture of the reactant, product, transition state, and final flanking points, if they were used for that reaction. When no flanking points were used, the paths shown on the graphs are QST paths between Reactant, TS, and Product.

When flanking points were used, except where noted, the reaction paths were constructed in the following manner. The path coordinate of each of the final flanking points was determined by forming QST paths linking them with the 'Reactant' and 'Product'. The 'Reactant' and 'Product' are then connected to the flanking points via LST paths. The flanking points and Transition State are connected with a QST path. In most cases, the bond orders were interpolated for the bonds which were breaking or forming. In other cases, the bond lengths were interpolated on this QST segment.

Figure Q contains stick figure drawings of each transition state, showing the numbering scheme used for the different atoms.

All optimizations, whether of transition states or flanking points, are iso-electronic optimizations. Transition state optimizations differ in that they start with, and periodically reset to, the LST or QST path maximums along the constraint vector. Also, for transition states, the bond lengths which form or break are converted to bond orders for the optimization, but for flanking points, the bond lengths are used directly. The end points of the constraint vector are the reactants, products, or appropriate flanking points. These end points are assigned the same values for both P_r and P_e, which are usually 0.0 or 1.0.

Table 2 gives the activation energy, the RMS gradient, and the path coordinates of each Transition State. Table 3 gives the XYZ coordinates of the reactant and product of each reaction.

For 11 of the 15 reactions studied, convergence was achieved without the use of flanking points. For the other 4 reactions, two, three, or five flanking points were used. The same transition state was found for 12 of the 13 reactions also studied by Thiel, et al. A different structure of slightly lower energy was found for reaction K. A brief description of the technique used and problems encountered for each reaction follows.

Reaction A, Ethylene to Methyl Carbene, converged in 30 line searches and 50 energy and gradient computations, after the initial LST path maximum.

Reaction B, Ethylene to Vinylidene and Hydrogen, converged after some difficulty. The optimized vinylidene fragment was placed with the carbons along the X axis. In the transition state, both of the leaving hydrogens have positive Y coordinates, that is,

they are on the same side of the $C=C$ bond. This requires one of the bond angles to increase beyond 180°, causing problems with the dihedral angle. This was solved by defining the dihedral angle as running through the two hydrogens, and then through the two carbons. In this way, all the optimized angles remained less than 180°. The 'product', which shall be referred to as structure P, was formed by placing a hydrogen molecule near the vinylidene. The $^5H-^3C$ bond length was fixed at 1.8 Å, and the $^4H-^5H$ bond length was fixed at 0.67 Å. The angle $^4H-^3C-^2C$ was fixed at 121.3°, and the angle $^5H-^4H-^3C$ was fixed at 16.3°. The dihedral angle $^5H-^4H-^3C-^2C$ was 180.0°. This made the angle $^5H-^3C-^2C$ 127.3°, and the angle $^4H-^5H-^3C$ 157.7°. The LST maximum between ethylene and P was found and optimized for 45 line searches, which reduced the heat of formation from 311.75 to 140.51 kcals, giving structure A. A structure of 0.8 along the QST path was found and optimized for 7 line searches, giving flanking point B. Point B was given a path coordinate of 0.0, and point A a path coordinate of 0.4. An extrapolation to a structure of 0.9 along this LST path was made. This point was optimized for 7 line searches, giving reactant side flanking point C. A QST path was constructed between C, A, and B. The QST maximum was found and optimized for 20 line searches, using the DFP update, until the transition state was found.

Reaction C, Ethylene to Acetylene and Hydrogen, also took a fair amount of effort. Acetylene was optimized and then forced into a cis configuration by laying the molecule along the X axis, and setting the Y coordinates of the hydrogens to 0.1 and 0.15 Å. The Y coordinates of the leaving hydrogens were fixed at -2.0 and -2.4 Å, and

the X coordinates at -0.328 and 0.3, with a carbon at the origin. Starting from P_R= 0.75, the LST maximum was found and optimized for 48 line searches, giving point A. This reduced the heat of formation from 169.39 to 164.80 kcals, and the gradient from 97.6 to 2.0. The gradient rose to 7.8 with the next line search, and remained high after this. Point A was used to define a QST path, and a structure of P_R= 0.35 was optimized at constant P_e for 6 lines searches, giving flanking point B. Point B was assigned a path coordinate of 0.0, and point A a coordinate of 0.5. A structure of P_R of 0.75 on this LST path was optimized at constant P_e for 7 line searches, giving flanking point C. During the optimization of these flanking points, the Z coordinates assumed small, but *non-zero* values. Since this has been observed to increase the gradient, and cause the iso-electronic adjustment to alter the dihedral angles, and therefore the Z coordinates, the Z coordinates (which ranged from 0.194 to -0.046) were set to zero, giving points A', B', and C'. A QST path was constructed between A', C', and B'. The QST maximum was found and optimized, using the DFP update. After 24 line searches, the heat of formation was reduced to 164.75 kcals, and the gradient to 0.024. After this, very little progress was made. At line 38, the energy had been reduced by $4x\,10^{-6}$, and the gradient had risen to 0.036. The structure found in line 24 agrees very closely with that found by Schröder and Thiel, and the search was halted. Schröder and Thiel report this structure as a stationary point, with two negative eigenvalues of the hessian. This might explain the inability to reduce the gradient to zero and the deviation from a planar system.

Reaction D. Methyl Carbene to Acetylene and Hydrogen. Again, the acetylene had to be 'modified' in order to be used. The dihedral angle is undefined for the linear structure, and had to be bent slightly into a 'cis' structure. Using this as the product, the transition state was found in 20 line searches and 33 energy and gradient computations, after the initial LST path maximum.

Reaction E. Acetylene to Vinylidene. The acetylene also had to be 'modified' in order to be used. The dihedral angle is undefined for the linear structure, and had to be bent slightly into a 'trans' structure. Using this as the reactant, the transition state was found in 30 line searches and 66 energy and gradient computations, after the initial LST path maximum.

Reaction F, the 1-2 sigmatropic shift reaction of Trans-Hydroxy Carbene to Formaldehyde, converged in 15 line searches and 29 energy and gradient computations, after the initial LST path maximum.

Reaction G, Carbon Monoxide and Hydrogen to Formaldehyde, converged in 49 line searches and 91 energy and gradient computations, after the initial LST path maximum. The activation energy was found to be 1.6 Kcals lower than that reported by Thiel, et al., but the geometries agree very closely.

For reaction H, the Trans to Cis rotation of Hydroxy Carbene, the transition state was found after 17 line searches using the iso-electronic constraint, but the gradient could only be reduced to 0.049. Note that no bonds are broken during this reaction, so

the iso-electronic method is not expected to be appropriate (but still works, more or less).

Reaction I, Cis-Hydroxy Carbene to Carbon Monoxide and Hydrogen, converged in 28 line searches and 59 energy and gradient computations. The structure reported by Schroder and Thiel is different, but will coincide if the carbon atom is switched with the oxygen. The activation energy was found to be 18.8 Kcals lower than their value. It is assumed a trivial error was made in reporting their data.

Reaction J, Methanol breaking up into Formaldehyde and Hydrogen, converged slowly, but without using any flanking points. The transition state was found after 80 line searches and 161 energy and gradient computations, after the initial LST maximum.

Reaction K, Methanol breaking up into Methylene and Water. A different structure was found from that reported by Schroder and Thiel. A two step mechanism was found, where first a hydrogen migrates to the oxygen atom, forming a stable intermediate, and then the C-O bond breaks, in a seperate step. A zero gradient Transition State was found for each step.

The structure found by the optimization strategy had zero gradient, but it was not the maximum along the path connecting the reactant and 'product'. This stationary point was used as the 'reactant', and the minimum between this and the 'product' was partially optimized. This minimum was used as the 'reactant', and the path max-

imum was found and optimized, until a new zero gradient structure was found. Neither of these path maximums matched the structure reported by Schroder and Thiel. The activation energies (using methanol as the reference) for the two stationary points were found to be 106.35 kcals/mol and 109.14 kcals/mol, while Schroder and Thiel reported an energy of 109.4 kcals/mol and a much more symmetrical TS. They report the migrating hydrogen as being equidistant from the carbon and oxygen atoms, and that the plane of the forming water molecule bisects the angle of the carbene. In this study, for the first path maximum, the migrating hydrogen was found to be 1.115 Å from the oxygen and 1.465 Å from the carbon, with a bond order of 0.536 to the oxygen and 0.338 to the carbon. The C-O bond distance was 1.435 Å, with a bond order of 0.966. The minimum in energy between the two path maximums seems to be a Lewis acid-Lewis base complex of methylene and water. The migrating hydrogen was 0.950 Å from the oxygen and 2.083 Å from the carbon, with a bond order of 0.932 to the oxygen and 0.004 to the carbon. The C-O bond distance was 1.576 Å, with a bond order of 0.640. In the true Transition State for the reaction, the C-O bond length was 2.16 Å, and the bond order was 0.126. The actual geometries of all the structures are given in table 3.

Two flanking points were used for the first search. The 'product' was constructed by fixing the C-O bond length at 2.7 Å, and freely optimizing all other variables. This was then further optimized, including the C-O bond length, giving point P. The final C-O bond length was 2.95 Å. The LST maximum between methanol and point P was found and optimized for 31 line searches, giving point A. This lowered the energy from

74.6 to 49.0 Kcals. A QST path was formed between methanol, A, and P, and a structure of P_R= 0.5 along this path was optimized for 6 line searches, giving flanking point B. This lowered the energy from -3.8 to -17.3 Kcals. Points B and A were assigned P_R= 0.0 and 0.5. A structure of P_R= 0.8 along this LST path was optimized for 6 line searches, giving flanking point C, which lowered the energy from 66.1 to 32.7 Kcals. A QST path was formed using points B, A, and C, and the QST maximum was found and optimized for 19 line searches using the Powell update. The search was continued using the DFP update, and convergence upon the first transition state was achieved after 5 additional line searches. The energy was found to be 49.02 kcals, and the gradient 0.0192.

The minimum along the LST path between point D and point P was optimized at constant P_e for 9 line searches, which lowered the energy from 43.4 to 34.9 Kcals, giving point E, with a gradient of 6.0. The LST maximum between points E and P was found and optimized for 31 line searches, giving the final Transition State. The heat of formation was 51.81 Kcals, and the gradient was 0.0107.

Reaction L, Methanol to Trans-Hydroxy Carbene and Hydrogen, required five flanking points. A structure of P_R= 0.8 along the LST path connecting methanol and the 'product' was optimized for 9 line searches, giving flanking point A, and lowering the heat of formation from 65.0 to 36.4 kcals. A structure of P_R= 0.4 along this same LST path was optimized for 11 line searches, giving flanking point B, and lowering the heat of formation from 20.8 to 18.1 Kcals. The LST maximum was found between points A and B, which was optimized for 27 line searches, giving flanking point C. This

lowered the heat of formation from 83.9 to 47.3 Kcals. An LST path was formed between points A and C, and flanking point D was found by optimizing a structure 1/2 way between them for 6 line searches. An LST path was formed with points D and C, with path coordinates of 0.0 and 0.5. A structure of $P_R = 0.8$ on this path was optimized for 7 line searches, giving flanking point E. A QST path was formed using points D, C, and E, and the QST maximum was found and optimized for 40 line searches, which lowered the heat of formation from 48.0 to 44.0 kcals, giving point F. An LST path was formed using D and F. A structure 1/2 way between these points was optimized for 1 line search, giving flanking point G. A QST path was formed using points G, F, and E, and the QST maximum was found and optimized, using the DFP updating scheme. The transition state was found after 38 line searches, with a heat of formation of 43.845 Kcals, and a gradient of 0.0162. The reaction path (see Fig L) was divided into 3 segments. The first segment was an LST path between methanol and point G. The end segment was an LST path between point E and the 'product'. The middle segment was a QST path connecting points G, F, and E, on which the bond orders, rather than the bond lengths, were interpolated, for the bonds which were breaking or forming.

Reaction M, Formic Acid degenerate rearrangement, converged in 31 line searches and 56 energy and gradient computations, after the initial LST path maximum. The activation energy was found to be 3.8 Kcals higher than that reported by Thiel, et al., but the geometries agree very closely.

Reaction N, the dis-rotatory ring opening of Cyclopropyl Anion to Allyl Anion, converged in 16 line searches, after the initial LST path maximum.

Reaction P, the con-rotatory ring closure of Butadiene to Cyclobutene, converged in 24 line searches, after the initial LST path maximum.

CONCLUSION

A new version of Synchronous Transit has been developed that is more reliable and much simpler to use than the original method. It is an efficient way of locating transition states of chemical reactions, and of mapping out good approximations to the reaction paths. Very few decisions are required from the user, which reduces the possibility of error. In cases where both the reactants and products are single molecules, it is extremely simple to use, and has converged very quickly for the 7 reactions of this type studied. When fragments are formed, their orientation is very important, and can greatly effect the ease of operation and speed of convergence. 4 of these reactions converged directly, requiring a small to moderate number or line searches, once a suitable orientation was used. The other 4 reactions studied required from 2 to 5 flanking points, and so more computer time and experience by the user is needed.

The changes were written with the intent of making them 'portable', in that it would be a relatively simple task to 'insert' them into other programs such as Gaussian 80. The Synchronous Transit package is called in the main routine to generate the starting point for the optimization. The optimization routine is relatively self-contained, and could be fairly easily substituted for the existing optimization routine.

Or the existing routine could be modified to include the iso-electronic orthogonalization and path maximum searches. The only subroutine which depends strongly on the method used for evaluating the energy is subroutine CPFH. Each method uses a different subset of the total number of orbital interactions, as would be used in a full ab initio study. Subroutine CPHF uses the integrals over atomic orbitals, and so must be changed for each method of evaluating the energy.

ACKNOWLEDGEMENTS

I wish to thank Merck, Sharpe & Dohme, Inc. for a generous grant in support of this project, and to the CUNY Computer Center and the CCNY Science Computer Center, for access to their computers and their valuable assistance. The FORTRAN code for the MNDO analytical gradient was supplied by James T. Stewart. Thanks also to Dr. Domingo Jon and Dr. Denise Garland, for many helpful conversations, and to my wife, Maryann, for her patience and understanding.

CAPTIONS FOR THE FIGURES

Fig A. Heat of Formation versus Geometric Path Coordinate for Reaction A. Energies given for the path defining structures are relative to the heat of formation of the reactant ethylene of 15.363 Kcals/mole.

Fig B. Heat of Formation versus Geometric Path Coordinate for Reaction B. Energies given for the path defining structures are relative to the heat of formation of the reactant ethylene of 15.363 Kcals/mole.

Fig C. Heat of Formation versus Geometric Path Coordinate for Reaction C. Energies given for the path defining structures are relative to the heat of formation of the reactant ethylene of 15.363 Kcals/mole.

Fig D. Heat of Formation versus Geometric Path Coordinate for Reaction D. Energies given for the path defining structures are relative to the heat of formation of the product acetylene + hydrogen of 59.063 Kcals/mole.

Fig E. Heat of Formation versus Geometric Path Coordinate for Reaction E. Energies given for the path defining structures are relative to the heat of formation of the reactant acetylene of 59.288 Kcals/mole.

Fig F. Heat of Formation versus Geometric Path Coordinate for Reaction F. Energies given for the path defining structures are relative to the heat of formation of the reactant formaldehyde of -32.929 Kcals/mole.

Fig G. Heat of Formation versus Geometric Path Coordinate for Reaction G. Energies given for the path defining structures are relative to the heat of formation of the reactant formaldehyde of -32.929 Kcals/mole.

Fig H. Heat of Formation versus Geometric Path Coordinate for Reaction H. Energies given for the path defining structures are relative to the heat of formation of the reactant trans-hydroxy carbene of 13.616 Kcals/mole.

Fig I. Heat of Formation versus Geometric Path Coordinate for Reaction I. Energies given for the path defining structures are relative to the heat of formation of the reactant cis-hydroxy carbene of 13.473 Kcals/mole.

Fig J. Heat of Formation versus Geometric Path Coordinate for Reaction J. Energies given for the path defining structures are relative to the heat of formation of the reactant methanol of -57.326 Kcals/mole.

Fig K. Heat of Formation versus Geometric Path Coordinate for Reaction K. Energies given for the path defining structures are relative to the heat of formation of the reactant methanol of -57.326 Kcals/mole.

CAPTIONS FOR THE FIGURES (cont)

Fig L. Heat of Formation versus Geometric Path Coordinate for Reaction L. Energies given for the path defining structures are relative to the heat of formation of the reactant methanol of -57.326 Kcals/mole.

Fig M. Heat of Formation versus Geometric Path Coordinate for Reaction M. Energies given for the path defining structures are relative to the heat of formation of the reactant formic acid of -92.561 Kcals/mole.

Fig N. Heat of Formation versus Geometric Path Coordinate for Reaction N. Energies given for the path defining structures are relative to the heat of formation of the reactant cyclopropyl anion of 56.885 Kcals/mole.

Fig P. Heat of Formation versus Geometric Path Coordinate for Reaction P. Energies given for the path defining structures are relative to the heat of formation of the reactant butadiene of 29.342 Kcals/mole.

Fig Q. Structures of Transition States for Reactions A-P.

Table I. Comparison of Internal Variables of the Transition States.

Reaction	Variable	Thiel et al.	Iso-Electronic	Gradient [a]
A	$^3C\,^2C$	1.378 Å	1.3780 Å	0.0026
	$^3C\,^5H$	1.320 Å	1.3166 Å	N.O. [b]
	$^3C\,^5H$	1.402 Å	1.4025 Å	-0.0098
$H_2C{=}CH_2$	$^2C\,^1H$	1.072 Å	1.0712 Å	-0.0001
-->	$^3C\,^4H$	1.099 Å	1.0991 Å	0.0005
$HCCH_3$	$^3C\,^6H$	1.100 Å	1.1002 Å	0.0010
	$^1H\,^2C\,^3C$	130.7°	130.92°	0.0026
	$^4H\,^3C\,^2C$	121.5°	121.40°	0.0023
	$^6H\,^3C\,^2C$	124.2°	124.31°	-0.0053
	$^1H\,^2C\,^3C\,^5H$	101.8°	101.67°	N.O.
	$^4H\,^3C\,^2C\,^5H$	88.1°	88.17°	0.00597
	$^6H\,^3C\,^2C\,^5H$	-113.9°	-113.76°	0.00323
B	$^2C\,^3C$	1.319 Å	1.3198 Å	-0.0010
	$^3C\,^5H$	1.284 Å	1.2829 Å	N.O.
$H_2C{=}CH_2$	$^3C\,^4H$	1.682 Å	1.6795 Å	0.0326
-->	$^5H\,^4H$	0.813 Å	0.8143 Å	-0.0162
$H_2C{=}C$	$^2C\,^6H$	1.091 Å	1.0909 Å	-0.0023
$+H_2$	$^2C\,^1H$	1.089 Å	1.0887 Å	-0.0031
	$^5H\,^3C\,^2C$	135.3°	135.11°	N.O.
	$^6H\,^2C\,^3C$	119.5°	119.48°	0.0027
	$^1H\,^2C\,^3C$	127.0°	126.99°	0.0016
C	$^2C\,^3C$	1.271 Å	1.2705 Å	0.0126
	$^2C\,^6H$	1.563 Å	1.5573 Å	0.1172
$H_2C{=}CH_2$	$^3C\,^6H$	1.367 Å	1.3691 Å	N.O.
-->	$^3C\,^5H$	1.370 Å	1.3759 Å	0.0125
$HCCH$	$^6H\,^5H$	0.985 Å	0.9866 Å	N.O.
$+H_2$	$^3C\,^4H$	1.076 Å	1.0757 Å	0.0006
	$^2C\,^1H$	1.051 Å	1.0506 Å	0.0000
	$^4H\,^3C\,^2C$	148.0°	148.44°	-0.0090
	$^1H\,^2C\,^3C$	160.9°	161.03°	0.0019

[a] The gradient is in units of Kcals/Å for bond lengths, Kcals/radian for angles.

[b] N.O. - These variables were not part of the optimization set.

Note that the atoms are arranged in a different order in the Thiel, et al. paper.

Table I. (Continued)

Reaction	Variable	Thiel et al.	Iso-Electronic	Gradient
D	$^{2}C\,^{3}C$	1.281 Å	1.2803 Å	0.0370
	$^{3}C\,^{4}H$	1.090 Å	1.0896 Å	0.0038
$HCCH$	$^{3}C\,^{5}H$	1.285 Å	1.2851 Å	0.0090
$+H_2$	$^{5}H\,^{6}H$	1.007 Å	1.0088 Å	N.O.
-->	$^{2}C\,^{1}H$	1.048 Å	1.0475 Å	0.0027
$HCCH_3$	$^{4}H\,^{3}C\,^{2}C$	132.4°	133.13°	0.0018
	$^{6}H\,^{2}C\,^{3}C$	165.5°	166.99°	-0.0116
E	$^{2}C\,^{3}C$	1.270 Å	1.2703 Å	0.0079
$HCCH$	$^{2}C\,^{4}H$	1.371 Å	1.3757 Å	N.O.
-->	$^{3}C\,^{4}H$	1.322 Å	1.3198 Å	-0.0247
$H_2C{=}C$	$^{2}C\,^{1}H$	1.059 Å	1.0583 Å	0.0073
	$^{1}H\,^{2}C\,^{2}$	167.0°	166.99°	0.0036
F	$^{2}C\,^{3}O$	1.283 Å	1.2831 Å	-0.0278
Trans-	$^{2}C\,^{3}H$	1.303 Å	1.3037 Å	N.O.
$HCOH$	$^{3}O\,^{3}H$	1.247 Å	1.2469 Å	-0.02737
-->	$^{2}C\,^{1}H$	1.098 Å	1.0983 Å	-0.0104
$H_2C{=}O$	$^{1}H\,^{2}C\,^{3}O$	121.4°	121.46°	-0.0063
G	$^{3}C\,^{4}O$	1.180 Å	1.1799 Å	0.0051
	$^{3}C\,^{1}H$	1.142 Å	1.1419 Å	N.O.
$CO+H_2$	$^{3}C\,^{2}H$	1.531 Å	1.5314 Å	-0.0395
-->	$^{1}H\,^{2}H$	1.060 Å	1.0601 Å	0.0033
$H_2C{=}O$	$^{1}H\,^{3}C\,^{4}O$	157.2°	157.22°	N.O.
	$^{2}H\,^{3}C\,^{4}O$	113.4°	113.44°	-0.0069
H	$^{2}C\,^{3}O$	1.328 Å	1.3271 Å	0.0034
Trans-	$^{3}O\,^{4}H$	0.942 Å	0.9416 Å	-0.0020
$HCOH$	$^{2}C\,^{1}H$	1.127 Å	1.1274 Å	0.0008
-->	$^{4}H\,^{3}O\,^{2}C$	120.8°	121.10°	-0.0024
Cis-	$^{1}H\,^{2}C\,^{3}O$	110.6°	110.91°	-0.0322
$HCOH$	$^{4}H\,^{3}O\,^{2}C\,^{1}H$	91.5°	91.84°	-0.0099
I	$^{2}C\,^{3}O$	1.255 Å	1.2540 Å	0.0006
Cis-	$^{3}O\,^{4}H$	1.470 Å	1.2891 Å	-0.0106
$HCOH$	$^{2}C\,^{1}H$	1.293 Å	1.4776 Å	-0.0409
-->	$^{4}H\,^{1}H$	1.058 Å	1.0581 Å	N.O.
$CO+H_2$	$^{4}H\,^{3}O\,^{2}C$	92.7°	77.58°	-0.0096
	$^{1}H\,^{2}C\,^{3}O$	77.2°	92.28°	0.0147

Table I. (Continued)

Reaction	Variable	Thiel et al.	Iso-Electronic	Gradient
J	$^{2}C\,^{3}O$	1.313 Å	1.3138 Å	0.0005
	$^{3}O\,^{4}H$	1.451 Å	1.4539 Å	0.0701
H_3COH	$^{2}C\,^{6}H$	1.406 Å	1.4035 Å	-0.0040
-->	$^{4}H\,^{6}H$	0.995 Å	0.9967 Å	N.O.
$H_2C{=}O$	$^{2}C\,^{1}H$	1.114 Å	1.1141 Å	-0.0001
$+H_2$	$^{4}H\,^{3}O\,^{2}C$	63.7°	63.46°	0.0154
	$^{6}H\,^{2}C\,^{3}O$	103.2°	103.62°	0.0140
	$^{1}H\,^{2}C\,^{3}O$	119.3°	119.23°	-0.0002
	$^{1}H\,^{2}C\,^{3}O\,^{4}H$	108.7°	108.95°	-0.0064
K	$^{2}C\,^{3}O$	1.897 Å	2.1597 Å	-0.0284
	$^{2}C\,^{5}H$	1.196 Å	2.5743 Å	N.O.
H_3COH	$^{3}O\,^{5}H$	1.196 Å	0.9452 Å	0.9452
-->	$^{3}O\,^{4}H$	0.941 Å	0.9452 Å	0.0104
CH_2	$^{2}C\,^{6}H$	1.091 Å	1.0946 Å	-0.0072
$+H_2O$	$^{2}C\,^{1}H$	1.091 Å	1.0946 Å	-0.0070
	$^{4}H\,^{3}O\,^{5}H$	98.1°	106.95°	N.O.
	$^{6}H\,^{2}C\,^{1}H$	114.0°	113.90°	N.O.
L	$^{2}C\,^{3}O$	1.327 Å	1.3270 Å	0.0134
	$^{3}O\,^{4}H$	0.951 Å	0.9513 Å	0.0062
	$^{2}C\,^{1}H$	1.097 Å	1.0969 Å	0.0025
H_3COH	$^{2}C\,^{6}H$	1.660 Å	1.6613 Å	0.0344
-->	$^{2}C\,^{5}H$	1.289 Å	1.2848 Å	0.0226
Trans-	$^{5}H\,^{6}H$	0.859 Å	0.8616	N.O.
$HCOH$	$^{4}H\,^{3}O\,^{2}C$	112.0°	112.03°	0.0018
$+H_2$	$^{1}H\,^{2}C\,^{3}O$	113.9°	113.88°	-0.0000
	$^{6}H\,^{2}C\,^{5}H$	95.4°	84.39°	N.O.
	$^{1}H\,^{2}C\,^{5}H$	112.1°	112.12°	N.O.
	$^{5}H\,^{2}C\,^{6}H$	30.7°	30.76°	N.O.
	$^{6}H\,^{2}C\,^{3}O$	108.6°	108.63°	0.0176
	$^{5}H\,^{2}C\,^{3}O$	119.7°	119.82°	-0.0046
	$^{4}H\,^{3}O\,^{2}C\,^{1}H$	162.7°	162.67°	-0.0152
M	$^{3}O\,^{2}C$	1.294 Å	1.2945 Å	-0.0081
	$^{3}O\,^{4}H$	1.318 Å	1.3184 Å	-0.0323
HCO_2H	$^{2}C\,^{4}H$	1.689 Å	1.6889 Å	N.O.
	$^{2}C\,^{1}H$	1.087 Å	1.0868 Å	0.0049
	$^{3}O\,^{2}C\,^{5}O$	100.7°	100.71°	-0.0183

Table I. (Continued)

Reaction	Variable	Iso-Electronic	Gradient
N	$^{2}C\,^{1}H$	1.1004 Å	0.0018
	$^{3}C\,^{2}C$	1.3997 Å	0.0030
	$^{4}C\,^{3}C$	1.3997 Å	0.0048
	$^{5}H\,^{4}C$	1.0998 Å	0.0005
	$^{6}H\,^{2}C$	1.0998 Å	-0.0001
	$^{7}H\,^{3}C$	1.0702 Å	-0.0044
	$^{8}H\,^{4}C$	1.1004 Å	-0.0041
	$^{3}C\,^{2}C\,^{1}H$	124.09°	0.0030
$CH_2CHCH_2^-$	$^{4}C\,^{3}C\,^{2}C$	84.66°	-0.0477
	$^{5}H\,^{4}C\,^{3}C$	124.08°	0.0041
	$^{6}H\,^{2}C\,^{3}C$	124.92°	0.0010
	$^{7}H\,^{3}C\,^{2}C$	137.66°	0.0002
	$^{8}H\,^{4}C\,^{3}C$	124.93°	0.0014
	$^{4}C\,^{3}C\,^{2}C\,^{1}H$	123.10°	-0.0107
	$^{5}H\,^{4}C\,^{3}C\,^{2}C$	-72.47°	-0.0064
	$^{6}H\,^{2}C\,^{3}C\,^{4}C$	72.52°	-0.0029
	$^{7}H\,^{3}C\,^{2}C\,^{6}H$	107.59°	-0.0008
	$^{8}H\,^{4}C\,^{3}C\,^{2}C$	123.10°	-0.0109
P	$^{2}C\,^{1}H$	1.0919 Å	0.0001
	$^{3}C\,^{2}C$	1.4172 Å	0.0015
	$^{4}C\,^{3}C$	1.4021 Å	0.0004
	$^{5}C\,^{4}C$	1.4172 Å	0.0009
	$^{6}H\,^{5}C$	1.0919 Å	-0.0003
	$^{7}H\,^{2}C$	1.0922 Å	0.0006
	$^{8}H\,^{3}C$	1.0811 Å	0.0005
	$^{9}H\,^{4}C$	1.0811 Å	0.0004
	$^{10}H\,^{5}C$	1.0923 Å	0.0005
	$^{3}C\,^{2}C\,^{1}H$	121.24°	0.0027
	$^{4}C\,^{3}C\,^{2}C$	103.53°	0.0017
$CH_2CHCHCH_2$	$^{5}C\,^{4}C\,^{3}C$	103.51°	0.0010
	$^{6}H\,^{5}C\,^{3}C$	121.23°	0.0019
	$^{7}H\,^{2}C\,^{3}C$	123.11°	0.0011
	$^{8}H\,^{3}C\,^{2}C$	128.89°	-0.0011
	$^{9}H\,^{4}C\,^{3}C$	128.90°	0.0036
	$^{10}H\,^{5}C\,^{4}C$	123.11°	0.0016
	$^{4}C\,^{3}C\,^{2}C\,^{1}H$	-142.18°	-0.0008
	$^{5}C\,^{4}C\,^{3}C\,^{2}C$	21.73°	-0.0091
	$^{6}H\,^{5}C\,^{4}C\,^{3}C$	-142.45°	-0.0017
	$^{7}H\,^{2}C\,^{3}C\,^{4}C$	60.58°	-0.0051
	$^{8}H\,^{3}C\,^{2}C\,^{1}H$	-150.83°	0.00387
	$^{9}H\,^{4}C\,^{3}C\,^{2}C$	-150.82°	-0.0024
	$^{10}H\,^{5}C\,^{4}C\,^{3}C$	60.52°	-0.0050

Table II Computed Transition States

Reaction	Heat of Formation Kcal/mol	RMS Gradient Kcal/unit [a]	Activation Energy Kcal/mol	Thiel et al Energy Kcal/mol	P_r	P_g
A	110.47	0.0102	95.11	95.1	0.61461	0.75252
B	140.11	0.0112	124.75	124.8	0.65713	0.55056
C	164.75	0.0237	149.40	149.4	0.52309	0.54599
D	140.73	0.0176	81.67	81.7	0.46644	0.56458
E	151.74	0.0135	94.40	94.4	0.61253	0.65209
F	75.77	0.0186	108.74	108.7	0.53687	0.58754
G	61.92	0.0200	94.85	96.5	0.67676	0.41424
H	34.03	0.0057	20.41	20.4	-1.1007	0.48830
I	123.02	0.0187	109.55	128.4	0.42946	0.87060
J	77.96	0.0098	135.32	135.3	0.46631	0.80675
K	51.81	0.0107	109.14	109.4	0.51078	0.75719
L	43.85	0.0162	101.21	101.2	0.60039	0.42205
M	-20.24	0.0171	72.32	68.5	0.49992	0.65407
N	75.52	0.0121	49.51		0.52179	0.29943
P	80.80	0.0029	51.43		0.50515	0.55318

[a] Kcals/coordinate unit, as defined in the text.

Table III. Coordinates of Reactants, Transition States and Products in Angstroms.

	REACTANT			TRANSITION STATE			PRODUCT		
	X	Y	Z	X	Y	Z	X	Y	Z
A) ETHYLENE --> METHYL CARBENE									
^1H	-1.423	0.298	0.560	-1.376	0.312	0.265	-1.428	-0.292	0.376
^2C	-0.617	-0.245	0.068	-0.584	-0.222	-0.219	-0.624	-0.834	-0.114
^3C	0.617	0.245	-0.068	0.731	0.124	-0.446	0.694	-0.202	-0.209
^4H	0.914	1.226	0.300	1.137	1.137	-0.312	0.634	0.756	-0.773
^5H	-0.913	-1.227	-0.299	0.488	-0.702	0.549	1.100	0.020	0.803
^6H	1.423	-0.298	-0.560	1.376	-0.479	-1.100	1.428	-0.852	-0.730
B) ETHYLENE --> VINYLIDENE + H2									
^1H	-1.100	0.707	-0.847	0.0	0.0	0.0	0.0	0.0	0.0
^2C	-0.578	-0.091	-0.321	1.089	0.0	0.0	1.089	0.0	0.0
^3C	0.578	0.091	0.321	1.883	1.054	0.0	1.807	1.105	0.0
^4H	1.088	1.052	0.369	0.897	2.414	0.0	0.75	3.3	0.0
^5H	1.100	-0.707	0.847	1.706	2.325	0.0	1.2	2.8	0.0
^6H	-1.089	-1.052	-0.369	1.524	-1.000	0.0	1.533	-0.995	0.0
C) ETHYLENE --> ACETYLENE + H2									
^1H	0.0	0.0	0.0	0.0	0.0	0.0	-2.245	0.1	0.0
^2C	1.089	0.0	0.0	1.051	0.0	0.0	-1.19	0.0	0.0
^3C	1.820	1.117	0.0	2.252	0.413	0.0	0.0	0.0	0.0
^4H	1.383	2.115	0.0	2.936	1.244	-0.001	1.051	0.15	0.0
^5H	2.909	1.117	0.0	3.196	-0.588	0.000	0.3	-2.4	0.0
^6H	1.525	-0.998	0.0	2.280	-0.955	0.004	-0.328	-2.0	0.0
D) METHYL CARBENE --> ACETYLENE + H2									
^1H	0.0	0.0	0.0	0.0	0.0	0.0	-2.245	0.01	0.0
^2C	1.087	0.0	0.0	1.048	0.0	0.0	-1.194	0.0	0.0
^3C	1.791	1.285	0.0	2.295	0.288	0.0	0.0	0.0	0.0
^4H	1.522	1.882	-0.900	2.656	1.413	-0.506	0.1	2.0	-0.3
^5H	1.522	1.882	0.901	2.656	1.414	0.503	0.0	2.281	0.3
^6H	2.892	1.153	-0.001	3.200	-0.319	0.001	1.051	-0.01	0.0
E) ACETYLENE --> VINYLIDENE									
^1H	-2.245	0.2	0.0	-1.119	0.726	0.001	0.0	0.0	0.0
^2C	-1.194	0.0	0.0	-0.087	0.489	0.001	1.089	0.0	0.0
^3C	0.0	0.0	0.0	1.183	0.488	0.001	1.807	1.105	0.0
^4H	1.051	-0.2	0.0	0.607	-0.699	-0.001	1.533	-0.995	0.2
F) Trans-HYDROXY CARBENE --> FORMALDEHYDE									
^1H	-1.267	-0.617	0.010	0.0	0.0	0.0	-0.889	-0.013	0.005
^2C	-0.158	-0.618	0.010	1.098	0.0	0.0	0.217	0.006	0.005
^3O	0.315	0.593	0.010	1.768	1.095	0.0	0.871	1.032	0.005
^4H	1.267	0.617	-0.010	2.402	0.021	0.0	0.667	-1.005	-0.005

Table III (Continued)

	REACTANT			TRANSITION STATE			PRODUCT		
	X	Y	Z	X	Y	Z	X	Y	Z
G) FORMALDEHYDE --> CARBON MONOXIDE + HYDROGEN									
^1H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
^2H	0.803	0.0	0.0	0.039	1.141	0.0	1.538	-1.019	0.001
^3C	-0.165	1.975	0.0	1.060	0.0	0.0	1.106	0.0	0.0
^4O	0.830	2.724	0.0	0.533	2.213	0.000	1.778	1.014	0.0
H) Trans-HYDROXY CARBENE --> Cis-HYDROXY CARBENE									
^1H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
^2C	1.110	0.0	0.0	1.127	0.0	0.0	1.113	0.0	0.0
^3O	1.583	1.211	0.0	1.601	1.240	0.0	1.703	1.135	0.0
^4H	2.535	1.235	0.020	1.799	1.685	0.806	1.166	1.932	0.010
I) Cis-HYDROXY CARBENE --> CARBON MONOXIDE + HYDROGEN									
^1H	-1.114	-0.501	0.048	0.0	0.0	0.0	-0.198	-0.599	0.003
^2C	-0.610	0.486	-0.056	1.478	0.0	0.0	-0.581	1.065	-0.013
^3O	0.669	0.501	-0.045	1.527	1.253	0.0	0.581	1.067	0.006
^4H	1.114	-0.378	0.056	0.258	1.026	0.0	0.273	-1.067	0.013
J) METHANOL --> FORMALDEHYDE + HYDROGEN									
^1H	-1.109	0.014	-0.539	0.0	0.0	0.0	-0.994	0.535	-0.813
^2C	-0.256	-0.111	0.179	1.114	0.0	0.0	-0.191	0.857	-0.122
^3O	0.981	0.040	-0.440	1.756	1.147	0.0	0.994	0.741	-0.370
^4H	1.109	-0.602	-1.124	1.805	0.375	-1.231	0.630	-1.429	0.735
^5H	-0.369	-1.112	0.665	1.338	-0.504	-1.290	0.173	-0.961	0.632
^6H	-0.347	0.675	0.967	1.519	-0.850	0.596	-0.580	1.294	0.816
K) METHANOL --> METHYLENE + WATER									
^1H	0.0	0.0	0.0				0.0	0.0	0.0
^2C	1.115	0.0	0.0				1.094	0.0	0.0
^3O	1.546	1.322	0.0				1.134	2.946	0.0
^4H	2.491	1.380	0.002				2.035	3.216	0.029
^5H	1.466	-0.562	0.902				0.783	3.235	0.823
^6H	1.467	-0.562	-0.902				1.538	-0.024	-1.000
	1st TRANSITION STATE			INTERMEDIATE			TRANSITION STATE		
^1H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
^2C	1.094	0.0	0.0	1.096	0.0	0.0	1.095	0.0	0.0
^3O	1.654	1.321	0.0	1.482	1.528	0.0	1.394	2.139	0.0
^4H	2.019	1.654	-0.824	2.412	1.597	-0.208	2.331	2.258	-0.007
^5H	2.128	0.683	0.781	1.371	1.868	0.880	1.085	2.432	0.844
^6H	1.524	-0.676	-0.749	1.527	-0.388	-0.931	1.475	-0.206	-1.005

Table III. (Continued)

	REACTANT			TRANSITION STATE			PRODUCT		
	X	Y	Z	X	Y	Z	X	Y	Z
L) METHANOL --> Trans-HYDROXY CARBENE + HYDROGEN									
^{1}H	-0.894	0.158	-0.900	-1.230	0.488	0.412	-0.296	0.843	-2.243
^{2}C	-0.288	-0.138	-0.012	-0.136	0.523	0.346	0.276	0.813	-1.293
^{3}O	0.448	0.969	0.396	0.321	0.904	-0.842	1.462	1.327	-1.441
^{4}H	0.986	0.772	1.149	1.230	1.179	-0.805	1.974	1.312	-0.638
^{5}H	-0.986	-0.500	0.784	0.419	-0.461	0.958	-1.974	-0.675	2.228
^{6}H	0.364	-0.999	-0.303	0.091	-1.114	0.501	-1.735	-1.272	2.066
M) FORMIC ACID DEGENERATE REARRANGEMENT									
^{1}H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
^{2}C	1.105	0.0	0.0	1.087	0.0	0.0	1.105	0.0	0.0
^{3}O	1.626	1.250	0.0	1.913	0.997	0.0	1.839	0.983	0.0
^{4}H	2.573	1.309	0.0	2.776	0.000	0.0	2.573	-1.309	0.0
^{5}O	1.839	-0.983	0.0	1.913	-0.997	0.0	1.626	-1.250	0.0
N) CYCLOPROPYL ANION --> PROPYLENE ANION									
^{1}H	-1.352	-1.161	0.071	-1.784	0.298	0.576	-2.161	0.252	-0.102
^{2}C	-0.780	-0.287	0.454	-0.925	-0.161	0.066	-1.238	-0.228	0.211
^{3}C	-0.027	0.599	-0.400	0.100	0.564	-0.555	-0.000	0.228	-0.196
^{4}C	0.769	-0.292	0.410	0.945	-0.145	0.306	1.261	-0.249	0.096
^{5}H	1.312	-1.169	-0.004	1.029	0.029	1.390	1.452	-1.106	0.733
^{6}H	-1.310	0.055	1.369	-1.120	-1.233	-0.089	-1.382	-1.084	0.862
^{7}H	-0.045	1.367	-1.119	0.192	1.307	-1.320	-0.023	1.113	-0.861
^{8}H	1.352	0.047	1.294	1.784	-0.772	-0.031	2.161	0.217	-0.300
P) BUTADIENE --> CYCLOBUTENE									
^{1}C	-1.572	-0.449	0.240	-1.073	-0.614	0.391	0.828	-0.192	0.462
^{2}C	-0.733	0.461	-0.285	-0.721	0.664	-0.110	0.691	0.805	-0.683
^{3}C	0.731	0.463	-0.286	0.654	0.752	0.154	-0.662	0.752	-0.694
^{4}C	1.572	-0.429	0.265	1.053	-0.602	0.284	-0.739	-0.253	0.449
^{5}H	-2.654	-0.339	0.177	-1.800	-1.244	-0.126	1.276	0.207	1.388
^{6}H	2.654	-0.318	0.198	1.800	-0.903	1.022	-1.170	-1.235	0.189
^{7}H	-1.263	-1.353	0.762	-0.953	-0.898	1.439	1.339	-1.138	0.209
^{8}H	-1.168	1.333	-0.790	-1.348	1.339	-0.676	1.447	1.327	-1.239
^{9}H	1.163	1.322	-0.814	1.253	1.631	0.348	-1.447	1.214	-1.264
^{10}H	1.267	-1.319	0.812	0.950	-1.334	-0.519	-1.232	0.109	1.368

Figure A

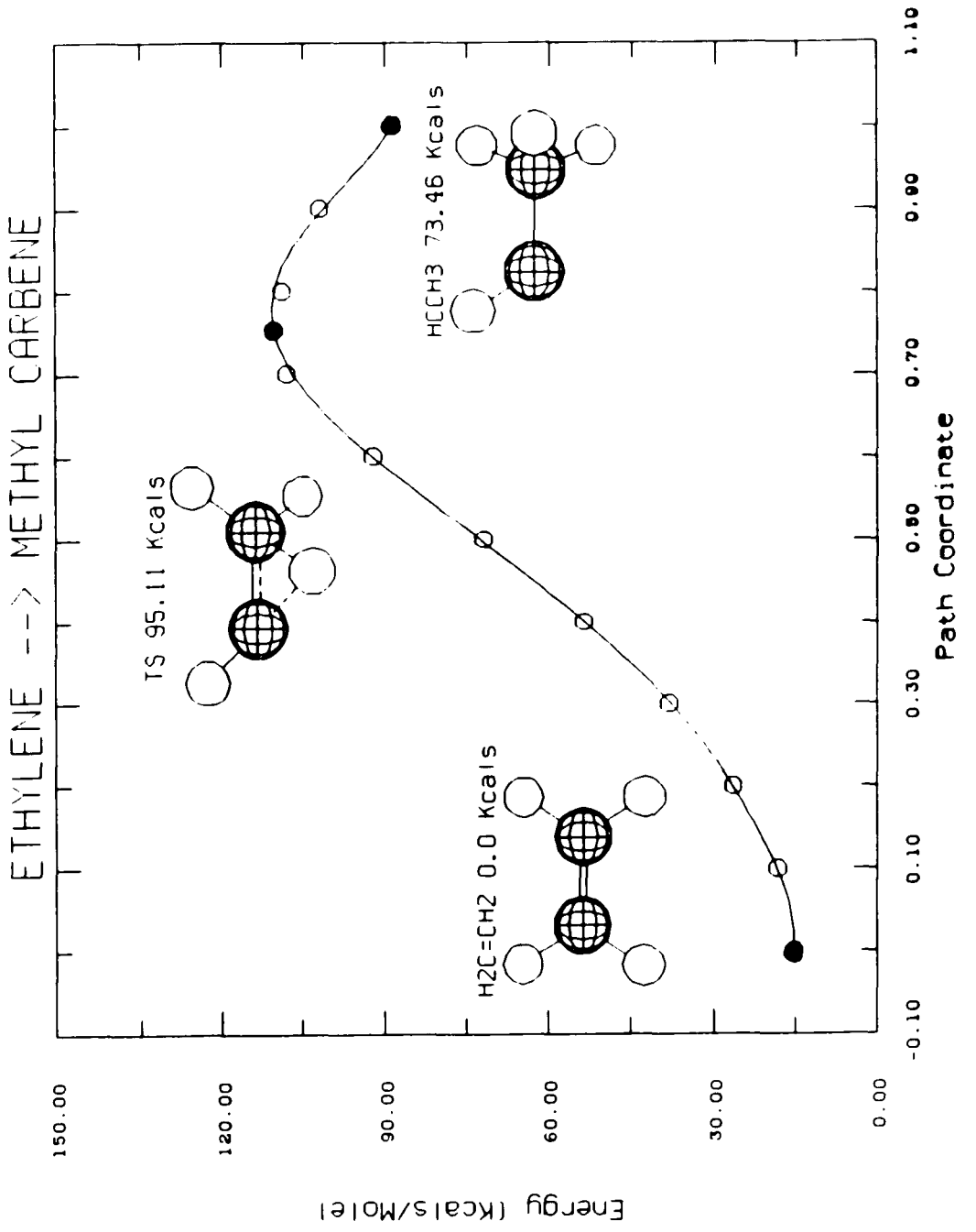
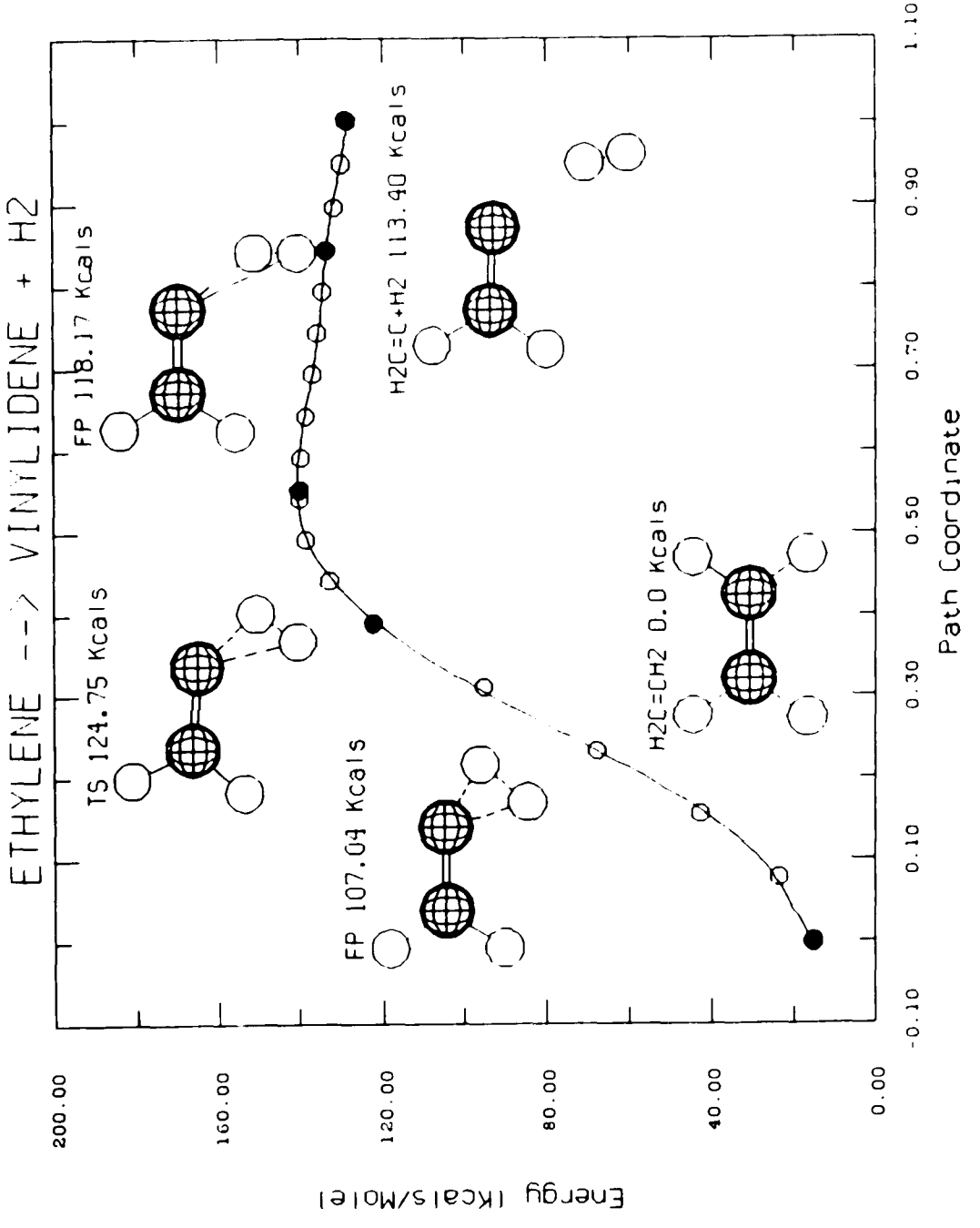

ETHYLENE --> METHYL CARBENE
TS 95.11 Kcals
HCCH3 73.46 Kcals
H2C=CH2 0.0 Kcals
Energy (Kcals/Mole)
150.00
120.00
90.00
60.00
30.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure 8


ETHYLENE --> VINYLIDENE + H2
TS 124.75 Kcals
FP 118.17 Kcals
H2C=C+H2 113.40 Kcals
FP 107.04 Kcals
H2C=CH2 0.0 Kcals
Energy (Kcals/Mole)
200.00
160.00
120.00
80.00
40.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

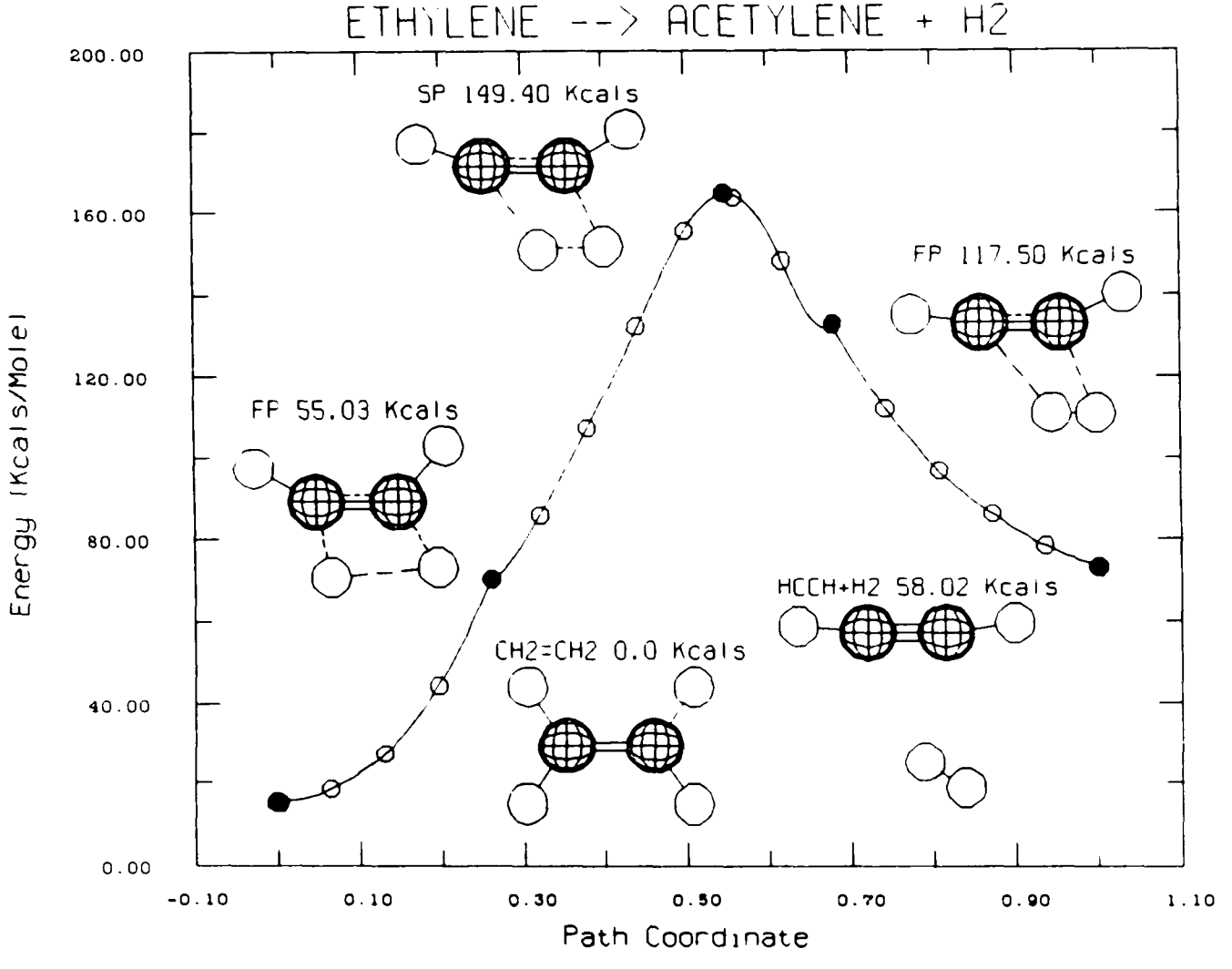

ETHYLENE --> ACETYLENE + H2
SP 149.40 Kcals
FP 117.50 Kcals
FP 55.03 Kcals
HCCH+H2 58.02 Kcals
CH2=CH2 0.0 Kcals
Energy (Kcals/Molel
200.00
160.00
120.00
80.00
40.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure C

Figure D

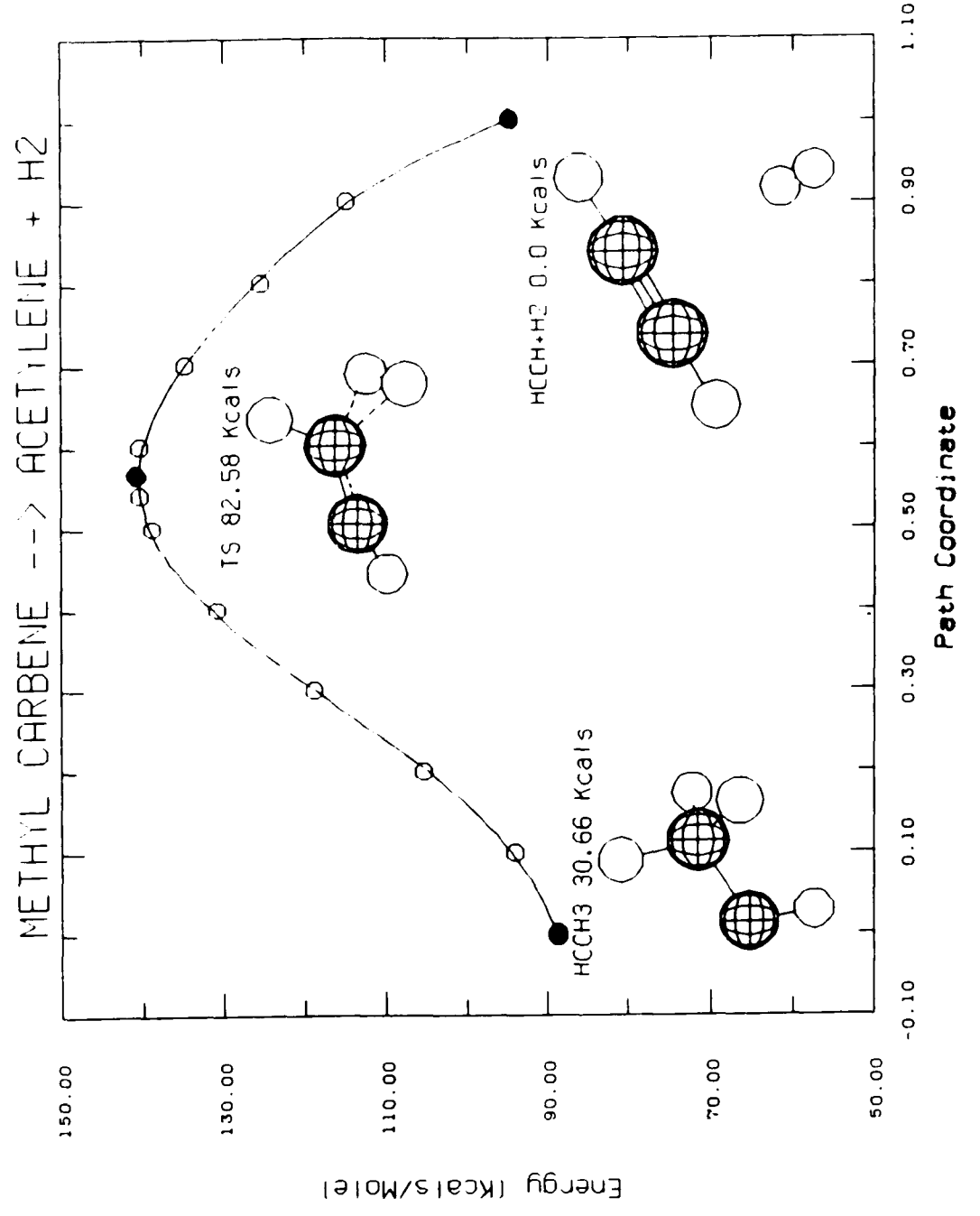

METHYL CARBENE --> ACETYLENE + H2
TS 82.58 Kcals
HCCH+H2 0.0 Kcals
HCCH3 30.66 Kcals
Energy (Kcals/Mole)
150.00
130.00
110.00
90.00
70.00
50.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure E

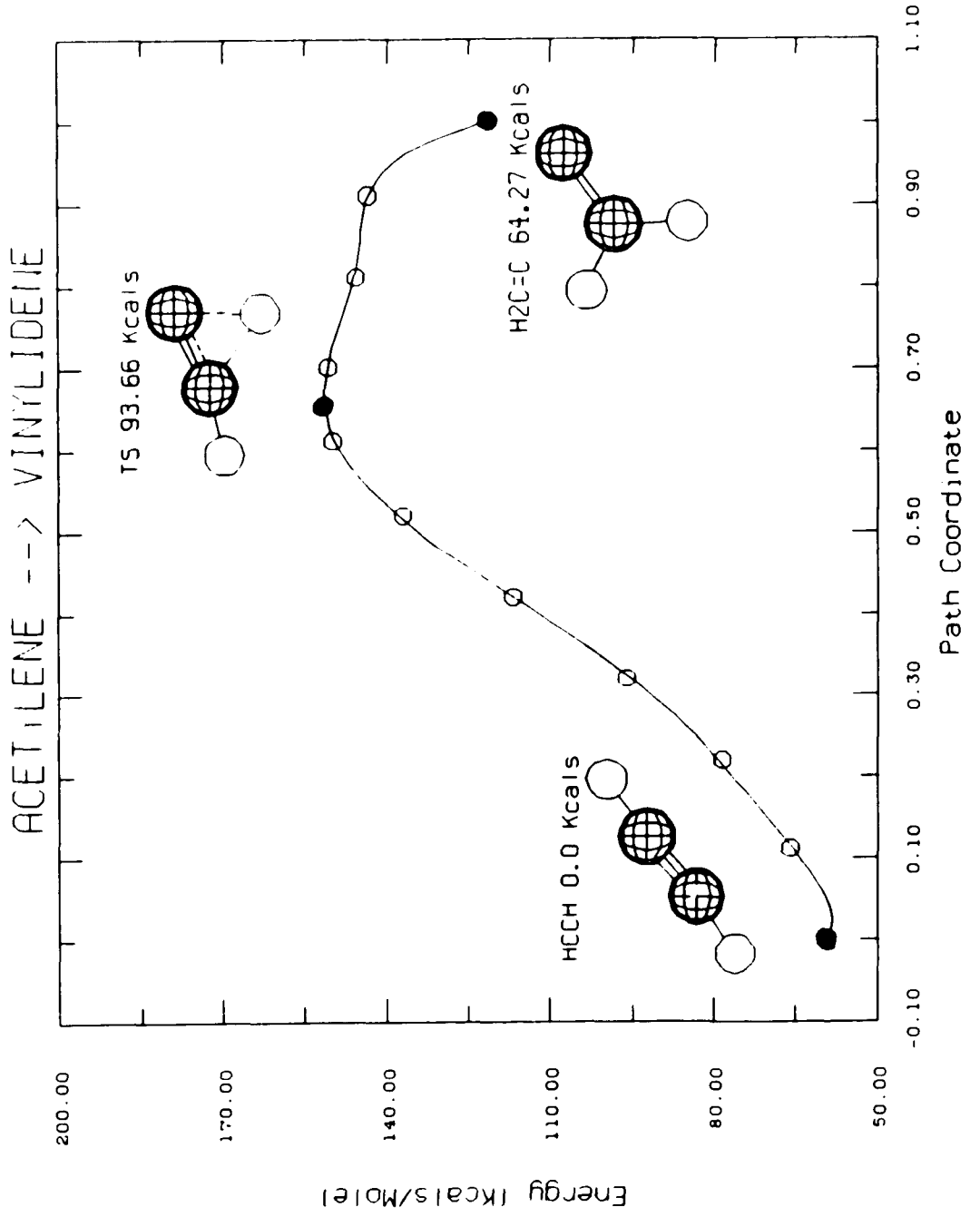

ACETYLENE --> VINYLIDENE
TS 93.66 Kcals
H2C=C 64.27 Kcals
HCCH 0.0 Kcals
Energy (Kcals/Mole)
Path Coordinate
200.00
170.00
140.00
110.00
80.00
50.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10

Figure F

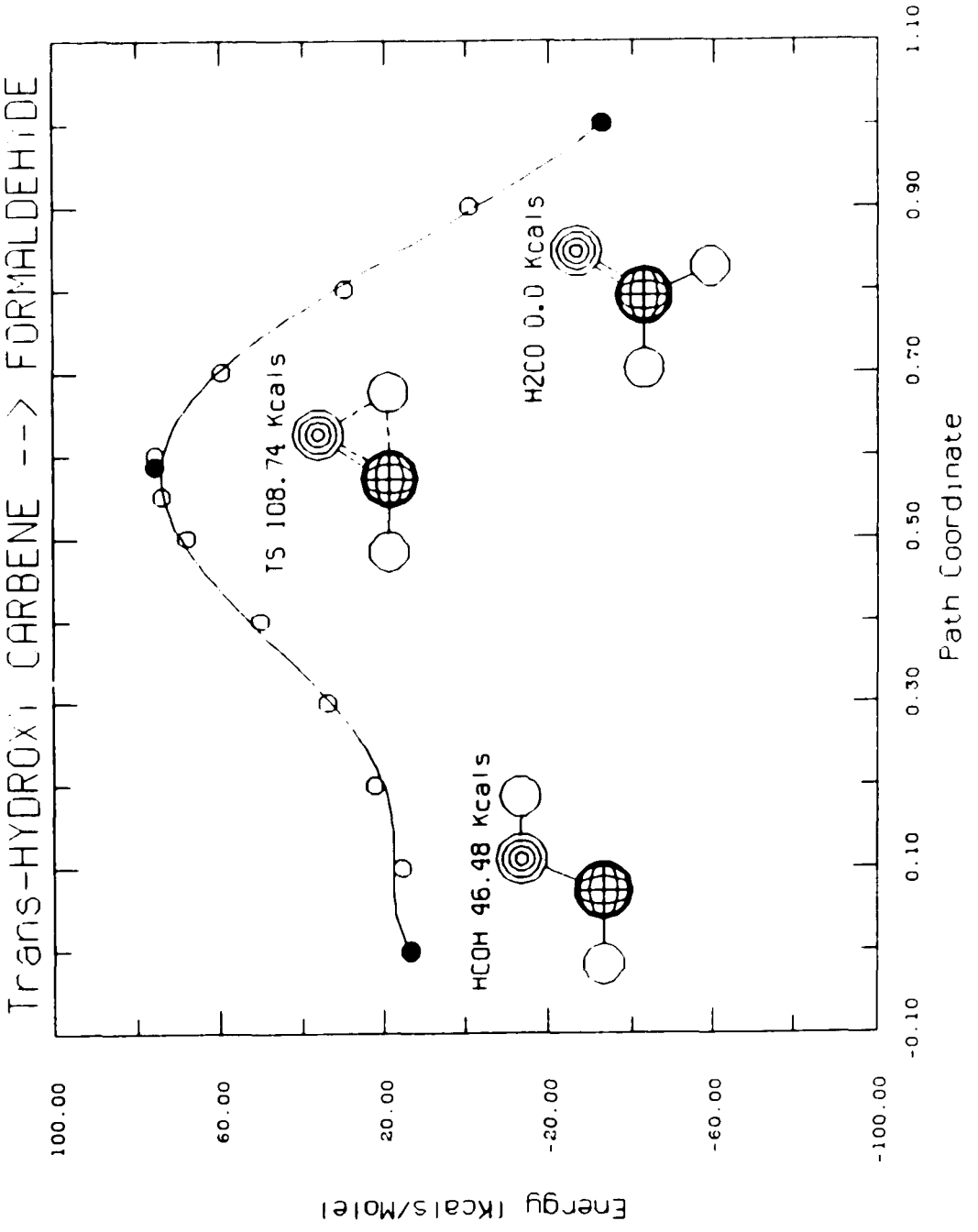

Trans-HYDROXY CARBENE --> FORMALDEHYDE
TS 108.74 Kcals
H2CO 0.0 Kcals
HCOH 46.48 Kcals
Energy (Kcals/Mole)
100.00
60.00
20.00
-20.00
-60.00
-100.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure G

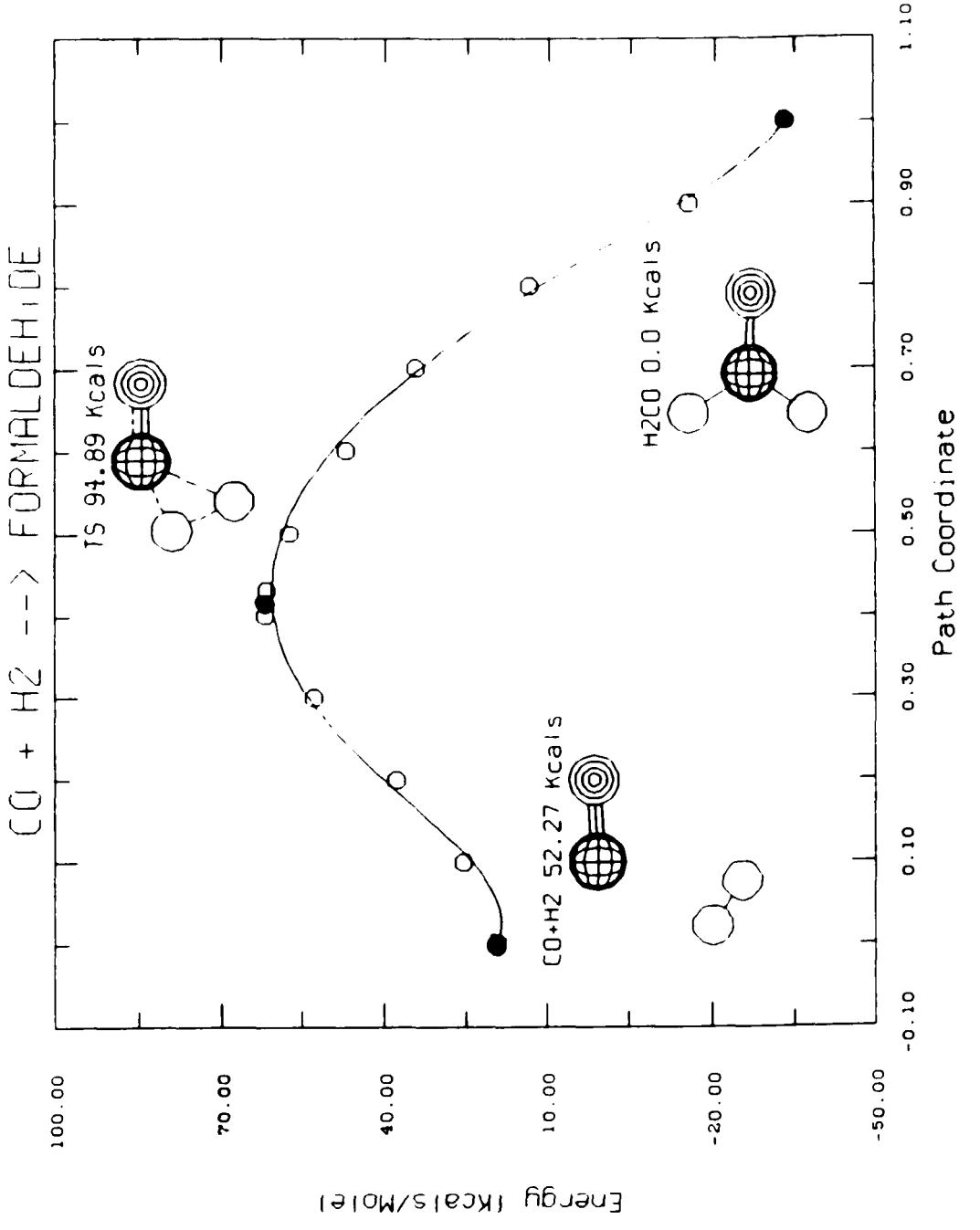

CO + H2 --> FORMALDEHYDE
TS 94.89 Kcals
H2CO 0.0 Kcals
CO+H2 52.27 Kcals
Energy (Kcals/Mole)
Path Coordinate
100.00
70.00
40.00
10.00
-20.00
-50.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10

Figure H

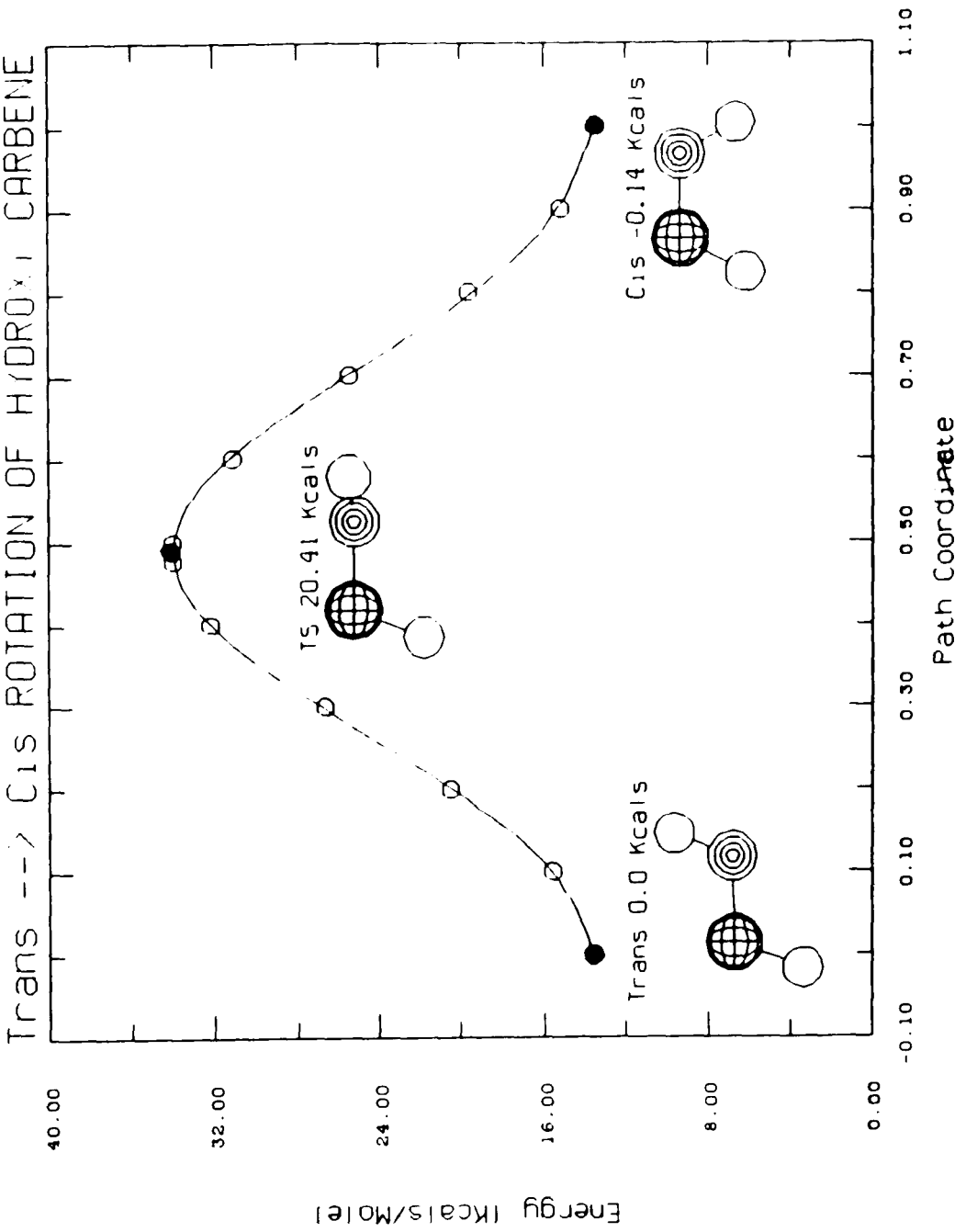

Trans --> Cis ROTATION OF HYDROXYL CARBENE
Energy (Kcals/Mole)
40.00
32.00
24.00
16.00
8.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate
Trans 0.0 Kcals
TS 20.41 Kcals
Cis -0.14 Kcals

Figure I

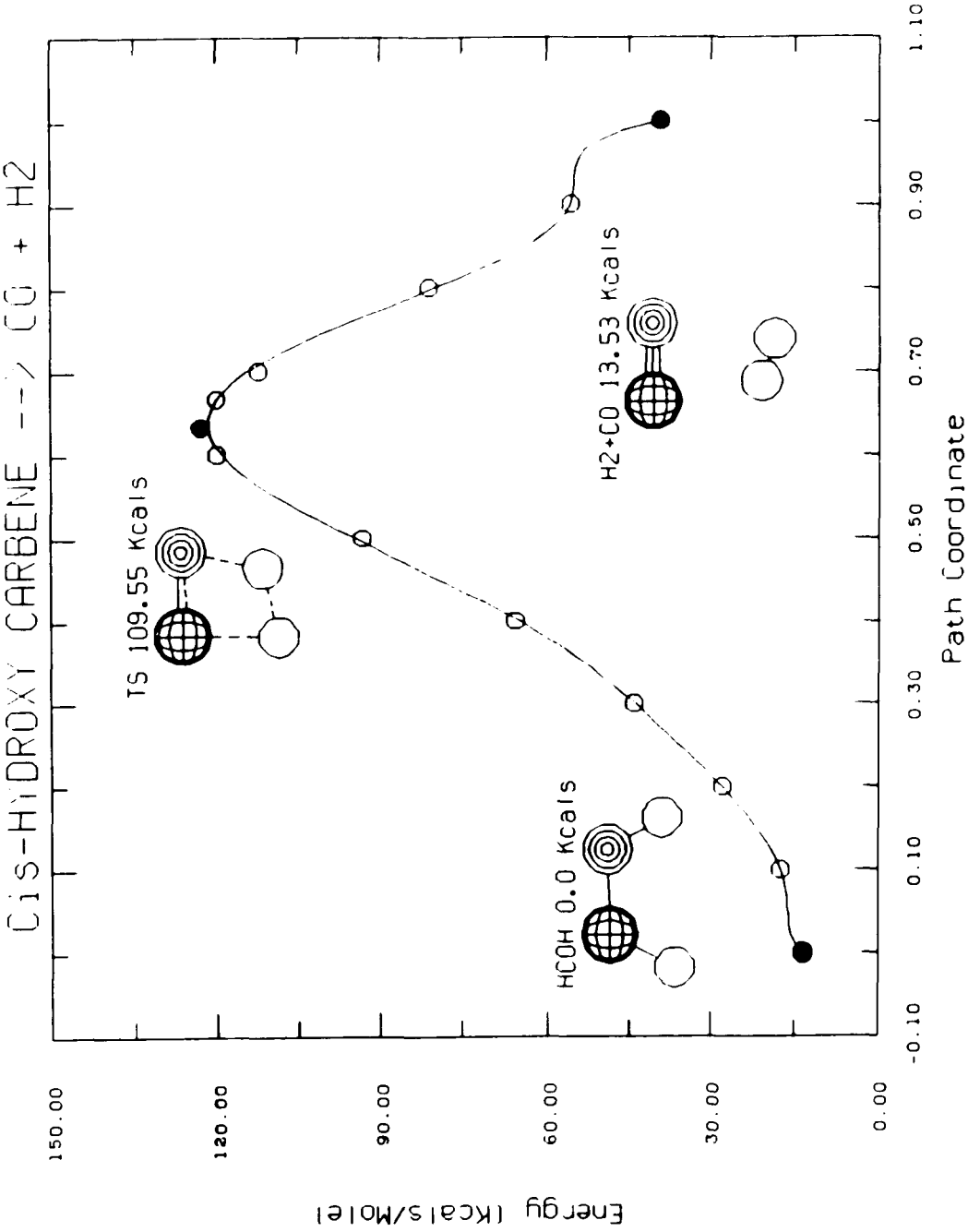

Cis-HYDROXY CARBENE --> CO + H2
TS 109.55 Kcals
H2+CO 13.53 Kcals
HCOH 0.0 Kcals
Energy (Kcals/Mole)
Path Coordinate
150.00
120.00
90.00
60.00
30.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10

Figure J

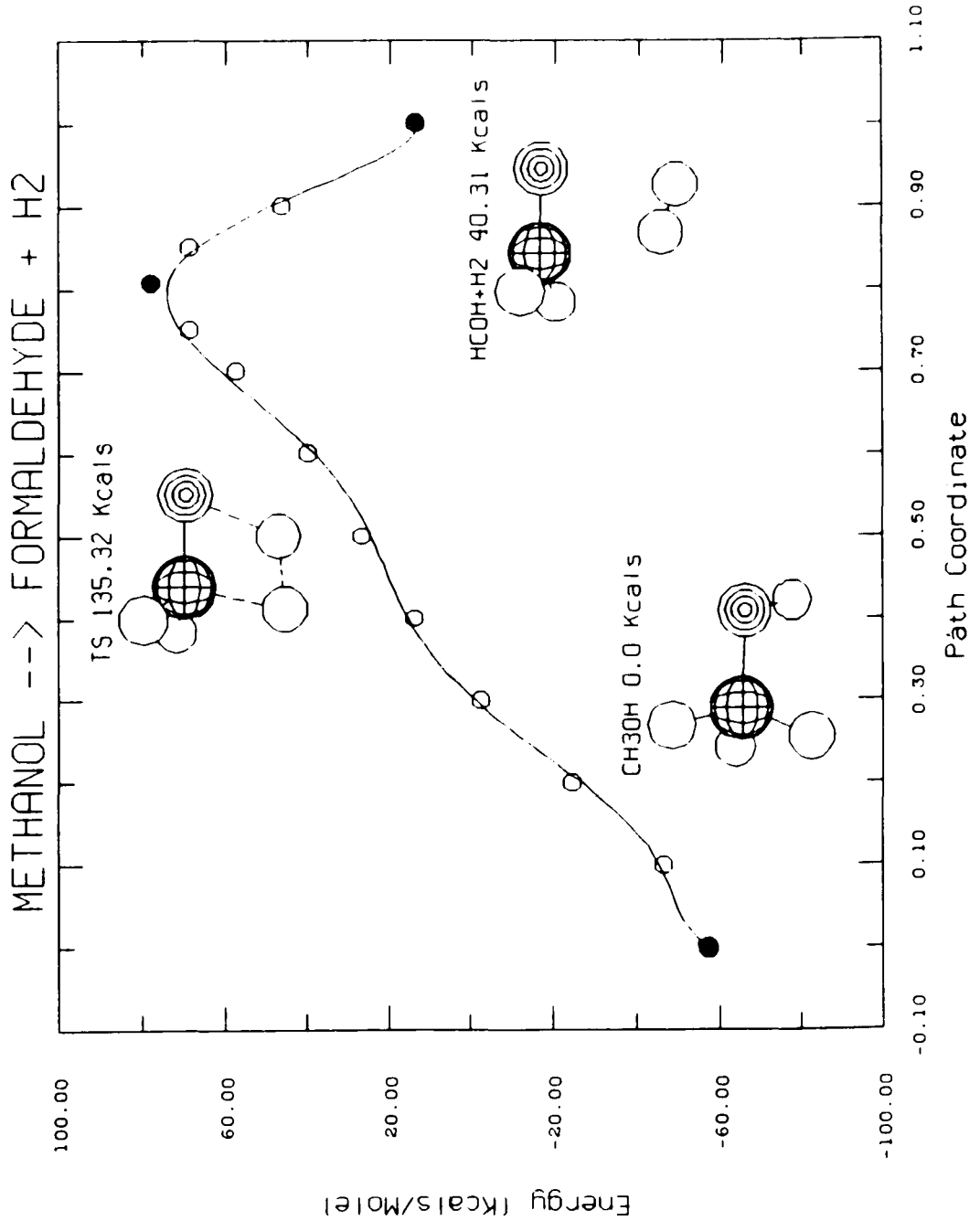

METHANOL --> FORMALDEHYDE + H2
TS 135.32 Kcals
HCOH+H2 40.31 Kcals
CH3OH 0.0 Kcals
Energy (Kcals/Mole)
100.00
60.00
20.00
-20.00
-60.00
-100.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure K

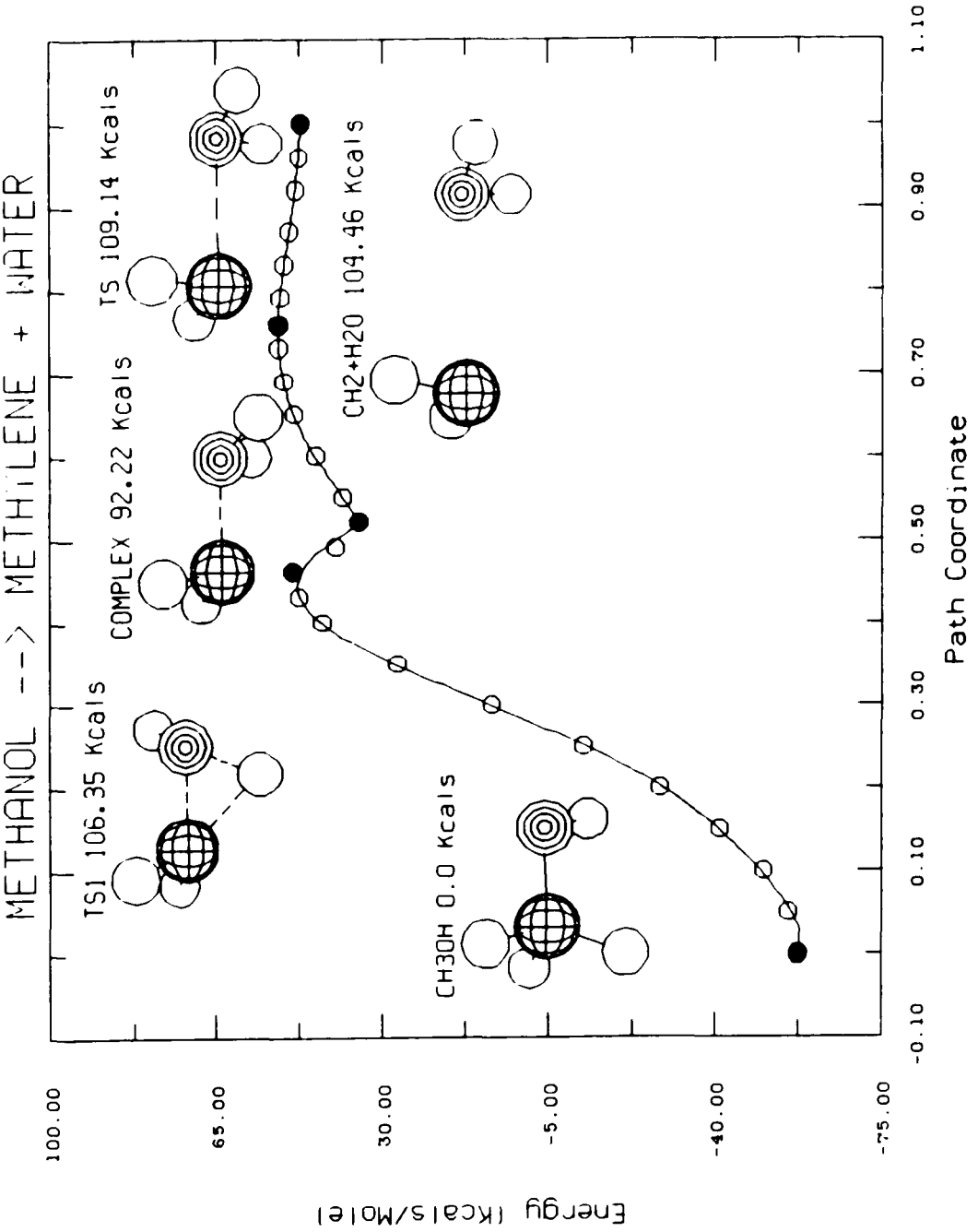

METHANOL --> METHYLENE + WATER
TS1 106.35 Kcals
COMPLEX 92.22 Kcals
TS 109.14 Kcals
CH2+H2O 104.46 Kcals
CH3OH 0.0 Kcals
100.00
65.00
30.00
-5.00
-40.00
-75.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Energy (Kcals/Mole)
Path Coordinate

Figure L

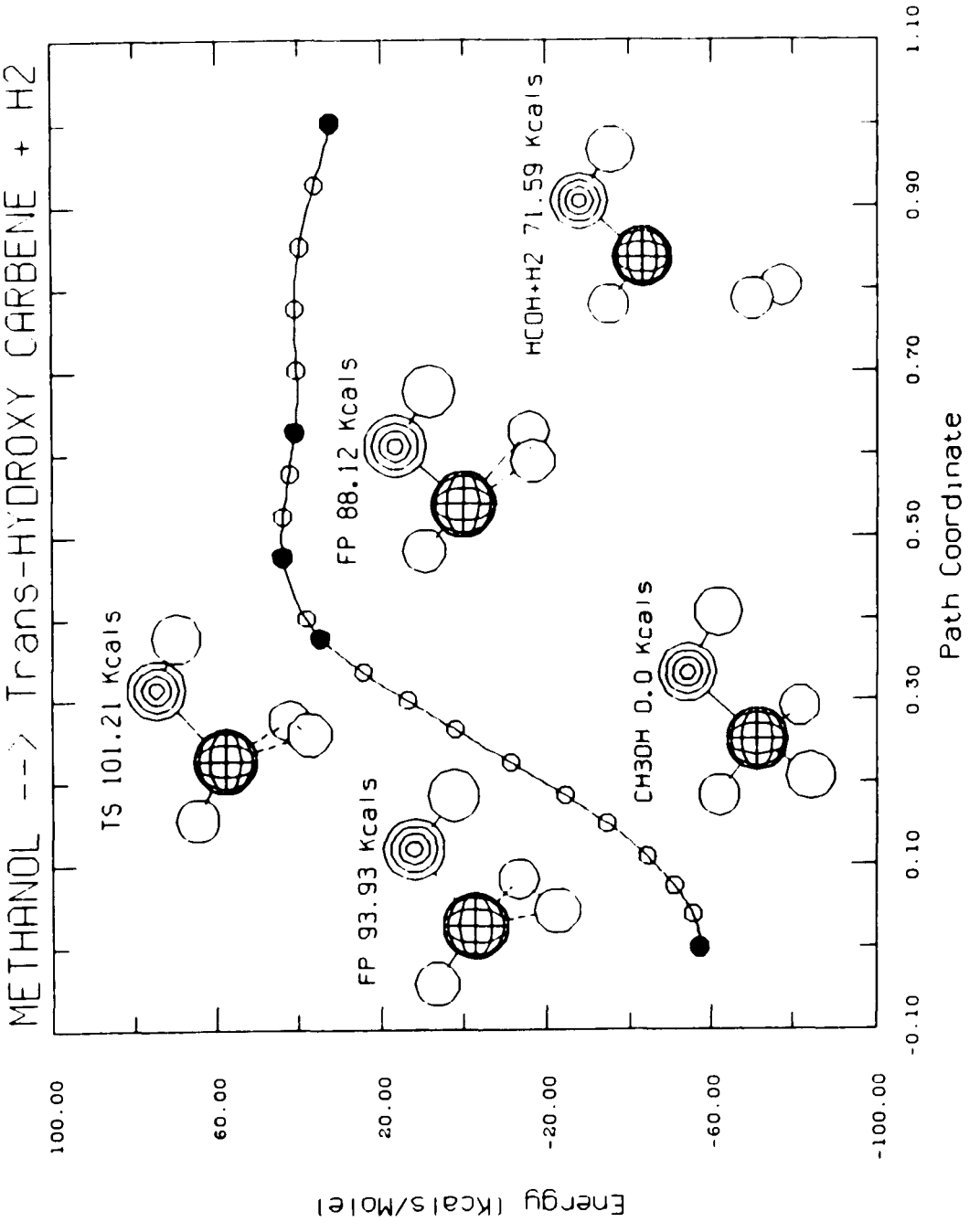

METHANOL --> Trans-HYDROXY CARBENE + H2
TS 101.21 Kcals
FP 88.12 Kcals
HCOH+H2 71.59 Kcals
FP 93.93 Kcals
CH3OH 0.0 Kcals
Energy (Kcals/Mole)
100.00
60.00
20.00
-20.00
-60.00
-100.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Figure M

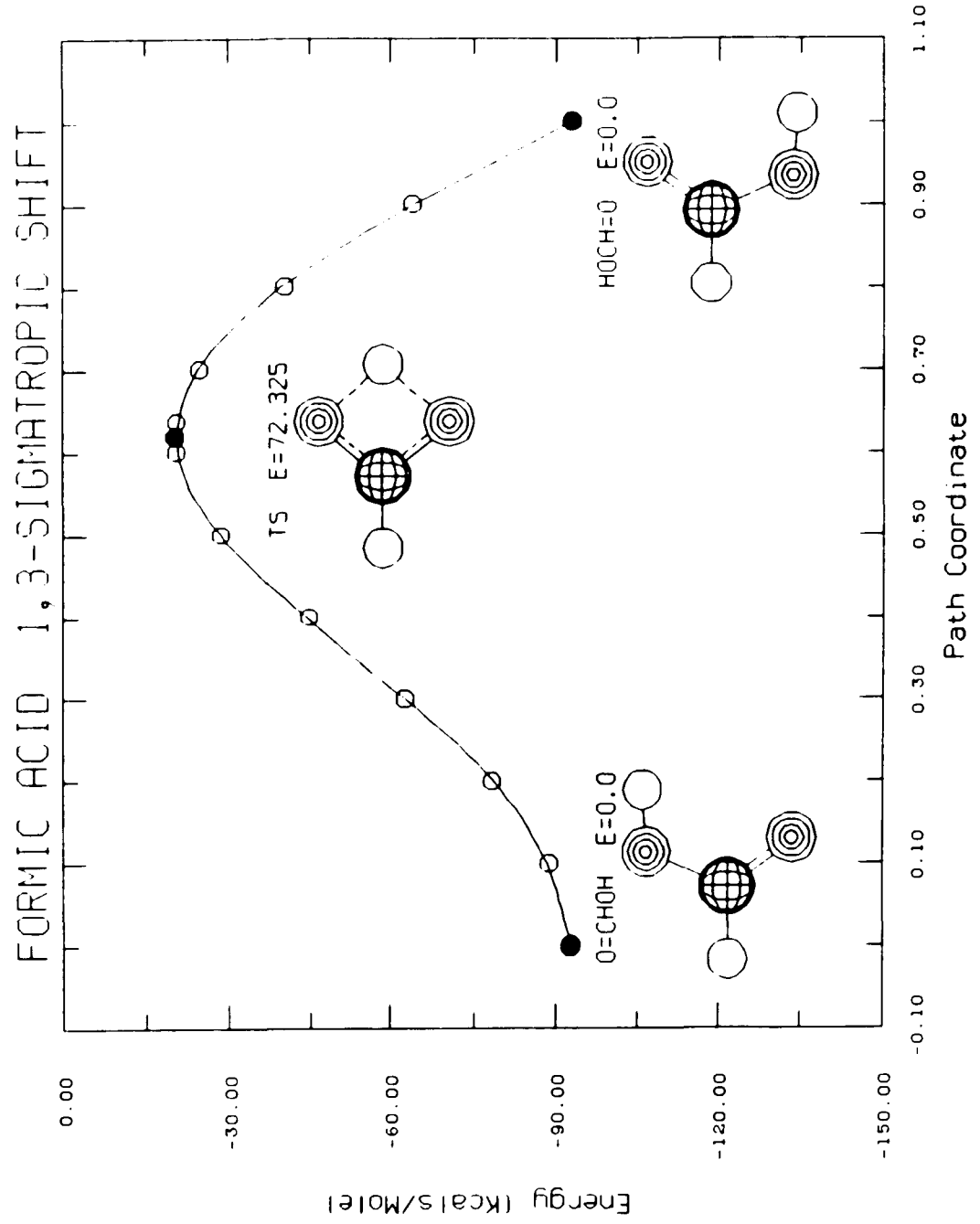

FORMIC ACID 1,3-SIGMATROPIC SHIFT
Energy (Kcals/Mole)
0.00
-30.00
-60.00
-90.00
-120.00
-150.00
Path Coordinate
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
O=CHOH E=0.0
TS E=72.325
HOCH=O E=0.0

Figure N

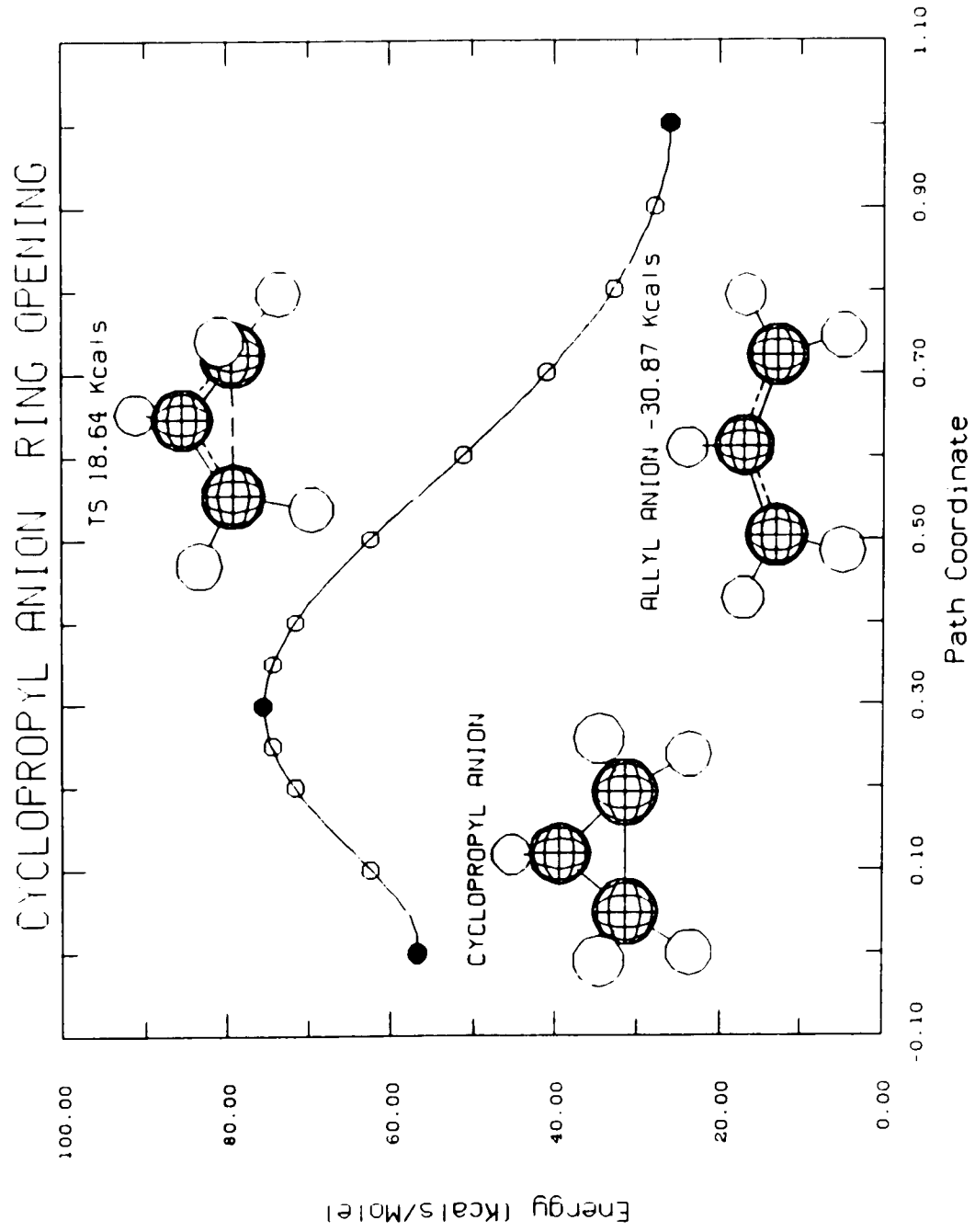
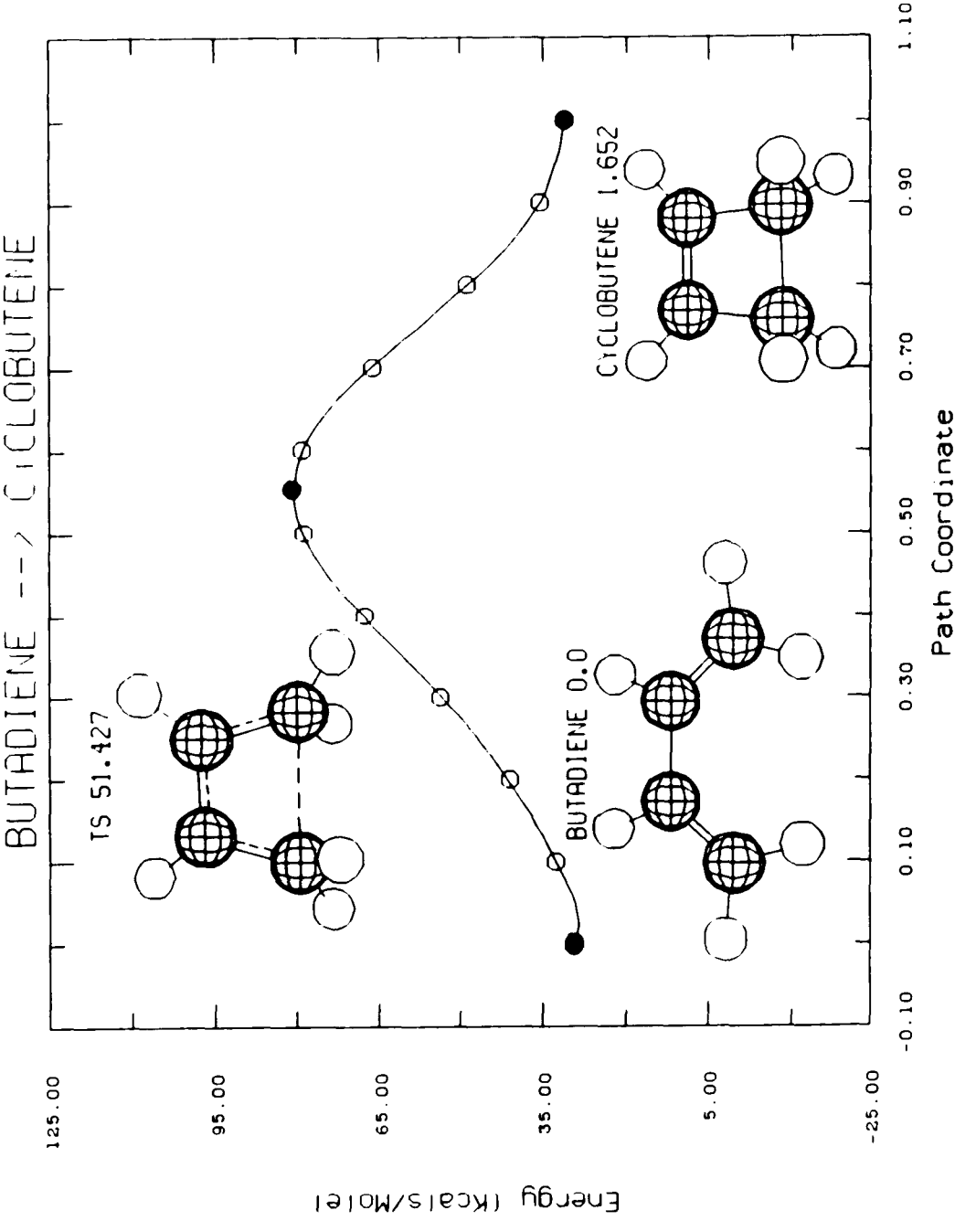

CYCLOPROPYL ANION RING OPENING
TS 18.64 Kcals
CYCLOPROPYL ANION
ALLYL ANION -30.87 Kcals
Energy (Kcals/Mole)
Path Coordinate
100.00
80.00
60.00
40.00
20.00
0.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10

Figure P


BUTADIENE --> CYCLOBUTENE
TS 51.427
BUTADIENE 0.0
CYCLOBUTENE 1.652
Energy (Kcals/Mole)
125.00
95.00
65.00
35.00
5.00
-25.00
-0.10
0.10
0.30
0.50
0.70
0.90
1.10
Path Coordinate

Structures of Transition States

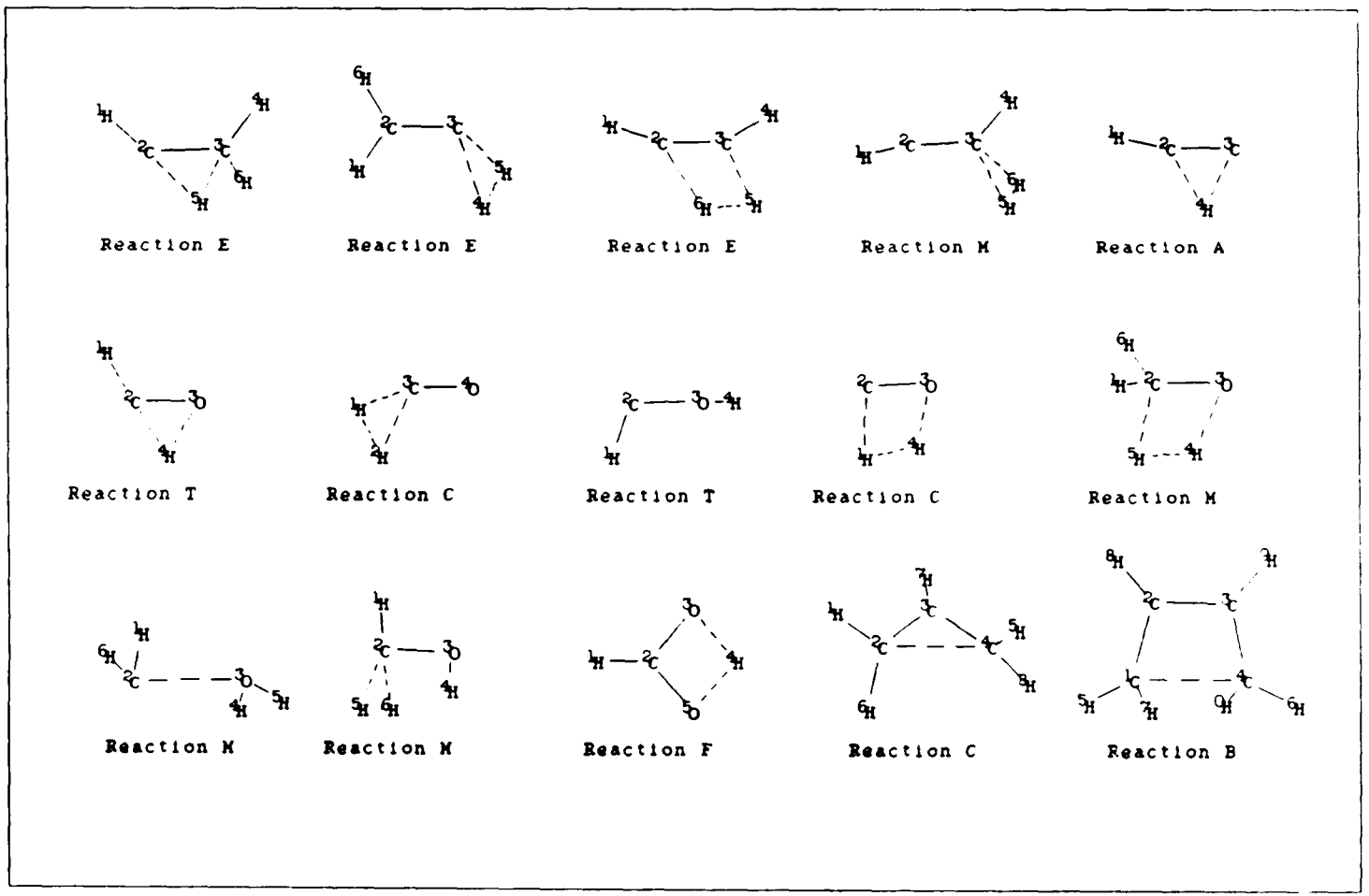


Figure 0

APPENDIX A

This appendix contains the key subroutines from MNDO and PRDDO which have to be changed or added in order to insert the Iso-Electronic Transition State search strategy into MNDO. Note that very few changes have to be made to MNDO.

The input is different, and a different subroutine is called for this.

Subroutine SYNTRN is called in the main routine to generate the starting structure. At the end of SYNTRN, the MNDO arrays are filled from the corresponding PRDDO-type arrays for the optimization. When PRDDO-type subroutines are called after this (for the path maximum searches), the PRDDO arrays are filled from the MNDO arrays.

The same array is used in both parts of the program to store the XYZ coordinates.

Also note that the COMMON blocks have been changed slightly to ensure compatability with both IBM and VAX hardware. On both systems, COMMON blocks will always start on a double word boundary. IBM also must have all double words on double word boundaries, and will pad with full words if a COMMON block alternates single word variables with double word variables. Since VAX systems will not do this padding, the single word variables have been moved to the end of the COMMON block.

THE CODE LISTED BELOW CONSISTS OF THE FOLLOWING ROUTINES:

1	MAIN	- main program of modified version
2	COMPFG	- computes functional value and gradient
3	SCF	- main routine for computing SCF energy
4	DRTP	- convert dE/dXYZ to dE/dRTP
5	INPUT	- input for MNDO data
6	DFPMIN	- structure optimizer
7	PTHCRD	- determines geometric path coordinate
8	ELECPC	- determines electronic path coordinate
9	PTHMAX	- finds maximum on LST or QST path
10	CPHF	- main routine for solution of CPHF equations
11	SCPHF	- called in solving CPHF equations
12	INV	- Gauss-Jordan inversion of square matrix
13	AX	- compute matrix product B = A B0
14	BONDS	- compute & store bonds in input structures
15	CONNEC	- find N-1 bonds needed to link structure
16	FILNVC	- fill NVC array
17	INPXYZ	- input XYZ coordinates for path calculations
18	SYNCAL	- called by SYNTRN
19	SYNTRN	- main routine in synchronous-transit
20	VCANON	- compute values for valence coordinates
21	VDIST	- compute RMS distance between structures

```
C  PROGRAM MNDO
C  BY WALTER THIEL  FACHBEREICH PHYSIKALISCHE CHEMIE DER PHILIPPS-
C  UNIVERSITAET  D-3550 MARBURG  WEST-GERMANY
C
C  PROGRAM TO CARRY OUT MNDO SCF CALCULATIONS AND TO LOCATE MINIMA
C  ON MNDO POTENTIAL SURFACES BY THE DFP ALGORITHM
C
C  REFERENCES
C  MNDO     M J S DEWAR AND W THIEL  J AM CHEM SOC  99  4899,
C           4907 (1977)
C  DFP      R FLETCHER AND M J D POWELL  COMP J  6  163 (1963)
C           W C DAVIDON  COMP J  10  406 (1968)
C  SYNTRN   T A HALGREN AND W N LIPSCOMB  CHEM PHYS LETT  49,
C           225  (1977)
C  CPHF     J A POPLE,R KRISHNAN,H B SCHLEGEL,J S BINKLEY,
C           I J QUANTUM CHEM  13  225  (1979)
C
C  THANKS ARE DUE TO ALL THOSE WHO WORKED ON EARLIER VERSIONS OF
C  SUBROUTINES IN THIS PROGRAM - AMONG THEM N C BAIRD, R C BINGHAM,
C  C E DOUBLEDAY, R C HADDON, H W KOLLMAR, D H LO, AND P K WEINER,
C  T A HALGREN, P BRENNAN
C
C  **********
C
C  OUTLINE OF INPUT
C
C  IN THIS SECTION, THE INPUT FOR THE PROGRAM IS SUMMARIZED BRIEFLY
C  A MORE DETAILED DESCRIPTION WILL BE GIVEN IN THE NEXT SECTION
C
C  THE INPUT CONSISTS OF THE FOLLOWING PARTS
C
C  1   TIME LIMIT CARD  THE PROGRAM WILL STOP A DFP OPTIMIZATION
C      IF THE TIME LIMIT IS APPROACHED  THE OPTIMIZATION CAN BE
C      CONTINUED IN ANOTHER JOB
C  2   ONE OR THREE CARDS FOR DFP OPTIONS  IT IS RECOMMENDED TO USE
C      THE DEFAULT VALUES  IN THIS CASE, ONLY THE FIRST DFP CARD IS
C      NEEDED (TO SPECIFY THE VARIABLES MAXEND AND MIDDLE)
C  3   CARDS FOR THE FIRST MOLECULE
C  3 1  TITLE CARD, INCLUDING OPTIONS
C  3 2  ATOMIC NUMBERS AND INTERNAL COORDINATES, ONE CARD PER ATOM,
C      WITH A BLANK CARD AT THE END
C  3 3  OPTION KSYM=1  SYMMETRY CONDITIONS, PART 1
C  3 4  OPTION KDEP=1  DEFINITION OF DEPENDENT PARAMETERS
C  3 5  OPTION KSYM=1  SYMMETRY CONDITIONS, PART 2
C  3 6  OPTION KCI =1  DATA FOR CONFIGURATION INTERACTION
C  4   CARDS FOR THE FOLLOWING MOLECULES (OPTIONAL)
C      THE PROGRAM ALLOWS COMPUTATIONS FOR AN ARBITRARY NUMBER
C      OF MOLECULES TO BE CARRIED OUT IN A SINGLE JOB  THE INPUT
```

```
C      FOR EACH NEW MOLECULE CONSISTS OF A NEW SET OF CARDS AS
C      DESCRIBED UNDER 3  A SIMPLIFIED INPUT FOR REACTION PATH
C      CALCULATIONS IS AVAILABLE  TOO
C 5    LAST CARD  THE INPUT IS TERMINATED BY A CARD WHICH HAS 99
C      PUNCHED IN COLUMNS 1-2
C
C INPUT FOR CONTINUATION JOBS
C IN ORDER TO CONTINUE AN UNCOMPLETED DFP OPTIMIZATION. THE INPUT
C DECK FOR THE CORRESPONDING MOLECULE IS RESUBMITTED AFTER SETTING
C MIDDLE=1 ON CARD 2  THE CONTINUATION JOB MAKES USE OF THE DFP
C INFORMATION SAVED ON FILE 4 BY THE PRECEDING JOB
C
C **********************
C
C DESCRIPTION OF INPUT
C
C COLUMNS  NAME   FORMAT       DESCRIPTION
C
C ***** TIME LIMIT CARD ****************************************
C
C   1-10 TLIMIT   F10 5   TIME LIMIT IN SECONDS. DEFAULT 100000.
C  11-20 SFDFP    F10 5   DFP SAFETY FACTOR  DEFAULT 1 5
C                 THE JOB WILL TERMINATE IF THE REMAINING
C                 TIME IS LESS THAN (SFDFP*TIME FOR THE
C                 PREVIOUS DFP CYCLE)
C  21-30 SFSCF    F10 5   DEFAULT 4 0
C
C ***** CARD FOR DFP OPTIONS ************************************
C
C   1-2  MAXEND    I2    MAXIMUM NUMBER OF SCF CALCULATIONS
C                 TO BE DONE FOR EACH MOLECULE
C                 THE MOLECULAR GEOMETRY WILL BE OPTIMIZED
C                 FOR MAXEND.EQ.0 AND FOR MAXEND.GT.2.
C                 =0 SET MAXEND=9999 (DEFAULT VALUE)
C                 =1 ONE SCF CALCULATION FOR EACH MOLECULE
C                 =2 ONE SCF CALCULATION FOR THE FIRST
C                   MOLECULE. THEN SET MAXEND=9999 FOR
C                   THE REMAINING MOLECULES (USEFUL FOR
C                   REACTION PATH CALCULATIONS)
C   3-4  IPRINT    I2    PRINT SWITCH  +K WILL PRINT GEOMETRY AND
C                 GRADIENT EVERY K-CYCLES. -K WILL PRINT
C                 THESE ALONG WITH THE SEARCH DIRECTIONS
C                 AND LINE MINIMIZATION INFORMATION EVERY
C                 K-CYCLES  DEFAULT=99
C   5-6  IREP     I2    =1 OPTIMIZED GEOMETRY IS SAVED ON FILE 7
C   7-8  IOPTC    I2    =1 READ DFP PARAMETERS ON THE FOLLOWING
C                   TWO CARDS
C                 FOR IOPTC NE 1 OMIT THE NEXT TWO CARDS
```

```
C                  FOR IOPTC NE 0 PRINT THE DFP OPTIONS
C    9-10 MIDDLE   I2    =0 NORMAL JOB
C                  =1 CONTINUATION OF A PREVIOUS JOB
C                    USING INFORMATION SAVED ON FILE 4
C   15-16 NWSKP    I2    =1 DETAILED PRINTING IN DFP OPTIMIZATION
C   19-20 JTDEL    I2    JTDEL*DELL=STEP SIZE IN RESTART SECTION
C                  WHEN ELEMENTS OF H MATRIX ARE NEGATIVE
C                  DEFAULT=6
C   21-30 PMSTE    F10 5  MAXIMUM STEP SIZE TAKEN AFTER RESTARTING
C                  H MATRIX  RECOMMEND A NUMBER 0 1 TO 0 5
C                  DEFAULT=0 1
C   31-40 DELL     F10 5  INCREMENT BY WHICH VARIABLES ARE CHANGED
C                  TO RESTART H MATRIX  THIS VALUE DECREASES
C                  AS ENERGY CHANGE DECREASES
C                  DEFAULT=0 01
C   41-80 KOMENT   10A4   COMMENT ON DFP OPTIMIZATION
C
C *** OMIT IF IOPTC NE.1 ****************************************
C  ***** SPECIAL OPTIONS FOR DFP AND CUBIC LINE SEARCH **************
C
C     1-2  MAXLIN   I2    NUMBER OF SCF CALCULATIONS ALLOWED FOR
C                    EACH LINE MINIMIZATION
C                    DEFAULT 15
C     3-4  IIN      I2    EXPONENT FOR LINE MINIMIZATION
C                    TOLERANCE (10**(-IIN))
C                    DEFAULT 4
C     5-6  IEND     I2    EXPONENT FOR DFP OPTIMIZATION
C                    TOLERANCE  (10**(-IEND))
C                    DEFAULT 4
C     7-8  NRST     I2    RESET H MATRIX IN THE DFP SECTION
C                    EVERY NRST CYCLES.
C                    DEFAULT 50
C     9-10 ISCF     I2    SCF TOLERANCE  (10**(-ISCF)) EV
C                    DEFAULT 4 FOR GEOMETRY OPTIMIZATION
C                    DEFAULT 5 FOR FIXED GEOMETRY
C    11-20 FAC      F10 5  LINE MINIMIZATION FACTOR
C                    DEFAULT 1 0
C    21-30 RST      F10 5  RESET H MATRIX WHENEVER THE COSINE OF
C                    THE ANGLE BETWEEN THE GRADIENT AND THE
C                    SEARCH DIRECTION IS LESS THAN RST
C                    DEFAULT 0 05
C    31-40 XFAC     F10 5  SCALING FACTOR FOR STEP SIZE IN GRADIENT
C                    CALCULATION
C                    DEFAULT 1 0 IN CUBIC SEARCH
C    46-50 IGG1     I5    IF AFTER CONVERGENCE IS ACHIEVED THERE
C                    ARE COMPONENTS OF THE GRADIENT GT 1.THE
C                    PROGRAM WILL TRY THIS MANY ADDITIONAL
C                    CYCLES TO GET ALL COMPONENTS LT 1
```

```
C                   DEFAULT=3
C    61-70 CCCOS    F10.5  THE VALUE OF RST UNTIL THE PREDICTED
C                   DECREASE IN ENERGY AS GIVEN BY
C                    ALPHA*P*G IS LE 1.0
C                   DEFAULT=MINIMUM(1.0/SQRT(NUMBER OF
C                   VARIABLES),0.15)
C    71-80 EYEAD    F10.5  CALCULATION ENDS WHEN -ALPHA*P*G LT EYEAD
C                   DEFAULT=0.005
C
C ***** OMIT IF IOPTC NE 1 ***************************************
C ***** SPECIAL OPTIONS FOR QUADRATIC LINE SEARCH ******************
C
C     1-5  MXCONT   I5    NUMBER OF SCF CALCULATIONS ALLOWED FOR
C                   EACH LINE SEARCH  DEFAULT 15
C     6-15 YMAXST   F10.5  VARIABLE USED TO DETERMINE MAXIMUM VALUE
C                   FOR ALPHA  DEFAULT 0.1
C    16-25 XCRIT    F10.5  CRITERION ON X  DEFAULT 0.0001
C    36-45 YFAC     F10.5  SCALING FACTOR FOR STEP SIZE IN GRADIENT
C                   CALCULATION.
C                   DEFAULT 100.0 IN QUADRATIC SEARCH.
C    46-50 ISCFF    I5    SCF TOLERANCE FOR CALCULATION OF GRADIENT
C                   (10**(-ISCFF)) EV.  DEFAULT 4
C    51-80 ITOL(I) 6I5    TOLERANCES (10**(-ITOL(I)))
C        I=1,6         DEFAULT 9,10,12,13,2,2
C
C ******  TITLE CARD FOR MOLECULE  ********************************
C
C   1-2  KHARGE   I2    MOLECULAR CHARGE
C                 =99 TO END THE RUN
C   3-4  IMULT    I2    MULTIPLICITY
C                 =0 GROUND STATE SINGLET
C                 =1 EXCITED SINGLET.
C                 =2 DOUBLET.
C                 =3 TRIPLET
C   5-6  KTRIAL   I2    INITIAL BOND ORDER MATRIX FOR SCF
C                 =0 STANDARD DIAGONAL MATRIX
C                 =1 BOND ORDER MATRIX IS IN THE COMPUTER
C                   FROM PREVIOUS RUN (USEFUL ALONG A
C                   REACTION PATH)
C                 =2 BOND ORDER MATRIX FROM DIAGONALIZATION
C                   OF THE CORE HAMILTONIAN
C                 =3 BOND ORDER MATRIX WILL BE READ IN
C                   FROM FILE 2.
C   7-8  KGEOM    I2    =0 THE TRIAL GEOMETRY IS READ IN
C                 =1 A NEW VALUE OF THE REACTION COORDINATE
C                   IS INPUT.  THE REST OF THE GEOMETRY IS
C                   TAKEN FROM THE PREVIOUS RUN.
C                 =2 NEW REACTION COORDINATE VALUE INPUT
```

```
C                   THE REST OF THE GEOMETRY IS ESTIMATED
C                   BY LINEAR EXTRAPOLATION FROM THE
C                   PREVIOUS TWO RUNS
C                  = 1 PROGRAM TERMINATES AFTER COMPUTING
C                    COORDINATES AND DISTANCES (USEFUL
C                    FOR CHECKING THE INPUT DATA)
C                  =4 READ XYZ IN ANGSTROMS
C                  =5 READ XYZ IN AOMIC UNITS
C   9-10  IPUBO    I2    =1 BOND ORDER MATRIX IS SAVED ON FILE 2.
C  11-12  IPUEV    I2    =1 EIGENVECTORS AND EIGENVALUES ARE
C                    SAVED ON FILE 1.
C  13-16  KITSCF   I4    NUMBER OF SCF ITERATIONS. DEFAULT=99
C  17-18  NPRINT   I2    =-1 PRINTS NO SCF-INFORMATION
C                  = 0 PRINTS C.P MATRICES. NET CHARGES.
C                    AND DIPOLE MOMENT (DEFAULT VALUE)
C                  =+1 PRINTS H.F MATRICES. TOO
C                  =+2 PRINTS ELECTRONIC ENERGY IN EACH
C                    SCF-ITERATION. TOO.
C                  =+3 PRINTS REPULSION INTEGRALS. TOO.
C  21-22  KSYM     I2    NON-ZERO IF SYMMETRY CONDITIONS ARE TO
C                  BE INPUT.
C  23-24  KDEP     I2    NON-ZERO IF DEPENDENT FUNCTIONS ARE TO BE
C                  INPUT. THESE DEPENDENT FUNCTIONS RELATE
C                  CHANGES IN THE PARAMETERS TO EACH OTHER.
C                  THE USER MUST SUPPLY THE REQUIRED
C                  EQUATIONS FOR EACH MOLECULAR SYSTEM
C                  INVESTIGATED. THESE ARE PART OF THE
C                  USER SUPPLIED SUBROUTINE DEPVAR.
C                  SOME STANDARD DEPENDENT FUNCTIONS ARE
C                  PREDEFINED IN SUBROUTINE HADDON.
C  25-26  KCI      I2    =0 NO CONFIGURATION INTERACTION
C                  =1 LIMITED CONFIGURATION INTERACTION
C                  FOR GROUND-STATE. CLOSED-SHELL 2*2 CI
C                  FOR EXCITED SINGLET. OPEN-SHELL 3*3 CI
C  27-28  NSTART   I2    FIRST SCF EXTRAPOLATION IN CYCLE NSTART.
C                  DEFAULT 4.
C  29-30  NSTEP    I2    FOLLOWING SCF EXTRAPOLATIONS EVERY NSTEP
C                  CYCLES. DEFAULT 4.
C  31-32  NTRAN    I2    OPTION FOR SYNCRONOUS TRANSIT CALCS
C                  0 - SINGLE POINT OPTIMIZATION
C                  1 - LINEAR SYNC TRANSIT
C                  2 - QUADRATIC SYNC TRANSIT
C    33-80  KTITLE  12A4   TITLE FOR THE MOLECULE
C
C ***** NTRAN NE 0 ****** PRDDO TYPE DATA SETS ************************
C
C  1-80   TITLE   80A1     SYNTRN TITLE CARD
C
```

```
C   1-5  NATOMS  I5        IF NATOMS LT 0 THE ATOMS USED TO DEFINE
C                          THE INTERNAL VARIABLES ARE SUPPLIED ON
C                          THE NEXT  NATOMS  CARDS FORMAT 4I2
C
C ***** (NTRAN+1) * NATOMS  CARDS MUST BE SUPPLIED **********************
C
C  4-5   NORBS   I2        NUMBER OF ORBITALS ON THE ATOM
C  6-15  CORE    F10.5      CORE CHARGE OF THE ATOM
C 16-45  X,Y,X  3F10.5      XYZ COORDS OF THE ATOM
C 46-50  WDS     A5         CHARACTERS TO IDENTIFY THE ATOM
C
C  1-5   NTRAN   I5         SAME AS NTRAN ON CARD 2
C NTRAN GT 0. THE BOND  ORDER  WILL BE OPTIMIZED FOR ALL BREAKING BONDS
C NTRAN LT 0. THE BOND DISTANCE WILL BE OPTIMIZED FOR ALL BREAKING BONDS
C 5-10   NPOINT  I5       NUMBER OF POINTS REQUESTED ON REACTION PATH
C
C 1-20 P(1),P(NTRAN+1) 2F10.5  PATH COORDINATES OF REACTANT AND PRODUCT
C
C NPOINT CARDS MUST BE SUPPLIED. ONE FOR EACH STRUCTURE REQUESTED
C 1-10   PATH    F10.5     PATH COORD OF STRUCTURE REQUESTED
C 21-25  MAX     I5       ABS(MAX) EQ 1  HOLD CONSTANT GEOMETRIC  PC
C                           ABS(MAX) EQ 2  HOLD CONSTANT ELECTRONIC PC
C                           MAX GT 0  LOOK FOR TRANSITION STATE
C                           MAX LT 0  OPTIMIZE FLANKING POINT
C
C SKIP TO DFP OPTIONS  AFTER SYMETRY AND CI OPTIONS
C
C ***** TRIAL MOLECULAR GEOMETRY ***** ONE CARD PER ATOM ***************
C
C     1-2   NAT(I)   I2   ATOMIC NUMBER OF ATOM (I)
C                         =99 IF THE ATOM IS A DUMMY (IE. TO ASSIST
C                           IN THE DEFINITION OF THE GEOMETRY)
C                         = 0 TO END INPUT OF GEOMETRY
C     11-20  A(1,I)  F10.5  INTERATOMIC SEPARATION (IN ANGSTROMS)
C                         BETWEEN ATOMS I AND NA(I).
C     23-24  LA      I2   =-1 IF A(1,I) IS THE REACTION COORDINATE
C                         = 0 IF A(1,I) IS NOT TO BE OPTIMIZED
C                         =+1 IF A(1,I) IS TO BE OPTIMIZED
C     31-40  A(2,I)  F10.5  THE ANGLE NB(I)-NA(I)-(I) IN DEGREES
C     43-44  LB      I2   =-1,0,+1 AS WITH LA
C     51-60  A(3,I)  F10.5  THE ANGLE BETWEEN THE VECTORS NC(I)-NB(I)
C                         NA(I)-(I) IN DEGREES  MEASURED CLOCKWISE
C                         IN THE DIRECTION B TO A
C     63-64  LC      I2   =-1,0,+1 AS WITH LA AND LB
C     71-72  NA(I)   I2   ATOM NUMBER (SEE ABOVE)
C     73-74  NB(I)   I2   ATOM NUMBER (SEE ABOVE)
C     75-76  NC(I)   I2   ATOM NUMBER (SEE ABOVE)
C
```

```
C       ***** NOTE ***** SIMPLIFIED INPUT FOR ATOMS 1 3
C                  ATOM 1  ONLY NAT(1)
C                  ATOM 2  ONLY NAT(2) A(1 2) LA
C                  ATOM 3  ONLY NAT(3) A(1 3) LA A(2 3) LB
C                  THE PROGRAM ALWAYS ASSUMES
C                  NA(2)=1  NA(3)=2  NB(3)=1
C
C                  END INPUT OF GEOMETRY WITH A BLANK CARD
C
C ***** SYMMETRY DATA ***** KSYM=1 ************************************
C
C       ***** PART 1 ***** SYMMETRY DATA TO BE IMPOSED BEFORE COM-
C                  PUTING DEPENDENT PARAMETER VALUES
C                  ONE CARD PER SET OF EQUAL BOND LENGTHS
C                  EQUAL ANGLES OR EQUAL TWIST ANGLES
C                  IF KDEP=1 BUT THERE ARE NO SYMMETRY
C                  CONDITIONS TO BE IMPOSED BEFORE COMPUTING
C                  THE PARAMETERS A BLANK CARD MUST BE
C                  INSERTED
C                  END INPUT WITH A BLANK CARD
C
C     1-2  ISYM(11,I) I2  ATOM NUMBER ON WHICH THE VARIABLE PARA-
C                  METER OF THE I-TH SET IS LOCATED
C      5   ISYM(12,I) I1  TYPE OF PARAMETER FOR THE I-TH SET
C                  =1. BOND-LENGTH
C                  =2. BOND-ANGLE
C                  =3. TWIST-ANGLE
C    11-12 ISYM(13,I) I2  THE NUMBER OF PARAMETERS IN THE I-TH SET -
C                  NOT INCLUDING THE VARIABLE PARAMETER
C    21-70 ISYM(J,I) (I2,  ATOM NUMBERS ON WHICH THE REMAINING PARA-
C         J=1,10   3X)  METERS OF THE SET ARE LOCATED
C
C ***** DEPENDENT PARAMETERS ***** KDEP=1 ****************************
C
C                  ONE CARD FOR EACH PARAMETER THAT IS
C                  DETERMINED BY CHANGES IN THE VARIABLE
C                  PARAMETER
C                  THE FUNCTIONS MUST BE SUPPLIED BY THE USER
C                  IF THE ANGULAR FUNCTIONS SUPPLIED IN
C                  SUBROUTINE HADDON ARE NOT SUFFICIENT
C                  END INPUT WITH A BLANK CARD
C
C     1-2  LOCDEP(I) I2  ATOM NUMBER ON WHICH THE I-TH DEPENDENT
C                  PARAMETER IS LOCATED
C      4   N       I1  TYPE OF THE I-TH DEPENDENT-PARAMETER
C                  =1. BOND-LENGTH
C                  =2. BOND-ANGLE
C                  =3. TWIST-ANGLE
```

```
C     10-11  IDEPFN(I)  I2   THE FUNCTION NUMBER USED TO EVALUATE THE
C                          I-TH DEPENDENT PARAMETER
C                          THIS IS USED IN A  GOTO(      ) IDEPFN
C                          STATEMENT IN SUBROUTINE HADDON OR THE USER
C                          SUPPLIED SUBROUTINE DEPVAR  VALUES 1 TO 14
C                          ARE RESERVED FOR USE IN SUBROUTINE HADDON
C                          VALUES GREATER THAN 14 MUST BE USED IN
C                          SUBROUTINE DEPVAR
C     14-15  M         I2   THE NUMBER OF VARIABLES USED TO DETERMINE
C                          DEPENDENT PARAMETER (M=1 IN THIS PROGRAM)
C     21-22  LOCPAR(I)  I2   LOCATION OF THE PARAMETER USED TO DEFINE
C                          THE I-TH DEPENDENT PARAMETER
C     24     LTYPE      I1   TYPE OF PARAMETER AT LOCPAR(I)
C
C ***** SYMMETRY DATA (CONTINUED) ***** KSYM=1  KDEP=1 ****************
C
C       ***** PART 2 ***** SYMMETRY DATA TO BE IMPOSED AFTER COM-
C                          PUTING DEPENDENT PARAMETER VALUES.
C                          IF THERE ARE NO MORE SYMMETRY CONDITIONS
C                          AFTER COMPUTING THE DEPENDENT PARAMETERS.
C                          A BLANK CARD MUST BE INSERTED.
C                          INPUT OF SYMMETRY DATA IS THE SAME AS IN
C                          PART 1.
C                          END INPUT WITH A BLANK CARD
C
C ***** CONFIGURATION INTERACTION ***** KCI=1 *************************
C
C      1-5   LROOT     I5    CI STATE WHOSE GEOMETRY IS OPTIMIZED.
C                           DEFAULT 1.
C      6-10  K         I5    K,L,M,N ARE THE MOS INVOLVED IN CI
C     11-15  L         I5    GROUND-STATE 2*2 CI. DOUBLE EXCITATION
C     16-20  M         I5    K TO L. DEFAULT HOMO TO LUMO
C     21-25  N         I5    EXCITED SINGLET 3*3 CI. K AND L SINGLY
C                           OCCUPIED. SINGLE EXCITATIONS K TO M,L TO N
C                           DEFAULT HOMO-1 TO HOMO. HOMO TO HOMO-1.
C
C ***** OPTIMIZATION OPTIONS ******************************************
C
C 612    OPTIONS FOR OPTIMIZATION    SEE SUBR DFPMIN
C
C ***** INPUT FOR THE NEXT MOLECULE ***********************************
C
C    AT THIS POINT, THE NEXT MOLECULE CAN BE READ IN STARTING WITH THE
C    TITLE CARD.  TO TERMINATE THE JOB, SET KHARGE=99 (COLUMNS 1-2 OF
C    THE TITLE CARD).
C
C    SIMPLIFIED INPUT FOR REACTION PATHS. AFTER THE FIRST POINT, SET
C    KGEOM=1 (OR 2) IN THE TITLE CARD OF THE NEXT POINTS, AND THEN READ
```

```
C    ** ONLY ** THE FOLLOWING CARD
C
C ***** A NEW REACTION COORDINATE VALUE ONLY ***** KGEOM GE 1 **********
C
C     1 2  LREACT(1) I2   ATOM NUMBER WITH REACTION COORDINATE
C     4 5  LREACT(2) I2   TYPE OF THE REACTION COORDINATE
C                   =1. BOND-LENGTH
C                   =2. BOND-ANGLE
C                   =3. TWIST-ANGLE
C    11-20  REACT   F10 5  THE NEW VALUE FOR THE REACTION COORDINATE
C    21-22  NPOINT   I2   IDENTIFICATION NUMBER ONE MAY USE TO
C                   DISTINGUISH BETWEEN DIFFERENT POINTS IN A
C                   REACTION PATH
C
C ***********************
C
C    FILE DICTIONARY
C
C    1   DISK FILE. USED TO STORE THE H-MATRIX IN THE DFP SECTION.
C        CONTAINS EIGENVECTORS AND EIGENVALUES AT JOB END. IF IPUEV=1
C    2   DISK FILE. USED TO STORE THE REPULSION INTEGRALS IN SCF PART
C        CONTAINS TRIAL BOND ORDER MATRIX AT JOB START. IF KTRIAL=3
C        CONTAINS FINAL BOND ORDER MATRIX AT JOB END. IF IPUBO=1
C    4   DISK FILE  USED TO STORE DFP INFORMATION AFTER EACH CYCLE
C        NECESSARY TO CONTINUE A DFP OPTIMIZATION IN A NEW JOB
C    5   INPUT  FILE  CARD READER
C    6   OUTPUT FILE  PRINTER
C    7   PUNCH  FILE. THE OPTIMIZED GEOMETRY IS PUNCHED. IF IREP=1
C
C ***********************
C
C
C    OVERLAY(MAIN,0,0)
C    PROGRAM MNDO (INPUT=512,OUTPUT=1001,PUNCH=101,TAPE1=512,
C    TAPE2=512,TAPE4=512,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=PUNCH)
     IMPLICIT REAL*8 (A-H,O-Z)
     PARAMETER (NUMATM=16)
     COMMON/UNITS/IN,IO,JO,LINES,LITOT
     /CNTROL/ ICNTRL,JCNTRL(5)
     /CYCLES/ NPBI,NPETER
C KOUTPT   ARE NEVER USED   NTO,NREM LOCAL IN PRTDFP
     /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
          KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
          ITIME,NTYPE,KRESET,IWADE,NTO,NREM,MAXEND
     /OVERLY/ IOV,JOV,KOV,LOV
     /RC    / IREACD
     /XPRJU / IREP,IJUMP
     /MMSYNC/ NDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
```

```
      /SAVCOM/ TIMEO LINCNT
C
C     OVERLAY CONTROL
      IOV=0
      JOV=0
      KOV=0
      LOV=0
C     SET UNDERFLOW EQUAL TO ZERO (IBM)
CALL UFLOWF( TRUE )
      CALL ERRSET(208 300 -1)
C     SET UPPER LIMIT FOR TIMING (IBM)
C     CALL STIME(1000000)
      TIME1=0 D0
      TIMEO=0 D0
      NPBI=O
      NPETER=0
      CALL CLOCK1
      ICNTRL=0
C  DEFINE UNIT NUMBERS
      IN=5
      IO=6
      JO=7
C     READ THE INPUT DATA
   10 CALL START
C IF A TRANSITION STATE CALCULATION.
C GENERATE THE STRUCTURE USING SYNCHRONOUS TRANSIT PACKAGE
      IF(MIDDLE EQ 0 AND NTRAN NE 0)CALL SYNTRN
      IF(MAXEND GT 2)GOTO 30
C     DO A SINGLE SCF-CALCULATION
   20 IWADE=0
      CALL SCF
      NPETER=1
      GOTO 40
C     OPTIMIZE THE MOLECULAR GEOMETRY
   30 CALL OPT
      IF(JOV EQ.0) GOTO 40
      CALL SCF
      GOTO 30
C     PRINT THE RESULTS.
   40 CALL FINAL
      IF(IJUMP EQ 1) STOP
C     DETERMINE THE COMPUTATION TIME
      TIME2=6 D1*CLOCK(I)
      TIME1=TIME2-TIME1
      IF(MIDDLE.EQ.1) TIME1=TIME1+SECADD
      IF(LINES+5.GT.LITOT) CALL PAGE
      LINES=LINES+5
      WRITE(IO,50) TIME1
```

```
      WRITE(IO,60) NPETER
C     PREPARE FOR THE NEXT MOLECULE
      TIME1=TIME2
      SECADD=0.0D0
      MIDDLE=0
      IF(IREACD.EQ.1)MAXEND=9999
      ICNTRL=1
      GOTO 10
   50 FORMAT(//5X,'COMPUTATION TIME =',F8.2,3X,'SECONDS')
   60 FORMAT(/,5X,'SCF CALCULATIONS =',I5)
      END
C
C ***********************
C
      SUBROUTINE COMPFG(NVAR,X,ENERGY,G,GZ,GMAX)
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR.DIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
C     *
C     THIS ROUTINE CALLS THE SCF PART
C     *
      COMMON
     /ERG   / EX,GG(I3NX)
     /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
              KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
              ITIME,NTYPE,KRESET,IWADE,NTO,NREM,MAXEND
     /OVERLY/ IOV,JOV,KOV,LOV
     /PARM1 / A(3,NUMATM),NATMD,NI(NUMATM),NA(NUMATM),NB(NUMATM),
              NC(NUMATM),NN(NUMATM),LOC(2,I3NX),LREACT(2)
     /MMSYNC/ NDHA,NEWDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
     /RHO   / P(MPAK),B(MPAK-10),AA(5),R0(5)
     /SYNCOM/ D1(I3NX,3),W(I3NX),VD(I3NX),S(11),NVC(I3NX),JD(14)
     /ATOMS / C(3,NUMATM),NATOMS,IDUM(I3NX)
     /INTVAR/ VF(I3NX),VL(I3NX),Z(I3NX),BF(MPAK),BL(MPAK),DFL,
              LOCVI(I3NX),MAX2
      DIMENSION G(1),X(1)
C
      IF(IOV.EQ.1 .AND. JOV.EQ.1) GOTO 60
CHANGE THE GEOMETRY
      NBDORD=0
      DO 10 I=1,NVAR
        XI=X(I)
        IF(NTRAN.EQ.0 .OR. NVC(LOCVI(I)).GT.0) GOTO 10
          NBDORD=NBDORD+1
          XI=DMAX1(1.D-6,XI)
          XI=R0(NBDORD)+AA(NBDORD)*DLOG10(XI)
   10   A(LOC(2,I),LOC(1,I))=XI
```

```
C INTERPOLATE XYZ COORDS FROM INTERNAL VARIABLES
      CALL SYMTRY( 1)
      CALL GMETRY(C  1)
C
CALL THE SCF PART
C
      IF(IOV EQ 0) GOTO 50
      JOV=1
      RETURN
   50 CALL SCF
   60 JOV=0
C STORE ENERGY AND GRADIENTS
      ENERGY=EX
      GMAX=0 D0
      IF(IWADE EQ 0) RETURN
      DO 70 I=1 NVAR
        G(I)=GG(I)
        GMAX=DMAX1(GMAX DABS(G(I)))
   70   IF(DABS(G(I)) LT 1 D-5) G(I)=1 D-5
C
      IF(NTRAN EQ 0) RETURN
C
COMPUTE DOT PRODUCT OF G AND Z VECTORS
C
      GZ=0 D0
      DO 80 I=1 NVAR
   80   GZ=GZ+G(I)*Z(I)*W(LOCVI(I))
      IF(IABS(MAX2) NE 1) RETURN
      RETURN
      END
C
C ***********************
C     OVERLAY(SCF 3 0)
C     PROGRAM SCF
C
      SUBROUTINE SCF
      IMPLICIT REAL*8 (A-H O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16)
      PARAMETER (I3NX=3*NUMATM)
C     *
C     SCF SECTION OF THE PROGRAM
C     *
      COMMON
      /UNITS / IN IO JO LINES LITOT
      /ERG   / ENERGY G(I3NX)
      /FLAGS / SCFCRT COMENT(5) TITLE(10) SECADD TIME1 KITSCF
            KHARGE KOUTPT IDUMMY KGEOM KOUNT KSYM KDEP MIDDLE
```

```
      ITIME,NTYPE,KRESET,IWADE,NTO,NREM,MAXEND
     /CYCLES/ JCYC,NPETER
     /PASS3 / COS,PNORM,JPRINT,JNRST,NCNT,I116,LNSTOP,IREPET
     /SAVCOM/ TIMEO,LINCNT
     /BESAFE/ TLIMIT,SFDFP,SFSCF,LLIMIT
     /XPRJU / IREP,IJUMP
     /CI    / KTHMO,LTHMO,MTHMO,NTHMO,LROOT,LPRINT,NC,CIYES,ELENGY,
      ENGYCI(25),VECTCI(25,25),JFROM(25,3),JTO(25,3)
     /HALFE / ODUM(5),IMULT,IDUM(7)
     /UHFCRS/ IUHF,ICROSS
     LOGICAL CIYES,IUHF,ANALY
C
C SEE IF TIME LIMIT OR LINE LIMIT HAS BEEN APPROACHED
C
     TIME=6.D1*CLOCK(I)
     DT=TIME-TIMEO
     L=LINCNT+LINES+LITOT
     IF(TIME-DT.LT.TLIMIT.AND.L.LT.LLIMIT) GOTO 130
     IF(LINES+7+NUMAT.GT.LITOT) CALL PAGE
     IF(TIME-DT.GE.TLIMIT) WRITE(IO,10) TIME,TLIMIT
  10 FORMAT(/' **** THE TIME LIMIT HAS BEEN APPROACHED ',F9.2,
    1' SECONDS HAVE BEEN USED.',F9.2,' SECONDS WERE REQUESTED ****')
     IF(L.GE.LLIMIT) WRITE(IO,20) L,LLIMIT
  20 FORMAT(/' **** THE LINE LIMIT HAS BEEN APPROACHED ',I9,
    1' LINES HAVE BEEN PRINTED.',I9,' LINES WERE REQUESTED ****')
     IF(JCYC.EQ.0) WRITE(IO,25)
  25 FORMAT(/5X,'THE FIRST LINE SEARCH HAS NOT BEEN COMPLETED.',
    1'SO THE HESSIAN HAS YET BEEN INITIALIZED'/15X,'RESUBMIT',
    2'THE SAME RUN WITH A LARGER VALUE OF TLIMIT ON CARD 1.')
     IF(JCYC.EQ.0) STOP
     IJUMP=JCYC.EQ.NSTOP
     CALL PUNOUT
     STOP
C
  130 TIMEO=TIME
C     TIME1=6.D1*CLOCK(I)
     NCNT=NCNT+1
     CALL FMTRX
     IF(IWADE.NE.0) CALL DRTP(G)
C     TIME2=6.D1*CLOCK(I)
C     CALL DXYZ
C     CALL DRTP(G)
C     TIME3=6.D1*CLOCK(I)-TIME2
C     TIME2=TIME2-TIME1
C     WRITE(6,151) TIME2,TIME3
C151   FORMAT(' SUBR SCF ',F7.3,' SECS TO COMPUTE THE ENERGY ',
C     1 F7.3,' SECS FOR THE GRADIENT ')
     RETURN
```

```
      END
C
C  **********************
C
      SUBROUTINE DRTP(G)
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
C
CONVERT DE/DXYZ TO DE/DRTP
C
      COMMON
     /UNITS / IN,IO,JO,LINES,LITOT
     /DFP   / SET(I3NX),NVAR
     /FLAGS / SCFCRT(14),KITSCF(11),KRESET,IWADE(4)
     /SYNCOM/ COMSYN(I3NX*5),SS(11),NVC(I3NX),JJD(5),NTRAN,JKD(8)
     /PARM1 / A(3,NUMATM),NATMD,NI(NUMATM,5),LOC(2,I3NX),LREACT(2)
     /SKIPA / NWSKP
     /VVV   / CHANGE(3),DEL(3),XFAC,YFAC
     /ATOMS / C(3,NUMATM),NATOMS,JDUM(I3NX)
     /FORCE1/ FORC1(20),FDER(3,NUMATM),SDER(3,NUMATM),NPTS
     /INTVAR/ VFL(I3NX*2),Z(I3NX),BFL(MPAK*2),DFL,LOCVI(I3NX),MAX
     /RHO   / PP(MPAK),BORD(MPAK-1),AA(5),R0(5)
     /FORCE2/ DF(I3NX,9),BINV(I3NX*I3NX)
     DIMENSION GG(I3NX),G(I3NX),COP(3,NUMATM)
          ,B(12,I3NX),IB(4,I3NX),LB(I3NX)
C
      KRESET=1
      NR=NI(1,1)
      B(1,NR)=1.D0
      DO 33 I=2,12
   33   B(I,NR)=0.D0
      II=NI(2,1)
      IB(1,NR)=II
      IB(2,NR)=NI(II,2)
      IB(3,NR)=0
      IB(4,NR)=0
      DO 35 I=1,NVAR
   35   IF(LOC(1,I).EQ.II .AND. LOC(2,I).EQ.1) LB(I)=NR
      NT=NATMD-1
      NP=NT+NT-1
      IF(NATMD.LE.2) GOTO 60
      DO 50 K=3,NATMD
      II=NI(K,1)
      JJ=NI(II,2)
C BOND LENGTH
      VJI1=C(1,II)-C(1,JJ)
```

```
      VJI2=C(2,II)-C(2,JJ)
      VJI3=C(3,II)-C(3,JJ)
      DJI=DSQRT(VJI1**2+VJI2**2+VJI3**2)
      VJI1=VJI1/DJI
      VJI2=VJI2/DJI
      VJI3=VJI3/DJI
      NR=NR+1
      DO 38 I=1,NVAR
   38   IF(LOC(1,I).EQ.II .AND. LOC(2,I).EQ.1) LB(I)=NR
      IB(1,NR)=II
      IB(2,NR)=JJ
      IB(3,NR)=0
      IB(4,NR)=0
      B(1,NR)=VJI1
      B(2,NR)=VJI2
      B(3,NR)=VJI3
      B(4,NR)=-VJI1
      B(5,NR)=-VJI2
      B(6,NR)=-VJI3
      DO 40 J=7,12
   40   B(J,NR)=0.D0
C
C BOND ANGLE
      KK=NI(II,3)
      NT=NT+1
      IB(1,NT)=II
      IB(2,NT)=JJ
      IB(3,NT)=KK
      IB(4,NT)=0
      DO 45 I=1,NVAR
   45   IF(LOC(1,I).EQ.II .AND. LOC(2,I).EQ.2) LB(I)=NT
      VJK1=C(1,KK)-C(1,JJ)
      VJK2=C(2,KK)-C(2,JJ)
      VJK3=C(3,KK)-C(3,JJ)
      DJK=DSQRT(VJK1**2+VJK2**2+VJK3**2)
      VJK1=VJK1/DJK
      VJK2=VJK2/DJK
      VJK3=VJK3/DJK
      DOTIJK=VJK1*VJI1+VJK2*VJI2+VJK3*VJI3
      SINJ=DSQRT(1.D0-DOTIJK**2)
      DK=DJK*SINJ
      DI=DJI*SINJ
      B(1,NT)=(DOTIJK*VJI1-VJK1)/DI
      B(2,NT)=(DOTIJK*VJI2-VJK2)/DI
      B(3,NT)=(DOTIJK*VJI3-VJK3)/DI
      B(7,NT)=(DOTIJK*VJK1-VJI1)/DK
      B(8,NT)=(DOTIJK*VJK2-VJI2)/DK
      B(9,NT)=(DOTIJK*VJK3-VJI3)/DK
```

```
      B(4,NT)=-(B(1,NT)+B(7,NT))
      B(5,NT)=-(B(2,NT)+B(8,NT))
      B(6,NT)=-(B(3,NT)+B(9,NT))
      B(10,NT)=0.D0
      B(11,NT)=0.D0
      B(12,NT)=0.D0
      IF(NATMD.LT.4) GOTO 60
      IF(K.LT.4) GOTO 50
C
C DIHEDRAL ANGLE
      LL=NI(II,4)
      NP=NP+1
      IB(1,NP)=II
      IB(2,NP)=JJ
      IB(3,NP)=KK
      IB(4,NP)=LL
      DO 48 I=1,NVAR
   48   IF(LOC(1,I).EQ.II .AND. LOC(2,I).EQ.3) LB(I)=NP
      F1=DJI*DOTIJK**2-DJI
      B(1,NP)=(VJK2*VJI3-VJK3*VJI2)/F1
      B(2,NP)=(VJK3*VJI1-VJK1*VJI3)/F1
      B(3,NP)=(VJK1*VJI2-VJK2*VJI1)/F1
      VKL1=C(1,LL)-C(1,KK)
      VKL2=C(2,LL)-C(2,KK)
      VKL3=C(3,LL)-C(3,KK)
      DOTJKL=VKL1*VJK1+VKL2*VJK2+VKL3*VJK3
      F4=VKL1**2+VKL2**2+VKL3**2-DOTJKL**2
      B(10,NP)=(VJK2*VKL3-VJK3*VKL2)/F4
      B(11,NP)=(VJK3*VKL1-VJK1*VKL3)/F4
      B(12,NP)=(VJK1*VKL2-VJK2*VKL1)/F4
      F1=DOTIJK*DJI/DJK-1.D0
      F4=DOTJKL/DJK
      B(4,NP)=B(1,NP)*F1+B(10,NP)*F4
      B(5,NP)=B(2,NP)*F1+B(11,NP)*F4
      B(6,NP)=B(3,NP)*F1+B(12,NP)*F4
      B(7,NP)=-(B(1,NP)+B(4,NP)+B(10,NP))
      B(8,NP)=-(B(2,NP)+B(5,NP)+B(11,NP))
      B(9,NP)=-(B(3,NP)+B(6,NP)+B(12,NP))
   50 CONTINUE
C
C APPLY MASK TO B  ZERO OUT XYZ FOR ATOM 1  YZ FOR ATOM 2  Z FOR ATOM 3
C
   60 DO 78 I=1,NP
         DO 76 J=1,4
            IBJ=IB(J,I)
            IF(IBJ.EQ.0) GOTO 78
            JJ=3*J-2
            IF(IBJ.EQ.NI(3,1)) GOTO 74
```

```
          IF(IBJ EQ NI(2 1)) GOTO 72
           IF(IBJ NE NI(1 1)) GOTO 76
             B(JJ I)=0 D0
  72        B(JJ+1 I)=0 D0
  74       B(JJ+2 I)=0 D0
  76      CONTINUE
  78    CONTINUE
C
COMPUTE B TRANSPOSE * B
C
  80 NPI=0
     DO 90 I=1 NP
     DO 90 J=1 I
       NPI=NPI+1
       R=0 D0
       DO 86 II=1 4
         IBI=IB(II I)
         IF(IBI EQ 0) GOTO 90
          DO 84 JJ=1 4
            IF(IB(JJ J) NE IBI) GOTO 84
              IP=3*II-3
              JP=3*JJ-3
              DO 82 K=1 3
  82             R=R+B(IP+K I)*B(JP+K J)
              GOTO 86
  84          CONTINUE
  86        CONTINUE
  90    BINV(NPI)=R
     EPS=1 D-10
     CALL DSINV(BINV NP EPS IER)
C
COMPUTE  BINV * B (TRANSPOSE) * FDER
C
     DO 105 J=1 NVAR
 105   G(J)=0 D0
     II=0
     DO 170 I=1 NP
       II=II+I-1
       R=0 D0
       L3=0
       DO 150 L=1 4
         IBI=IB(L I)
         IF(IBI EQ 0) GOTO 160
          DO 140 K=1 3
 140          R=R+B(L3+K I)*FDER(K IBI)
 150        L3=L3+3
 160 DO 170 J=1 NVAR
       JJ=LB(J)*(LB(J)-1)/2
```

```
      IJ=MAX0(II+LB(J) JJ+I)
  170   G(J)=G(J)+BINV(IJ)*R
C
CONVERT DE/DR TO DE/D(BOND ORDER)
C
    IF(NTRAN.NE.0) THEN
      NBDORD=0
      DO 210 I=1,NVAR
        IF(NVC(LOCVI(I)).GT.0) GOTO 210
          NBDORD=NBDORD+1
          G(I)=G(I)*AA(NBDORD)/(2.302585093D0*SET(I))
  210     CONTINUE
    ENDIF
    KRESET=0
    RETURN
    END
C
C **********************
C
    SUBROUTINE INPUT
    IMPLICIT REAL*8 (A-H,O-Z)
C    INCLUDE 'NDFOR.DIM/NOLIST'
    PARAMETER (NUMATM=16,NUMORB=34)
    PARAMETER (MPAK=(NUMORB*(NUMORB+1))/2,NDIE=6*NUMORB*NUMATM)
    COMMON
    /VVV   / CHANGE(3),DELTA(3),XFAC,YFAC
    /ATDATA/ F0(18),USS(18),UPP(18),VS(18),VP(18),CORE(18),
          EISOL(18),EHEAT(18)
    /BOND  / KTRIAL
    /SETCI / NR,KCI
    /VIRTUL/ IVIR,IBIRA
    /RHO   / P(MPAK),PBETA(MPAK)
    /BNDATA/ BDUM(110),ZS(18),ZP(18)
    /FORCE6/ AMS(17),INAT(NUMATM)
    /ATOMS / COORD(3,NUMATM),NUMAT,NAT(NUMATM),
          NFIRST(NUMATM),NLAST(NUMATM)
    /FORCE5/ MAXF,IPRNT,IPNT
    /UORB  / NALPHA,NBETA
    /CI    / KTHMO,LTHMO,MTHMO,NTHMO,LROOT,LPRINT,NCI,CIYES,ELENGY,
          ENGYCI(25),VECTCI(25,25),JFROM(25,3),JTO(25,3)
    /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
          KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
          ITIME,NTYPE,KRESET,IWADE,NTO,NREM,MAXEND
    COMMON
    /HALFE / IMULT,NOCCO,NCLOSE,IOPEN,IODD,JODD,XIIII,XJJJJ,XIJIJ
    /CNTROL/ ICNTRL,KCNTRL,LCNTRL,MCNTRL,NCNTRL
    /SCFPRT/ SCFPRT
    /UHFCRS/ IUHF
```

```
      /ORBITS/ NUMB NORBS C(NUMORB NUMORB) H(MPAK) F(MPAK)
           E1ELN(NUMORB) Q(NUMORB 2)
      /PASS3 / COS PNORM JPRINT JNRST NCNT I116 LNSTOP IREPET
      /GAUSS / CC(NUMORB 6) ZZ(NUMORB 6) SG(4 4)
      /GAUSSI/ CSH(6) ZSH(6) CFIR(6 2) ZFIR(6 2) CSEC(6 2) ZSEC(6 2)
      /MMSYNC/ NDHA NEWDHA NTRAN NTRN1 IJKL(2 NUMATM)
      /XSKPRT/ IPUBO IPUEV NPOINT NPRINT
      LOGICAL QUADRS IUHF CIYES ISOTOP SCFPRT
C
      IBIRA=0
   60 IVIR=0
      IJUMP=0
      KRESET=0
      NPBI=0
      NPOINT=0
      QUADRS= FALSE
C
      IF(MIDDLE EQ 0) GOTO 3
      CALL PUNIN
      MAXEND=MAXEND+NCNT
      GOTO 7

C *** TITLE CARD FOR THE MOLECULE

    3 READ(5 195)KHARGE IMULT KTRIAL KGEOM IPUBO IPUEV KITSCF NPRINT
     1 KDUMMY KSYM KDEP KCI NSTART NSTEP NTRAN (TITLE(I) I=1 6)
    7 WRITE(7 195)KHARGE IMULT KTRIAL KGEOM IPUBO IPUEV KITSCF NPRI
     1NT KDUMMY KSYM KDEP KCI NSTART NSTEP NTRAN (TITLE(I) I=1 6)
      IF(KHARGE GE 99) STOP
      IF(IMULT GT 3) WRITE(6 205)
      IF(IMULT GT 3) STOP
      IF(NPRINT GT 1 AND MAXEND GT 3) NPRINT=1
      IF(NSTART EQ 0) NSTART=4
      IF(NSTEP EQ 0) NSTEP=4
      IF(KCI NE 0 OR IMULT NE 0) QUADRS= TRUE
      IF( NOT QUADRS) NWSKP=0
      ZFAC=XFAC
      IF(QUADRS) ZFAC=YFAC
      DO 10 I=1 3
   10 DELTA(I)=CHANGE(I)*ZFAC
   10 DELTA(I)=CH
      IF (IUHF AND KITSCF EQ 0) KITSCF=250
      IF (KITSCF EQ 0) KITSCF=99
      SCFPRT=KITSCF LT -1
      KITSCF=IABS(KITSCF)
      CIYES=KCI NE 0
      IF (KCI LT 0 AND IMULT EQ 0) IMULT=1
C
C     ******     READ IN VALUES FOR CI     ******
```

```
C
      IF ( NOT CIYES) GOTO 150
      READ(5,600) NR,LROOT,LPRINT
      IF (IMULT NE 0) GOTO 70
      NX=3
      J=2
   70 IF (IMULT EQ 1) J=4
      IF (IMULT EQ 3) J=2
      IF (IMULT GT 0) NX=NR+J-1
      IF (NR GT 0) READ(5,610) ((JFROM(M,N),JTO(M,N) N=1,3),M=J,NX)
      NCI=NX
      IF (IMULT EQ 0 OR NR EQ 0) GOTO 140
C
C     *****    ORDER JFROM AND JTO AND CHECK FOR ERRORS     *****
      DO 130 M=J,NX
        DO 80 I=1,3
      IF (JFROM(M,1) EQ 0 OR JTO(M,1) EQ 0) GOTO 81
          IF (JFROM(M,I) NE 0 AND JTO(M,I) NE 0) GOTO 80
          IF (JFROM(M,I) EQ JTO(M,I)) GOTO 80
   81 WRITE(6,620) M
          CALL EXIT
   80   CONTINUE
        DO 100 I=1,2
          K=I
          JF=JFROM(M,I)
          JT=JTO(M,I)
          IP1=I+1
          DO 90 L=IP1,3
            IF (JFROM(M,L) GE JF) GOTO 90
            K=L
            JF=JFROM(M,L)
            JT=JTO(M,L)
   90     CONTINUE
          IF (K EQ I) GOTO 100
          JFROM(M,K)=JFROM(M,I)
          JTO(M,K)=JTO(M,I)
          JFROM(M,I)=JF
          JTO(M,I)=JT
  100   CONTINUE
  110   IF (JFROM(M,1) NE 0) GOTO 130
        DO 120 I=1,2
          IP1=I+1
          JFROM(M,I)=JFROM(M,IP1)
  120   JTO(M,I)=JTO(M,IP1)
        JFROM(M,3)=0
        JTO(M,3)=0
        GOTO 110
  130 CONTINUE
```

```
 140 CONTINUE
     IF (IMULT.GT.0) GOTO 150
     KTHMO=JFROM(2,1)
     LTHMO=JFROM(3,1)
     MTHMO=JTO(2,1)
     NTHMO=JTO(3,1)
 150 CONTINUE
C
C *** INPUT (OR COMPUTE) THE TRIAL GEOMETRY AND COMPUTE THE COORDINATES
C
     KRESET=0
     IF(NTRAN.EQ.0) THEN
       KG=IABS(KGEOM)
       CALL ESTIM
       IF(KG.LE.3.OR.MAXEND.NE.1) CALL GMETRY(COORD,+1)
     ELSE
       CALL INPXYZ
     ENDIF
     IF (CIYES.AND.LPRINT.LE.0) ISKPBO=1
     ISOTOP=.FALSE.
     DO 41 I=1,NUMAT
     INAT(I)=NAT(I)
     IF(NAT(I).EQ.2) ISOTOP=.TRUE.
     IF(NAT(I).EQ.2) NAT(I)=1
     IF(NAT(I).EQ.3) ISOTOP=.TRUE.
     IF(NAT(I).EQ.3) NAT(I)=6
     IF(NAT(I).EQ.10) ISOTOP = .TRUE.
     IF(NAT(I).EQ.10) NAT(I)=6
     IF(NAT(I).EQ.11) ISOTOP = .TRUE.
     IF(NAT(I).EQ.11) NAT(I)=8
     IF(NAT(I).EQ.12) ISOTOP = .TRUE.
  41 IF(NAT(I).EQ.12) NAT(I)=5
C
C***  OUTPUT TRIAL COORDINATES. COMPUTE NUMBER OF OCCUPIED MOS. ETC
C
     IF(NTRAN.EQ.0) THEN
       WRITE(6,630) (TITLE(I),I=1,6),(COMENT(I),I=1,5)
       IF (IREP.NE.0) WRITE(7,632) (TITLE(I),I=1,6),(COMENT(I),I=1,5)
       WRITE(6,800)
 632   FORMAT (6A8/5A8)
     ENDIF
     NELNS=-KHARGE
     IA=1
     DO 190 I=1,NUMAT
       NFIRST(I)=IA
       NI=NAT(I)
       IF (NI.LE.2) THEN
         NELNS=NELNS+NAT(I)
```

```
        IB=IA
       ELSEIF (NI.LE.10) THEN
        NELNS=NELNS+NAT(I)*2
        IB=IA+3
       ELSE
        NELNS=NELNS+NAT(I)*10
        IB=IA+3
       ENDIF
       NLAST(I)=IB
     IF(NTRAN.NE.0) GOTO 190
       WRITE(6,810) I,NAT(I),(COORD(J,I),J=1,3),IA,IB
     IF (IREP.NE.0) WRITE(7,634) NAT(I),(COORD(J,I),J=1,3)
 634 FORMAT (I5,3F10.6)
 190 IA=IB+1
C    SET-UP THE STEWART'S STO-6G EXPANSIONS
     DO 32 I=1,NUMAT
     NI=NAT(I)
     IA=NFIRST(I)
     IB=NLAST(I)
     XI=ZS(NI)
     IF(NI.GT.2) GOTO 27
     DO 26 J=1,6
     CC(IA,J)=CSH(J)
  26  ZZ(IA,J)=ZSH(J)*XI**2
     GOTO 32
  27 IF(NI.GT.10) GOTO 29
     DO 33 K=IA,IB
     L=1
     IF(K.GT.IA) L=2
     IF(K.GT.IA) XI=ZP(NI)
     DO 28 J=1,6
     CC(K,J)=CFIR(J,L)
  28  ZZ(K,J)=ZFIR(J,L)*XI**2
  33 CONTINUE
     GOTO 32
  29 DO 34 K=IA,IB
     L=1
     IF(K.GT.IA) L=2
     IF(K.GT.IA) XI=ZP(NI)
     DO 34 J=1,6
     CC(K,J)=CSEC(J,L)
     ZZ(K,J)=ZSEC(J,L)*XI**2
  34 CONTINUE
  32 CONTINUE
     IF(.NOT.ISOTOP) GOTO 51
     WRITE(6,618)
 618 FORMAT(///,5X,'ISOTOPIC SUBSTITUTIONS PRESENT',/)
     DO 42 I=1,NUMAT
```

```
      IF(INAT(I).EQ. 2)WRITE(6,619)I
      IF(INAT(I).EQ. 3)WRITE(6,624)I
      IF(INAT(I).EQ.10)WRITE(6,621)I
      IF(INAT(I).EQ.11)WRITE(6,622)I
   42 IF(INAT(I).EQ.12)WRITE(6,623)I
  619 FORMAT(6X,'ATOM ',I3,' = DEUTERIUM')
  624 FORMAT(6X,'ATOM ',I3,' =CARBON 13')
  621 FORMAT(6X,'ATOM ',I3,' = CARBON 14')
  622 FORMAT(6X,'ATOM ',I3,' = OXYGEN 18')
  623 FORMAT(6X,'ATOM ',I3,' = BORON 10')
   51 CONTINUE
      IF(NTRAN.EQ.0) CALL AXIS
      IF(MAXF.LT.0) GOTO 60
      NORBS=NLAST(NUMAT)
      MOS=NELNS/2
      KLM=NELNS-2*MOS
      NOCCO=MOS+KLM
      NCLOSE=MOS
      IODD=0
      JODD=0
      IF (IMULT.EQ.0.AND.KLM.EQ.1) IMULT=2
      IF (IMULT.EQ.0.OR.IMULT.EQ.2) GOTO 200
      NOCCO=MOS+1
      NCLOSE=MOS-1
  200 IOPEN=NOCCO-NCLOSE
      IF (IMULT.EQ.0.AND.IUHF) IBIRA=1
C
C **** CHECK FOR INPUT ERRORS
C
      ILIGHT=0
      DO 210 I=1,NUMAT
  210   IF (NAT(I).LE.2) ILIGHT=ILIGHT+1
      IHEAVY=NUMAT-ILIGHT
      NOREP=50*IHEAVY*(IHEAVY-1)+ILIGHT*(ILIGHT-1)/2+10*ILIGHT*IHEAVY
      IF (NOREP.LE.NDIE) GOTO 220
      WRITE(6,640) NOREP
      WRITE(6,680)
      CALL EXIT
  220 IF (NORBS.LE.NUMORB) GOTO 230
      WRITE(6,650) NORBS
      WRITE(6,680)
      CALL EXIT
  230 IF (NOCCO.LE.NUMORB) GOTO 240
      WRITE(6,660) NOCCO
      WRITE(6,680)
      CALL EXIT
  240 IF (IMULT.LE.3) GOTO 250
      WRITE(6,670)
```

```
      WRITE(6,680)
      CALL EXIT
  250 IF (IMULT.EQ.0) GOTO 270
      IF (IMULT.EQ.2) GOTO 260
      IODD=NCLOSE+1
      JODD=NOCCO
      GOTO 270
  260 IODD=NOCCO
  270 IF (IUHF) GOTO 290
      WRITE(6,820) KHARGE,NELNS,NCLOSE
      IF (IOPEN.EQ.1) WRITE(6,690) IMULT,IODD
      IF (IOPEN.EQ.2) WRITE(6,700) IMULT,IODD,JODD
      IF (IMULT.EQ.0.OR.IMULT.EQ.2.OR.IUHF) GOTO 280
  280 CONTINUE
      IF (KTRIAL.EQ.2) WRITE(6,710)
      IF ( .NOT.CIYES.OR.KGEOM.NE.0) GOTO 310
C
C     ******   SET DEFAULT VALUES AND INITIAL VALUES FOR CI   ******
C
      CALL SETUP
C
      GOTO 310
  290 CONTINUE
      NALPHA=MOS
      NBETA=MOS
      IF (IMULT.EQ.0) GOTO 300
      NALPHA=MOS+1
      NBETA=MOS-1
      IF (IMULT.EQ.3) GOTO 300
      NALPHA=MOS+1
      NBETA=MOS
      IF (IMULT.NE.2) CALL EXIT
  300 CONTINUE
      WRITE(6,720) KHARGE,NELNS,IMULT,NALPHA,NBETA
      IF (IBIRA.EQ.1) WRITE(6,730)
  310 CONTINUE
C
COMPUTE OR INPUT A TRIAL DENSITY MATRIX
C
      LIN = MPAK
      IF (MIDDLE.EQ.1) RETURN
      IF (KTRIAL.EQ.3) GOTO 390
      IF (KTRIAL.GT.0) RETURN
C
COMPUTE THE TRIAL DENSITY MATRIX
C
      IF(IUHF) GOTO 370
      CALL GUESSP(P,KHARGE,NORBS)
```

```
      RETURN
C
C     SET UP TRIAL MATRIX FOR REGULAR UHF
C
  370 MCNTRL=1
      IF(IBIRA EQ 1) RETURN
      MCNTRL=0
      WSUM=0
      DO 381 I=1,NUMAT
        NI=NAT(I)
        XX=1 0D0
        IF(NI GT 2) XX=0 25D0
        W=CORE(NI)*XX-YY
        IA=NFIRST(I)
        IB=NLAST(I)
      DO 381 J=IA,IB
        DO 382 L=1,J
          K=K+1
          P(K)=0.0D0
  382   PBETA(K)=0 0D0
        WSUM=WSUM+W
  381 P(K)=W
      DIAGA=DFLOAT(NALPHA)/WSUM
      DIAGB=DFLOAT(NBETA)/WSUM
      K=0
      DO 383 I=1,NORBS
        K=K+I
        P(K)=P(K)*DIAGA
  383 PBETA(K)=P(K)*DIAGB
      RETURN
C
C     INPUT MATRIX
C
  390 READ(9,END=400)(P(IZZ),IZZ=1,LIN)
      IF(IUHF) READ(9,END=400)(PBETA(IZZ),IZZ=1,LIN)
      RETURN
  400 WRITE(6,740)
      CALL EXIT
C
  195 FORMAT(6I2,I4,8I2,6A8)
  205 FORMAT(/' MULTIPLICITY GREATER THAN 3'//)
  450 FORMAT (5I2,2E10 4,F10 6,F5 3,I2,I3,3F10 3)
  460 FORMAT (I5,3F10 5,6I5)
  470 FORMAT (I5,7F10 5)
  590 FORMAT (5I2,3X,I3,2X,4I2,4X,6A8,2X)
  600 FORMAT (3I5)
  610 FORMAT (12I5,20X)
  620 FORMAT (1H0,10X, 47HYOU MADE A SILLY INPUT MISTAKE IN CONFIGURATIO
```

```
     1N,I3, 23H  THERE MIGHT BE MORE )
  630 FORMAT (1H1// 10X,6A8 /10X,5A8)
  640 FORMAT(///' $$$$ NUMBER OF DIATOMIC ELECTRON REPULSION INTEGRALS'
     1 I5 ' EXCEEDS LIMIT')
  650 FORMAT(///' $$$$ NUMBER OF ATOMIC ORBITALS' I5 ' EXCEEDS LIMIT')
  660 FORMAT (///' $$$$ NUMBER OF OCCUPIED M.O. I5 ' EXCEEDS LIMIT')
  670 FORMAT (///' $$$$ MULTIPLICITY GREATER THAN 3 ')
  680 FORMAT (///' $$$$ CALCULATION ABANDONED')
  690 FORMAT (/10X,15H MULTIPLICITY = I4,10X,23HONE ELECTRON IN ORBITAL,
     1 I4/)
  700 FORMAT (/10X,15H MULTIPLICITY = I4,10X,24HONE ELECTRON IN ORBITALS
     1,I4,2X, 3HAND,I4/)
  710 FORMAT (//,10X, 37HPOPLE BOND ORDER MATRIX IS BEING USED.//)
  720 FORMAT (//,10X,7HCHARGE= I3,I8,2X,28HELECTRONS  MULTIPLICITY =
     1,I3,5X, 7HNALPHA= I3,5X, 6HNBETA= I3 /)
  730 FORMAT (//,10X, 35H$$$$ SANTIAGO UHF CLOSED SHELL $$$$.//)
  740 FORMAT (1H ,5X, 16H EOF OR EOR READ)
  800 FORMAT (//10X,44HTRIAL SET OF ATOMIC COORDINATES (ANGSTROMS) //4X,
     18HATOM NO ,9X,10HATOMIC NO ,9X,12HX-COORDINATE,8X,12HY-COORDINATE,
     28X,12HZ-COORDINATE,7X,15HATOMIC ORBITALS//)
  810 FORMAT (8X,I2,16X,I2,14X,3(F10.6,10X),I3,4H TO ,I2)
  820 FORMAT (//19H MOLECULAR CHARGE = I3,10X,I3,15H ELECTRONS, AND,I3,2
     10H DOUBLY OCCUPIED MOS)
C
      END
C
C ****************
C
      SUBROUTINE DFPMIN
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
C *
C  THIS SUBROUTINE ATTEMPTS TO MINIMIZE A REAL-VALUED FUNCTION F OF
C  THE N-COMPONENT REAL VECTOR X BY THE DAVIDON-FLETCHER-POWELL
C  ALGORITHM. THE USER MUST SUPPLY THE SUBROUTINE COMPFG(N,X,F,G)
C  WHICH COMPUTES FUNCTION VALUES F AND GRADIENTS G AT GIVEN
C  VALUES FOR THE VARIABLES X. THE MINIMIZATION PROCEEDS BY A
C  SEQUENCE OF ONE-DIMENSIONAL MINIMIZATIONS. SOLVE THE SUBPROBLEM OF
C  MINIMIZING THE FUNCTION F ALONG THE LINE X+ALPHA*P. WHERE X
C  IS THE VECTOR OF CURRENT VARIABLE VALUES. ALPHA IS A SCALAR
C  VARIABLE, AND P IS A SEARCH-DIRECTION VECTOR PROVIDED BY THE
C  DAVIDON-FLETCHER-POWELL ALGORITHM. EACH ITERATION STEP CARRIED
C  OUT BY DFPMIN PROCEEDS BY FINDING A VALUE FOR ALPHA WHICH
C  MINIMIZES F ALONG X+ALPHA*P, BY UPDATING THE VECTOR X BY THE
C  AMOUNT ALPHA*P, AND FINALLY BY GENERATING A NEW VECTOR P.
C  THE CONVERGENCE IS TESTED BY THE 3-ELEMENT VECTOR TOLEND.
C  THE TEST ON X IS SATISFIED IF THE RELATIVE CHANGE IN X, MEASURED
C  BY ITS NORM, OVER ANY TWO SUCCESSIVE ITERATIONS DROPS BELOW
```

```
C  TOLEND(1)  THE TEST ON G IS SATISFIED  IF THE RMS VALUE OF THE
C  GRADIENT IS LESS THAN TOLEND(3)
C  A SUCCESSFUL TERMINATION OF THE OPTIMIZATION OCCURS WHEN THE
C  TESTS ON X AND G (NSUCC=1) ARE SATISFIED
C  AN UNSUCCESSFUL TERMINATION OCCURS WHEN THE NUMBER OF FUNCTION
C  EVALUATIONS EXCEEDS MAXEND (NSUCC=11)
C  *
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
      PARAMETER (NUMH=(3*NUMATM-6)*(3*NUMATM-5)/2)
      COMMON
     /UNITS / IN,IO,JO,LINES,LITOT
     /BESAFE/ TLIMIT,SFDFP,SFSCF
     /CYCLES/ JCYC,NPETER
     /DFP   / X(I3NX),NVAR
     /DFPGO / LINMAX,MXCALL,GLST(I3NX),XLAST(I3NX),H(NUMH)
     /FLAGS / SCFCRT,COMENT(13),KITSCF(8),MIDDLE,
              ITIME,NTYPE,KRESET,IWADE,NTO,NREM,MAXEND
     /FLPOCM/ TOLN(3),FC,G(I3NX),DX(I3NX),DOT,ATOT,TOTALE,
              NNN,MXLN,ISWTCH,I113
     /ERG   / ENERG,GTOT(I3NX)
     /SHFTCM/ ZCURVE,ETARG,MAXZ
     /PARM1 / XXDUM(I3NX),NATMD,IDDUM(NUMATM,5),LOC(2,I3NX),IDD(2)
     /PASS2 / TOLIN(3),TOLEND(6),MAXLIN(4),NSUCC,I114
     /PASS3 / COS,PNORM,JPRINT,JNRST,NCNT,I116,LNSTOP,IREPET
     /SC    / SCFF,EINC,CCCOS,EYEAD,IGG1,I115
     /SLVER / FREPF,YEAD,DEL
     /XPRJU / IREP,IJUMP
     /XXMAXM/ XMAXM,PMSTE,DELL,TDEL
     COMMON
     /MMSYNC/ NDHA,NEWDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
     /SAVCOM/ TIMEO,LINCNT
     /ATOMS / C(3,NUMATM),NATMS,JDUM(I3NX)
     /INTVAR/ VF(I3NX),VL(I3NX),Z(I3NX),BF(MPAK),BL(MPAK),DFL,
              LOCVI(I3NX),MAX
     /FCOMN / CORE(NUMATM),WDS(NUMATM),NORB(NUMATM)
     /ANTDER/ ANALY,JSKPA
     /SYNCOM/ CC(I3NX,5),S(3),P(6),TARG,SC,NVC(I3NX),KKK(14)
     DIMENSION XSAV(I3NX),GSAV(I3NX),D(I3NX),DG(I3NX),Y(I3NX)
     LOGICAL ANALY
     DATA ZERO,ONE,TWO,FIVE,PI/0 D0,1 D0,2 D0,5 D0,3 1415926536D0/
C INPUT OPTIONS:
C IGR  REDUCE GRADIENT ALONG D VECTOR TO 0 1**IGR * ORIGINAL VALUE
C IUPDAT NE 0  TAKE NEW STEP TO INITIALIZE HESSIAN
C IEXTRP       MAXIMUM NUMBER OF STEPS ON LINE  DEFAULT=2
C KFLG EQ 0  UPDATE H FROM X-XLAST
C KFLG EQ 1  UPDATE H FROM X-XSAV
C KFLG EQ 2  UPDATE H FROM X-(XLAST OR XSAV, DEPENDING ON WHICH E IS LOWER
```

```
C LFLG NE 0  DONT ALLOW D TO GET LARGER THAN XMAX AFTER CPHF
C MFLG NE 0  CONSTRAIN HOW SMALL ALPHA CAN GET AT START OF LINE
      READ (IN,40) METHOD,IGR,IUPDAT,IEXTRP,KFLG,LFLG,MFLG
      WRITE(JO,40) METHOD,IGR,IUPDAT,IEXTRP,KFLG,LFLG,MFLG
      IF(METHOD.LT.1.OR.METHOD.GT.3) RETURN
      IF(IEXTRP.LE.0) IEXTRP=4
      IF(IGR.LE.0) IGR=2
      GR=0.1D0*IGR
   40 FORMAT(8I2)
C     *
C     INITIALIZE CERTAIN VARIABLES
C     *
      GMIN=1.D-4
      HMIN0=1.D-2
      TDEL=3.D0
      DELX=1.D-2
      TDLX=TDEL*DELX
      XMAX=5.D0*TDLX
      XMAXL=XMAX*10.D0
      IF(LFLG.NE.0) XMAXL=XMAX
      GVM=TOLEND(3)
      XVM=TOLEND(1)
C     IF(GVM.LE.ZERO) GVM=5.D-2
      GVM=0.02D0
      XVM=0.0001D0
C     IF(XVM.LE.ZERO) XVM=2.D-3
      GVM2=GVM+GVM
      DELTA=2.D0
      IF(NTRAN.EQ.0) DELTA=5.D0
      IHFLAG=0
      IJUMP=0
      SCFGRD=SCFF
      IF(SCFGRD.GE.1.D-4) SCFF=1.D-5
      FREPF=9.D6
      DPEDY=ZERO
      DPEDZ=ZERO
      LNSTOP=1
      IWADE=1
      IREPET=1
      JNRST=0
      NH=NVAR*(NVAR+1)/2
      TWOPI=PI+PI
      PTH1=ZERO
      IF(NTRAN.NE.0) CALL PTHCRD(X,PTH1,NVAR)
      WRITE(JO,1576)(I,NORB(I),CORE(I),(C(J,I),J=1,3),WDS(I),I=1,NATMS)
      TARGET=PTH1
      PTH2=ZERO
      ZCURVE=ZERO
```

```
      SQRTN=DSQRT(DFLOAT(NVAR))
      IF(NTRAN.EQ.0) MAX=0
      IMAX=IABS(MAX)
      DZ=ZERO
      GZ=ZERO
      EPS=1.D-10
      INVERT=0
      AMIN=0.1D0
      DP=(P(3)-P(1))*0.1D0
      P1=P(1)+DP
      P3=P(3)-DP
C
      IF(MIDDLE.NE.1) GOTO 90
C PRINT POINT STARTING FROM  IF THIS IS A CONTINUATION OPTIMIZATION
C
      WRITE(IO,170) (X(I),I=1,NVAR)
      WRITE(IO,190) (G(I),I=1,NVAR)
      IF(IMAX.LT.2) WRITE(IO,190) (GTOT(I),I=1,NVAR)
      I=4.01+0.2*FLOAT(NVAR)
      IF(LINES+8+I.GT.LITOT) CALL PAGE
      LINES=LINES+8+I
      WRITE(IO,6200) 'CONTINUE'
      IF(METHOD.EQ.1) WRITE(IO,6210)
      IF(METHOD.EQ.2) WRITE(IO,6220)
      IF(METHOD.EQ.3) WRITE(IO,6230)
      WRITE(IO,6250)
      WRITE(IO,6600) XVM,GVM
      WRITE(IO,85) NVAR
   85 FORMAT(/I10,' VARIABLES WILL BE OPTIMIZED. CURRENT VALUES ARE :/)
      WRITE(IO,170) (X(I),I=1,NVAR)
      WRITE(IO,180)
      WRITE(IO,190) (GTOT(I),I=1,NVAR)
      IF(NTRAN.NE.0) WRITE(IO,195)
  195 FORMAT(/10X,'THE CONSTRAINT AXIS FOR THE OPTIMIZATION IS :/)
      IF(NTRAN.NE.0) WRITE(IO,2247)(Z(I),I=1,NVAR)
 2247  FORMAT(' Z :',13F10.5)
      GRMS=ZERO
      GZ=ZERO
        DO 80 I=1,NVAR
        IF(NTRAN.EQ.0) GOTO 80
        GZ=GZ+GTOT(I)*Z(I)
   80   GRMS=GRMS+GTOT(I)**2
      GRMSN=DSQRT(GRMS)
      GRMS=GRMSN/SQRTN
      TIME=6.D1*CLOCK(I)
      TIMAX=TLIMIT/DFLOAT(MAXEND)
      ELAST=TOTALE
      IF(NTRAN.NE.0) CALL ELECPC(EPC)
```

```
      IF(LINES+5.GT.LITOT) CALL PAGE
      LINES=LINES+5
      WRITE(IO,5800)
      WRITE(IO,5900)JCYC,NCNT,TOTALE,ZERO,GRMS,EPC,PTH1,GZ,ZERO,
     1ZERO,ZERO,ZERO,TIME,ZERO,GRMSN,PTH2,ZCURVE,ZERO,ZERO,ZERO
      GRMTOP=GRMS
      GOTO 1605
C
   90 STIME=6.D1*CLOCK(I)
      TARGET=PTH1
      EPCGRD=ZERO
      LINMAX=0
      MXCALL=3
C
      CALL COMPFG(NVAR,X,TOTALE,G,GZ,GMAX)
      IF(NTRAN.NE.0) CALL ELECPC(EPC)
      ETARG=EPC
      CALL PAGE
      I=4.01+0.2*FLOAT(NVAR)
      LINES=LINES+8+I
      WRITE(IO,6200) '  START'
      IF(METHOD.EQ.1) WRITE(IO,6210)
      IF(METHOD.EQ.2) WRITE(IO,6220)
      IF(METHOD.EQ.3) WRITE(IO,6230)
      WRITE(IO,6250)
      WRITE(IO,6600) XVM,GVM
      WRITE(IO,85) NVAR
      WRITE(IO,170) (X(I),I=1,NVAR)
      WRITE(IO,180)
      WRITE(IO,190) (GTOT(I),I=1,NVAR)
  170 FORMAT(' X',13F10.5)
  180 FORMAT(1X)
  190 FORMAT(' G',13F10.5)
      GRMS=ZERO
      DO 400 I=1,NVAR
        XSAV(I)=X(I)
        GSAV(I)=G(I)
  400   GRMS=GRMS+GTOT(I)**2
      GRMSN=DSQRT(GRMS)
      GRMS=GRMSN/SQRTN
      KPOINT=0
      IF(LINES+5.GT.LITOT) CALL PAGE
      WRITE(IO,5800)
      LINES=LINES+5
      TIME=6.D1*CLOCK(I)
      TIMAX=3.D0*(TIME-STIME)
      WRITE(IO,5900)JCYC,NCNT,TOTALE,ZERO,GRMS,EPC,PTH1,GZ,ZERO,
     1ZERO,ZERO,ZERO,TIME,DZ,GRMSN,PTH2,ZCURVE,ZERO,ZERO,ZERO
```

```
      GRMTOP=GRMS
C
C TAKE STEP OF INFINTESIMAL LENGTH ALONG G TO INITIALIZE THE HESSIAN
C
  600 ELAST=TOTALE
      INVERT=0
      GRMSL=GRMS
      EPCL=EPC
      GD0=ZERO
      A=2.D-2
      IF(JCYC.GT.0) A=DMAX1(1.D-4,0.5D0*XRMS)
      DLEN=A*SQRTN
      DO 620 I=1,NVAR
        D(I)=-DSIGN(A,G(I))
  620   GD0=GD0+G(I)*D(I)
      IF(IMAX.EQ.2 .AND. IHFLAG.LE.2) THEN
        WRITE(6,2243)(D(I),I=1,NVAR)
        CALL CPHF(X,D,NVAR,EPC,DPEDY,DPEDZ)
        BETA=-DPEDY*DLEN/DPEDZ
        WRITE(6,812) DLEN,BETA,GD0
        DLEN=ZERO
        GD0=ZERO
        DO 640 I=1,NVAR
          D(I)=D(I)+BETA*Z(I)
          GD0=GD0+G(I)*D(I)
  640     DLEN=DLEN+D(I)**2
        DLEN=DSQRT(DLEN)
        WRITE(6,812) DLEN,BETA,GD0
      ENDIF
      WRITE(6,2243)(D(I),I=1,NVAR)
      DO 660 I=1,NVAR
        XLAST(I)=X(I)
        X(I)=X(I)+D(I)
        IF(LOC(2,I).EQ.1) GOTO 660
          IF(X(I).GT. PI) X(I)=X(I)-TWOPI
          IF(X(I).LT.-PI) X(I)=X(I)+TWOPI
  660   GLST(I)=G(I)
      CALL COMPFG(NVAR,X,TOTALE,G,GZY,GMAX)
      WRITE(IO,170) (X(I),I=1,NVAR)
C
      WRITE(IO,190) (G(I),I=1,NVAR)
      IF(NTRAN.EQ.0) GOTO 680
        CALL PTHCRD(XLAST,PTH1,NVAR)
        CALL PTHCRD(X,PTH2,NVAR)
        CALL ELECPC(EPC)
        DY=A*SQRTN
C       DPGDY=(PTH2-PTH1)/DY
C
```

```
 680 GRMS=ZERO
     DO 700 I=1,NVAR
 700    GRMS=GRMS+GTOT(I)**2
     GRMSN=DSQRT(GRMS)
     GRMS=GRMSN/SQRTN
     IF(LINES+3 LE LITOT) GOTO 720
      CALL PAGE
      WRITE(IO,5800)
      LINES=LINES+2
 720 LINES=LINES+3
     TIME=6 D1*CLOCK(I)
     WRITE(IO,5900)JCYC,NCNT,TOTALE,A,GRMS,EPC,PTH1,GZ,ZERO,ZERO,
    1 DPEDY,A,TIME,DZ,GRMSN,PTH2,ZCURVE,ZERO,ZERO,DPEDZ
     WRITE(JO,1574) JCYC,NCNT,TOTALE,GRMS,TIME
1574 FORMAT(/' LINE ',I3,I5,' CALCS. ',F9 3,' KCALS/MOL.  GRMS = '
    1 F8 3,F10 3,' SECS')
     WRITE(JO,1576)(I,NORB(I),CORE(I),(C(J,I),J=1,3),WDS(I),I=1,NATMS)
     IF(NTRAN NE 0) WRITE(JO,1575)PTH,EPC
1575 FORMAT('PATH COORDS  GEOMETRIC = ',F9 5,'  ELECTRONIC = ',F9 5)
1576 FORMAT(I2,I3,F6 1,4X,3F10 5,5X,A4)
C
C INITIALIZE HESSIAN
C
1000 DO 1100 I=1,NVAR
       DX(I)=X(I)-XLAST(I)
       IF(LOC(2,I) EQ 1) GOTO 1100
        IF(DX(I) GT  PI) DX(I)=DX(I)-TWOPI
        IF(DX(I) LT -PI) DX(I)=DX(I)+TWOPI
1100   DG(I)=G(I)-GLST(I)
     GOTO 1151

1150 IF(IUPDAT NE 0) GOTO 600
1151 SXG=ZERO
     SG=ZERO
     DO 10 I=1,NVAR
       IF((DABS(DX(I)) LT 5 D-5 OR DABS(DG(I)) LT 5 D-5)  AND
    1     INVERT GT 0) GOTO 600
       DXG=DX(I)*DG(I)
       IF(DXG LE ZERO) GOTO 10
        SXG=SXG+DXG
        SG=SG+DG(I)**2
  10   CONTINUE
     INVERT=0
     HAV=ZERO
     IF(SXG GT ZERO) HAV=SXG/SG
     HMIN=0 5D0*HAV
     IF(HMIN EQ ZERO) HMIN=HMIN0
C
```

```
      SXG=ZERO
      GHG=ZERO
      II=0
      DO 15 I=1 NVAR
      DO 15 J=1 I
        II=II+1
        IF(J LT I) THEN
          H(II)=ZERO
        ELSE
          IF(DABS(DG(I)) LT GMIN) THEN
            H(II)=HMIN
          ELSE
            GABS=DABS(GLST(I))
            IF(TOTALE LT ELAST) GABS=DABS(G(I))
            H(II)=DX(I)/DG(I)
            IF(H(II) LT ZERO AND GABS LT GMIN) H(II)=HMIN
            IF(H(II) LT ZERO) H(II)=TDLX/GABS
          ENDIF
          IF(GABS LT GMIN) GABS=GMIN
          HMAX=TDLX/GABS
          IF(H(II) GT HMAX) H(II)=HMAX
          IF(METHOD EQ 4) H(II)=0 01D0
          SXG=SXG+DX(I)*DG(I)
          GHG=GHG+DG(I)**2*H(II)
        ENDIF
   15   CONTINUE
C
C     CALL VECPRT(H,NVAR)
C     CALL HDIAG(H,NVAR)
      IF(NCNT GE MAXEND) GOTO 5700
      IF(JCYC EQ 0)THEN
        ATOT=DMIN1(ONE,GHG/SXG/FIVE)
        ATOT=DMAX1(ATOT,TWO/FIVE)
      ENDIF
C
      IF(TOTALE LE ELAST  AND  GRMS LT 1 75D0*GRMSL) GOTO 1600
        DO 800 I=1 NVAR
          X(I)=XSAV(I)
  800     G(I)=GSAV(I)
        IF(GRMS GT 2 D0*GRMSL)THEN
          CALL COMPFG(NVAR,X,TOTALE,G,GZ,GMAX)
          IF(NTRAN NE 0) CALL ELECPC(EPC)
          GRMS=ZERO
          DO 805 I=1,NVAR
  805       GRMS=GRMS+G(I)**2
          GRMSN=DSQRT(GRMS)
          GRMS=GRMSN/SQRTN
        ELSE
```

```
        EPC=EPCL
        GRMS=GRMSL
        GRMSN=GRMS*SQRTN
        TOTALE=ELAST
       ENDIF
C
C    DETERMINE SEARCH DIRECTION
C
1600  WRITE(IO,170) (X(I),I=1,NVAR)
     WRITE(IO,190) (G(I),I=1,NVAR)
     IF(NTRAN.NE.0.AND.IMAX.LT.2) WRITE(IO,190) (GTOT(I),I=1,NVAR)
     CALL PUNOUT
     TIME=6.D1*CLOCK(I)
     DT=TIME-STIME
     SAFETY=SFDFP*DT
     IF(TIME+SAFETY.LE.TLIMIT) GOTO 1605
      IF(LINES+15.GT.LITOT) CALL PAGE
      WRITE(IO,6150) TIME,TLIMIT
      IJUMP=1
      CALL PUNOUT
      STOP
C
 1605 IF(INVERT.GE.2) GOTO 1150
     ALAST=ZERO
     IHFLAG=IHFLAG+1
     A=ATOT
     STIME=TIME
     GRMSL=GRMS
     GRMSAV=GRMS
     ELAST=TOTALE
     ESAV=TOTALE
     KPOINT=0
     DLEN=ZERO
     GD0=ZERO
     DMAX=ZERO
     KK=0
     DO 2100 I=1,NVAR
       GSAV(I)=G(I)
       XSAV(I)=X(I)
       D(I)=ZERO
       DO 1800 J=1,I
 1800      D(I)=D(I)+H(KK+J)*G(J)
       IF(I.GE.NVAR) GOTO 2000
        JJ=I+1
        K=KK+I
        KK=K
        DO 1900 J=JJ,NVAR
          K=K+J-1
```

```
 1900        D(I)=D(I) + H(K)*G(J)
 2000    D(I)= -D(I)
       ABSDI=DABS(D(I))
       IF(ABSDI GT XMAX) D(I)=DSIGN(XMAX D(I))
       DMAX=DMAX1(DMAX ABSDI)
       DLEN=DLEN + D(I)**2
 2100    GD0=GD0 + G(I)*D(I)
     WRITE(IO,2243)(D(I),I=1,NVAR)
     DLEN=DSQRT(DLEN)
     GD0N=GD0/DLEN
     IF(GD0 LT ZERO) GOTO 2210
 2150 IF(LINES+4 GT LITOT) CALL PAGE
     LINES=LINES+4
     IF(IHFLAG GT 3) GOTO 2170
     WRITE(IO,2160) GD0
 2160 FORMAT(//15X,'*** THE HESSIAN IS NOT POSITIVE DEFINITE  GD0 =
    1 F9 3,'  THE HESSIAN WILL BE RECOMPUTED ***'/)
     GOTO(1000,600,600,2170),IHFLAG
 2170 WRITE(IO,2180)
 2180 FORMAT(//10X,'THE HESSIAN IS NOT POSITIVE DEFINATE IN THIS '
              'REGION  USE THE BEST STRUCTURE OBTAINED SO FAR'
           /10X,'IN A QST PATH MAXIMUM  IF THIS DOES NOT '
              'WORK, FLANKING POINTS WILL HAVE TO BE USED ')
     STOP
C
 2210 IF(NTRAN EQ 0)  GOTO 2250
COMPUTE PATH COORDS OF BASE POINT, RESULT OF Y SHIFT, & SHIFT ALONG Z
     CALL PTHCRD(XSAV,PTH1,NVAR)
     PTHSAV=PTH1
     IF(IMAX EQ 1) THEN
C ORTHOGONALIZE D TO CONSTRAINT AXIS
       DZ=A*DLEN
       FACT=DZ
       DO 2211 I=1,NVAR
         X(I)=XSAV(I)+A*D(I)
         XLAST(I)=XSAV(I)+FACT*Z(I)
         IF(LOC(2,I) EQ 1) GOTO 2211
           IF(X(I) GT  PI) X(I)=X(I)-TWOPI
           IF(X(I) LT -PI) X(I)=X(I)+TWOPI
           IF(XLAST(I) GT  PI) XLAST(I)=XLAST(I)-TWOPI
           IF(XLAST(I) LT -PI) XLAST(I)=XLAST(I)+TWOPI
 2211      CONTINUE
      CALL PTHCRD(XLAST,PTHZ,NVAR)
      CALL PTHCRD(X,PTH2,NVAR)
      PTHY=PTH2
     WRITE(6,1234)PTHSAV,PTH1,PTH2,PTHY,PTHZ
1234  FORMAT(' PTHSAV,PTH1,PTH2,PTHY,PTHZ',5F10 5)
      CALL ORTHOG(A,Z,D,XSAV,LOC,DLEN,TARGET,PTHY,PI,TWOPI,XMAXL)
```

```
      DY=A*DLEN
      DPGDY=(PTHY-PTH1)/DY
      DPGDZ=(PTHZ-PTH1)/DZ
     WRITE(6,1234)PTHSAV,PTH1,PTH2,PTHY,PTHZ
      BETA=ZERO
     ELSE
      IF(IMAX.EQ.2) THEN
C ADJUST D TO HOLD CONSTANT ELECTRONIC PATH COORDINATE
       CALL CPHF(XSAV,D,NVAR,EPC,DPEDY,DPEDZ)
       BETA=-DPEDY*DLEN/DPEDZ
       WRITE(6,812) DLEN,BETA,GD0
      ENDIF
     ENDIF
     DLEN=ZERO
     GD0=ZERO
     DMAX=ZERO
     DO 2242 I=1,NVAR
      X(I)=XSAV(I)
      D(I)=D(I)+BETA*Z(I)
      ABSDI=DABS(D(I))
      IF(ABSDI.GT.XMAXL) D(I)=DSIGN(XMAX,D(I))
      ABSDI=DABS(D(I))
      DMAX=DMAX1(DMAX,ABSDI)
      DLEN=DLEN+D(I)**2
 2242   GD0=GD0+G(I)*D(I)
812  FORMAT(' DLEN,BETA,GD0 ',3F12.6)
     DLEN=DSQRT(DLEN)
     GD0N=GD0/DLEN
     WRITE(6,812) DLEN,BETA,GD0
2243  FORMAT(' D ',13F10.6)
     WRITE(6,2243)(D(I),I=1,NVAR)
     IF(GD0.GT.ZERO) WRITE(6,2240)
 2240 FORMAT(' ***  WARNING  *** GD0 IS POSITIVE AFTER THE CPHF
    1 'ADJUSTMENT  ***')
C
C    FIND MINIMUM ALONG THIS SEARCH DIRECTION
C    DETERMINE STEP LENGTH BY CUBIC INTERPOLATION
C
 2250 A=DELTA*ATOT
     A=DMIN1(ONE,A)
     IF(A*DMAX.GT.XMAX) A=XMAX/DMAX
     IF(A.LE.ZERO) A=DMAX1(1.D-2,ATOT)
     ATOT=A
C
 2300 KPOINT=KPOINT+1
     XRMS=ZERO
     DO 2310 I=1,NVAR
      XLAST(I)=X(I)
```

```
      X(I)=XSAV(I)+A*D(I)
      XRMS=XRMS+D(I)**2
      IF(LOC(2,I) EQ 1) GOTO 2310
       IF(X(I) GT  PI) X(I)=X(I)-TWOPI
       IF(X(I) LT -PI) X(I)=X(I)+TWOPI
 2310   GLST(I)=G(I)
    XRMS=A*DSQRT(XRMS)/SQRTN
C
    CALL COMPFG(NVAR,X,TOTALE,G,GZ,GMAX)
    IF(NTRAN.NE.0) THEN
     CALL ELECPC(EPC)
     CALL PTHCRD(X,PTH2,NVAR)
    ENDIF
C
    WRITE(IO,170) (X(I),I=1,NVAR)
    WRITE(IO,190) (G(I),I=1,NVAR)
    IF(NTRAN.NE.0.AND.IMAX.LT.2) WRITE(IO,190) (GTOT(I),I=1,NVAR)
    GDA=ZERO
    GDLAST=ZERO
    GRMS=ZERO
    DO 2800 I=1,NVAR
      GRMS=GRMS+GTOT(I)**2
      GDA=GDA+G(I)*D(I)
 2800   GDLAST=GDLAST+GLST(I)*D(I)
    GDAN=GDA/DLEN
    GRMSN=DSQRT(GRMS)
    GRMS=GRMSN/SQRTN
    EA=TOTALE
    IF(NTRAN.EQ.0 .OR. IMAX.NE.1  OR  KPOINT.GT.1) GOTO 2845
C
C SHOULD D BE ADJUSTED TO REDUCE GRADIENT ALONG Z?
C IF NOT, END LINE SEARCH HERE. DONT SEARCH ANY FURTHER ON THIS LINE
C
    IF(DABS(GZ).LT.DMAX1(GVM,GRMS*0.3D0)) GOTO 3200
C PRINT RESULT OF Y SHIFT HERE
    WRITE(6,1234)PTHSAV,PTH1,PTH2,PTHY,PTHZ
    TIME=6.D1*CLOCK(I)
    WRITE(IO,5910) NCNT,TOTALE,A,GRMS,EPC,PTHY,GZ,GD0,GDA,DPGDY,
   1 XRMS,TIME,DZ,GRMSN,PTH2,ZCURVE,GD0N,GDAN,DPGDZ
C
C GLST IS ONLY USED IN THE INITIAL CALL, AND IS NOT CHANGED HERE
C XSAV IS THE SAME AS XLAST AT THIS POINT, SO CAN BE USED IN SUBR
C
    CALL ZSHFT(A,EPC,G,DUM,XSAV,GZ,LOC,PI,TOTALE,GVM,GMAX)
C
    GDLAST=ZERO
    GDA=ZERO
    GRMS=ZERO
```

```
      DLEN=ZERO
      DO 2830 I=1,NVAR
        GRMS=GRMS+GTOT(I)**2
        XSAV(I)=XLAST(I)
        DXX=X(I)-XLAST(I)
        IF(LOC(2,I).EQ.1) GOTO 2820
          IF(DXX.GT. PI) DXX=DXX-TWOPI
          IF(DXX.LT.-PI) DXX=DXX+TWOPI
 2820   D(I)=DXX/A
        DLEN=DLEN+D(I)**2
        GDLAST=GDLAST+GLST(I)*D(I)
 2830   GDA=GDA+G(I)*D(I)
      DLEN=DSQRT(DLEN)
      GDAN=GDA/DLEN
      GD0=GDLAST
      GD0N=GD0/DLEN
      GRMSN=DSQRT(GRMS)
      GRMS=GRMSN/SQRTN
      XRMS=DSQRT(XRMS)/SQRTN
      EA=TOTALE
      CALL PTHCRD(X,PTH2,NVAR)
C
C IF SLOPE IS SMALL ENOUGH AND NEW ENERGY IS LOWER, END LINE SEARCH
C
 2845 IF(DABS(GDA/GD0).LT.GR .AND. EA.LT.ELAST) GOTO 3200
      IF(GRMS.LE.GVM) GOTO 3200
C INTERPOLATE?
      IF(NTRAN.EQ.0.OR.GDA.GT.ZERO.OR.EA.GE.ELAST) GOTO 2847
C ONLY INPUT NUMBER OF EXTRAPOLATIONS IF CONSTRAINED
      IF(KPOINT.GE.IEXTRP)GOTO 3200
      IF(IMAX.EQ.2 .AND. DABS(EPC-ETARG).GT.0.03D0) GOTO 3200
C
 2847 IF(LINES+3.LE.LITOT) GOTO 2850
      CALL PAGE
      WRITE(IO,5800)
      LINES=LINES+2
 2850 LINES=LINES+3
      TIME=6.D1*CLOCK(I)
      WRITE(IO,6700) NCNT,TOTALE,A,GRMS,EPC,PTH1,GZ,GD0,GDA,DPEDY,
     1 XRMS,TIME,DZ,GRMSN,PTH2,ZCURVE,GD0N,GDAN,DPEDZ
C
      IF(GDA.GT.ZERO.OR.EA.GE.ELAST) GOTO 2870
      IF(A.GE.TWO*DELTA**2) GOTO 3200
C
C  ESTIMATE HOW LARGE THE NEXT EXTRAPOLATION INCREMENT SHOULD BE
C
      ALAST=A
      A=DABS(A*GD0/(GD0-GDA))
```

```
      IF(A GT 0 05D0) A=DMIN1(A ALAST*DELTA)
C       A=DMIN1(GD0/(GD0-GDA)/DELTA ONE)*A
C       IF(A LT ZERO) A=ALAST
C       A=A*DELTA
      A=DMIN1(A TWO*DELTA**2)
      ATOT=A
      ELAST=EA
      IF(A/ALAST GT 1 1D0) GOTO 2300
      GOTO 3200
C
C   INTERPOLATE TO FIND THE MINIMUM
C
 2870 AX=A-ALAST
      ZZ=3 D0*(ELAST-EA)/AX+GDA+GDLAST
      W=DSQRT(DABS(ZZ**2-GDA*GDLAST))
CUBIC INTERPOLATION
      AX3=(ONE-(GDA+W-ZZ)/(GDA-GDLAST+W+W))
C QUADRATIC APPROXIMATION
      DELG=GDA-GDLAST
      AX2=-GDLAST/DELG
      AX=AX3
      IF(AX2 GT ZERO AND AX2 LT DABS(AX3-AX2)) AX=AX2
      WRITE(6,2222)A,ALAST,ZZ,GDA,GDLAST,W,AX2,AX3,EA,ELAST
2222  FORMAT(' A,ALAST,ZZ,GDA,GDLAST,W,AX2,3,EA,ELAST ',8F9 5,2F9 4)
      IF(AX/(A-ALAST) LT 0 1D0 AND EA LT ELAST) GOTO 3200
C
      DO 2885 I=1,NVAR
        XL=X(I)-XLAST(I)
        IF(LOC(2,I) EQ 1) GOTO 2877
          IF(XL GT PI) XL=XL-TWOPI
          IF(XL LT -PI) XL=XL+TWOPI
 2877     XL2=XLAST(I)
        XLAST(I)=X(I)
        X(I)=XL2+AX*XL
        IF(LOC(2,I) EQ 1) GOTO 2885
          IF(X(I) GT PI) X(I)=X(I)-TWOPI
          IF(X(I) LT -PI) X(I)=X(I)+TWOPI
 2885     GLST(I)=G(I)
      IF(NTRAN EQ 0) GOTO 3000
        CALL PTHCRD(XLAST,PTH1,NVAR)
        CALL PTHCRD(X,PTH2,NVAR)
C
 3000 A=ALAST+AX*(A-ALAST)
      CALL COMPFG(NVAR,X,TOTALE,G,GZ,GMAX)
      IF(NTRAN NE 0) CALL ELECPC(EPC)
      WRITE(IO,170) (X(I),I=1,NVAR)
      WRITE(IO,190) (G(I),I=1,NVAR)
      IF(NTRAN NE 0 AND IMAX LT 2) WRITE(IO,190) (GTOT(I),I=1,NVAR)
```

```
      WRITE(6,19)TOTALE,EA,ELAST,ESAV
 19   FORMAT(' TOTALE,EA,ELAST,ESAV ',4F12.6)
      IF(KFLG.EQ.2 .AND. EA.GT.ESAV) THEN
       EA=ESAV
       DO 3190 I=1,NVAR
         XLAST(I)=XSAV(I)
 3190    GLST(I)=GSAV(I)
      ENDIF
C
 3200 ELAST=EA
      IF(MFLG.NE.0) THEN
       IF(A.LT.AMIN .AND. AMIN.GT.1.D-4) AMIN=AMIN*0.5D0
       ATOT=DMAX1(A,AMIN)
      ENDIF
      XRMS=ZERO
      GRMS=ZERO
      GDA=ZERO
      DXSQ=ZERO
      DO 3300 I=1,NVAR
        GRMS=GRMS+GTOT(I)**2
        GDA=GDA+G(I)*D(I)
        DXL=XLAST(I)
        DGL=GLST(I)
        IF(KFLG.EQ.1) THEN
          DXL=XSAV(I)
          DGL=GSAV(I)
        ENDIF
        DX(I)=X(I)-DXL
        DG(I)=G(I)-DGL
        DXX=X(I)-XSAV(I)
        IF(LOC(2,I).EQ.1) GOTO 3250
          IF(DXX.GT. PI) DXX=DXX-TWOPI
          IF(DXX.LT.-PI) DXX=DXX+TWOPI
          IF(DX(I).GT. PI) DX(I)=DX(I)-TWOPI
          IF(DX(I).LT.-PI) DX(I)=DX(I)+TWOPI
 3250   DXSQ=DXSQ+DX(I)**2
 3300   XRMS=XRMS+DXX**2
      GRMSN=DSQRT(GRMS)
      GRMS=GRMSN/SQRTN
      XRMS=DSQRT(XRMS)/SQRTN
      GDAN=GDA/DLEN
C
C     PRINT SUMMARY
C
      JCYC=JCYC+1
      LINMAX=LINMAX+1
      IHFLAG=0
      IF(LINES+3.LE.LITOT) GOTO 3310
```

```
      CALL PAGE
      WRITE(IO 5800)
      LINES=LINES+2
 3310 LINES=LINES+3
     TIME=6 D1*CLOCK(I)
     WRITE(IO 5900)JCYC NCNT TOTALE A GRMS EPC PTH1 GZ GD0 GDA
    1 DPEDY XRMS TIME DZ GRMSN PTH2 ZCURVE GD0N GDAN DPEDZ
     WRITE(JO 1574) JCYC NCNT TOTALE GRMS TIME
     WRITE(JO 1576)(I NORB(I) CORE(I) (C(J I) J=1 3) WDS(I) I=1 NATMS)
     IF(NTRAN NE 0) THEN
       PTH=PTH2
       WRITE(JO 1575)PTH EPC
       IF(IMAX EQ 1) TARGET=PTH
     ENDIF
C SEE IF CONVERGENCE HAS BEEN ACHIEVED
     IF(GRMS LE GVM  AND  GMAX LE GVM2  AND  XRMS LE XVM) GOTO 5200
     WRITE(6 6969) TOTALE ESAV GRMS GRMSAV GRMSL
 6969  FORMAT(' TOTALE ESAV GRMS SAV SL ' 5F10 5)
     IF( (TOTALE GT ESAV  AND  GRMS GT GRMSAV)
    1  OR  GRMS GT 10 D0*GRMSAV) THEN
       DO 3329 I=1 NVAR
         X(I)=XSAV(I)
         XLAST(I)=XSAV(I)
         GLST(I)=GSAV(I)
 3329      G(I)=GSAV(I)
       TOTALE=ESAV
       GRMS=GRMSAV
       PTH=PTHSAV
     ENDIF
C
     IF(NCNT GE MAXEND) GOTO 5700
     WRITE(6 3331)(DX(I) I=1 NVAR)
 3331  FORMAT(' DX' 13F10 5)
     WRITE(6 3332)(DG(I) I=1 NVAR)
 3332  FORMAT(' DG' 13F10 5)
     IF(METHOD EQ 4) GOTO 5020
C    UPDATE H
     HTIME=6 D1*CLOCK(I)
     KK=0
     IF(METHOD EQ 3) THEN
C UPDATE IS TO THE HESSIAN  NOT ITS INVERSE  FIRST INVERT H
       REWIND 3
       WRITE(3) (H(I) I=1 NH)
       CALL DSINV(H NVAR EPS IER)
       IF(IER NE 0) THEN
         WRITE(IO 5120)
         WRITE(IO 5122)
 5122     FORMAT('+' 54X 'IT CANNOT BE OBTAINED FROM THE INVERSE'
```

```
          HESSIAN BEFORE THE UPDATE ')
        INVERT=2
        GOTO 5020
      ENDIF
      DXSQ=ONE/DXSQ
      VD=0.D0
      DO 5100 I=1,NVAR
        HD=ZERO
        DO 4800 J=1,I
          KK=KK+1
4800        HD=HD+H(KK)*DX(J)
        IF(I.GE.NVAR) GOTO 5000
          K=KK
          DO 4900 J=I+1,NVAR
            K=K+J-1
4900          HD=HD+H(K)*DX(J)
5000      Y(I)=(DG(I)-HD)*DXSQ
5100      VD=VD+Y(I)*DX(I)
      VD=-VD*DXSQ
      K=0
      DO 5110 I=1,NVAR
      DO 5110 J=1,I
        K=K+1
5110      H(K)=H(K) + Y(I)*DX(J) + DX(I) * (Y(J) + VD*DX(J))
      CALL DSINV(H,NVAR,EPS,IER)
5120 FORMAT(/5X,'** THE HESSIAN CANNOT BE INVERTED BY SUB DSINV ')
5125 FORMAT(' +',54X,'IT WILL BE RESET TO ITS PREVIOUS VALUE **'/)
      IF(IER.EQ.0) THEN
        INVERT=0
      ELSE
        WRITE(IO,5120)
        INVERT=INVERT+1
        IF(INVERT.EQ.1) THEN
          WRITE(IO,5125)
          REWIND 3
          READ(3)(H(I),I=1,NH)
        ELSE
          WRITE(6,5127)
5127 FORMAT(' +',54X,'IT WILL BE INITIALIZED TO A DIAGONAL MATRIX **'/)
          GOTO 1150
        ENDIF
      ENDIF
    ELSE
      SXG=ZERO
      SG=ZERO
      GHG=ZERO
      DO 3900 I=1,NVAR
        Y(I)=ZERO
```

```
      DO 3700 J=1,I
3700      Y(I)=Y(I) + H(KK+J)*DG(J)
      IF(I GE NVAR) GOTO 3850
        K=KK+I
        KK=K
        DO 3750 J=I+1,NVAR
          K=K+J-1
3750        Y(I)=Y(I) + H(K)*DG(J)
3850    SXG=SXG + DX(I)*DG(I)
      SG=SG+DG(I)**2
3900    GHG=GHG + DG(I)*Y(I)
    HAV=GHG/SG
    SXG=ONE/SXG
    KK=0
    IF(METHOD.EQ.1) THEN
C FLETCHER UPDATE
      BETA=(ONE+GHG*SXG)
      DO 4500 I=1,NVAR
        YC=-Y(I)*SXG
        PC=-DX(I)*SXG
        PB=-PC*BETA
        DO 4400 J=1,I
          K=KK+J
4400        H(K)=H(K) + YC*DX(J) + PC*Y(J) + PB*DX(J)
4500      KK=KK+I
    ELSE
C DAVIDON-FLETCHER-POWELL UPDATE
      GHG=-ONE/GHG
      DO 4200 I=1,NVAR
        PC=DX(I)*SXG
        YC=Y(I)*GHG
        DO 4100 J=1,I
          K=KK+J
4100        H(K)=H(K) + PC*DX(J) + YC*Y(J)
4200      KK=KK+I
    ENDIF
  ENDIF
  HTIME=6.D1*CLOCK(I)-HTIME
  WRITE(6,4499) HTIME
4499 FORMAT(10X,'IT TOOK',F9.4,' SECS TO UPDATE THE HESSIAN')
C
 5020 IF(MAX.EQ.0) GOTO 1600
C WRITE CURRENT POINT TO DISK  THIS IS ALSO DONE AT LABEL 1600
  HTIME=6.D1*CLOCK(I)
  CALL PUNOUT
  STIME=6.D1*CLOCK(I)
  SAFETY=SFDFP*TIMAX
  HTIME=6.D1*CLOCK(I)-HTIME
```

```
      WRITE(6,4498) HTIME
4498  FORMAT(' +  52X,  F9 4  SECS FOR SUBR PUNOUT ')
      IF(STIME+SAFETY LE TLIMIT) GOTO 5141
      IF(LINES+15 GT LITOT) CALL PAGE
      WRITE(IO,6150) STIME,TLIMIT
      IJUMP=1
      CALL PUNOUT
      STOP
C
CHECK TO SEE IF QST MAXIMUM IS NEEDED  ALSO CALL IF GZ IS TOO HIGH
 5141 IF(MAX LE 0) GOTO 1605
      IF(IMAX EQ 1  AND  DABS(GZ) LT DMAX1(GVM,GRMS*0 5D0)  AND
        LINMAX LT 2) GOTO 1605
      IF(IMAX EQ 2  AND  LINMAX LT MXCALL  AND  GRMS LT TWO*GRMTOP AND
        DABS(EPC-ETARG) LT 0 03D0) GOTO 1605
      IF(GRMS LT 0.75D0*GRMSL  AND  EPC GT P1  AND  EPC LT P2)GOTO 1605
      IF(GRMS LT 1.5D0*GVM AND LINMAX LT MXCALL+2) GOTO 1605
      LINMAX=0
C     IF(INVERT GT 1) THEN
C       INVERT=1
C       REWIND 3
C       READ(3)(H(I),I=1,NH)
C     ENDIF
      MXCALL=4
      IF(GRMS GT TWO*GRMTOP) MXCALL=3
      ELAST=TOTALE
      CALL PTHMAX(TOTALE,XSAV,G,PTH,EPC,GRMS,GZ,GMAX,XRMS)
      JCYC=JCYC+1
      TARGET=PTH
      ETARG=EPC
      GRMSN=GRMS*SQRTN
      IF(LINES+5 GT LITOT) CALL PAGE
      LINES=LINES+5
      WRITE(IO,5800)
      TIME=6 D1*CLOCK(I)
      WRITE(IO,5920)JCYC,NCNT,TOTALE,GRMS,EPC,PTH,GZ,XRMS,TIME,GRMSN
      TIMAX=(TIME-STIME+TIMAX)*0 5D0
      IF(TOTALE-ELAST GT 4 D0) MXCALL=3
      GRMTOP=GRMS
C
      IF(GRMS GT GVM  OR  GMAX GT GVM2  OR  XRMS GT XVM) GOTO 1600
C     NORMAL EXIT
 5200 IF(LINES+2 GT LITOT) CALL PAGE
      LINES=LINES+2
      WRITE(IO,6000)
      NSUCC=1
C
C   SAVE THE HESSIAN FOR POSSIBLE REUSE IN A RESTART
```

```
C
 5400 CALL PUNOUT
      CALL HDIAG(H,NVAR)
      IWADE=0
C
      NPETER=NCNT
      SCFF=SCFGRD
      RETURN
C
C     MAXEND EXIT
 5700 IF(LINES+5+NATMS GT LITOT) CALL PAGE
      TIMEO=-1.D9
      WRITE(IO,6100)
      NUSCC=11
      GOTO 5400
 5800 FORMAT(/' XRMS KE ENERGY-TIME ALPHA-DZ  GRMS     EPC
     1 '  PATH1.2  GZ/ZCURVE  GD0     GDA   DPEDY/DZ')
 5900 FORMAT(/' $',I3,I4,F11.6,F8.4,F10.4,5F10.5,F10.6,/
     1 ' $',F10.6,F8.3,F8.4,F10.4,F20.5,F10.3,2F10.5,F10.6)
 5910 FORMAT(/4H $Y,I5,F11.6,F8.4,F10.4,5F10.5,F10.6,/
     1 4H $Y,F8.6,F8.3,F8.4,F10.4,F20.5,F10.3,2F10.5,F10.6)
 5920 FORMAT(/' $M',I2,I4,F11.6,F18.4,3F10.5/
     '  $M',F9.6,F8.3,F18.4)
 6000 FORMAT(/20X,'*****  OPTIMIZATION CONVERGED  *****')
 6100 FORMAT(/6X,'MAXIMUM NUMBER OF ITERATIONS REACHED  EXITING')
 6150 FORMAT(/15X,'**** THE TIME LIMIT MIGHT BE EXCEEDED BEFORE '
     1'ANOTHER LINE SEARCH COULD BE COMPLETED ****'//25X,F9.2,
     2' SECONDS HAVE BEEN USED.',F9.2,' SECONDS WERE REQUESTED ')
 6200 FORMAT(//9X,'*****',A8,' VARIABLE METRIC OPTIMIZATION *****'//)
 6210 FORMAT(15X,'FLETCHER UPDATING SCHEME')
6220 FORMAT(6X,'COMPLIMENTARY DAVIDON-FLETCHER-POWELL UPDATING SCHEME')
 6230 FORMAT(15X,'POWELL UPDATING SCHEME')
 6250 FORMAT(20X,'DAVIDON CUBIC INTERPOLATION FOR ONE-DIMENSION SEARCH')
 6600 FORMAT(6X,'CONVERGENCE CRITERIA ARE XRMS  LT  ',F7.4,
     1 ' AND GRMS  LT  ',F7.3/)
 6700 FORMAT(/' $',I7,F11.6,F8.4,F10.4,5F10.5,F10.6,/
     1 ' $',F10.6,F8.3,F8.4,F10.4,F20.5,F10.3,2F10.5,F10.6)
      END
C
C ******************
C
      SUBROUTINE PTHCRD(X,PATH,NVAR)
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
      COMMON
     /PARM1 / A(I3NX),NATMD,NADUM(NUMATM,5),LOC(2,I3NX),NN2(2)
```

```
      /INTVAR/ VF(I3NX) VL(I3NX) Z(I3NX) BF(MPAK) BL(MPAK) DFL
     1     LOCVI(I3NX) MAX
      /SYNCOM/ D1(I3NX 3) W(I3NX) VI(I3NX) S(3) P(4) DRP D2(3)
     1     NVC(I3NX) NV JD(13)
      /ATOMS / C(3 NUMATM) NNDUM IDUM(I3NX)
      /MMSYNC/ NDHA NEWDHA NTRAN NT1 IJKL(2 NUMATM)
      /RHO   / PP(MPAK) B(MPAK-10) AA(5) R0(5)
      DIMENSION X(1)
C
      NBDORD=0
      DO 10 I=1 NVAR
        XI=X(I)
        IF(NVC(LOCVI(I)) GT 0) GOTO 10
          XI=DMAX1(1 D-6 XI)
          NBDORD=NBDORD+1
          XI=R0(NBDORD)+AA(NBDORD)*DLOG10(XI)
   10   VI(LOCVI(I))=XI/S(LOC(2 I))
C
      CALL VDIST(DUM DUM W S VI VF DRM NVC NV 0)
      CALL VDIST(DUM DUM W S VI VL DPM NVC NV 0)
C
      PATH=0 5D0*(P(1)+P(NT1)+(P(NT1)-P(1))*(DRM**2-DPM**2)/DRP**2)
      RETURN
      END
C
C ******************
C
      SUBROUTINE ELECPC(EPC)
      IMPLICIT REAL*8 (A-H O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16 NUMORB=34)
      PARAMETER (I3NX=3*NUMATM MPAK=(NUMORB*(NUMORB+1))/2)
      COMMON
      /INTVAR/ VF(I3NX) VL(I3NX) Z(I3NX) BF(MPAK) BL(MPAK) DFL
     1     LOCVI(I3NX) MAX
      /ATOMS / C(3 NUMATM) NATOMS IDUM(I3NX)
      /MMSYNC/ NDHA NTRAN NT1 IJKL(2 NUMATM)
      /SYNCOM/ DD(I3NX 5) S(3) PCS(8) ID(I3NX) JD(14)
      /ORBITS/ NUMB NORBS CC(NUMORB NUMORB) H(MPAK) F(MPAK)
     1     E1ELN(NUMORB) Q(NUMORB 2)
      /RHO   / P(MPAK) B(MPAK)
C
      CALL OAOPOP(P B F F(NATOMS+1) F(2*NATOMS+1) 1)
C SUM SQUARES OF DIFFERENCES BETWEEN THE OFF-DIAG TERMS OF B ARRAYS
      N=1
      DFC=0 D0
      DLC=DFC
      DO 20 I=2 NATOMS
```

```
      IM1=I-1
      DO 10 J=1,IM1
        K=N+J
        DFC=DFC+(BF(K)-B(K))**2
   10    DLC=DLC+(BL(K)-B(K))**2
   20   N=N+I
C
     EPC=0.5D0*(PCS(1)+PCS(NT1)+(PCS(NT1)-PCS(1))*(DFC-DLC)/DFL)
     RETURN
     END
C
C ******************
C
     SUBROUTINE PTHMAX(ENERGY,XSAV,G,PATH,EPC,GRMS,GZ,GMAX,XRMS)
     IMPLICIT REAL*8 (A-H,O-Z)
C    INCLUDE 'NDFOR.DIM/NOLIST'
     PARAMETER (NUMATM=16,NUMORB=34)
     PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
     COMMON
     /DFP   / X(I3NX),NVAR
     /SYNCOM/ CC(3,NUMATM,3),W(I3NX),VI(I3NX),S(3),P(3),DELP,DRP,
          FRAC,TARG,SC,NVC(I3NX),NV,NN(4),NTYPE,K(8)
     /ATOMS / C(3,NUMATM),NUMAT,KDUM(I3NX)
     /INTVAR/ VF(I3NX),VL(I3NX),Z(I3NX),BF(MPAK),BL(MPAK),DFL,
          LOCVI(I3NX),MAX
     /PARM1 / A(3,NUMATM),NATMD,NI(NUMATM,5),LOC(2,I3NX),LREACT(2)
     /MMSYNC/ NDHA,NEWDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
     /ERG   / EX,GG(I3NX)
     /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
          KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
          ITIME,NTYP,KRESET,IWADE,NTO,NREM,MAXEND
     /ANTDER/ ANALY,JSKPA
     /CYCLES/ JCYC,NPETER
     /RHO   / PP(MPAK),B(MPAK-10),AA(5),R0(5)
     LOGICAL ANALY
     DIMENSION XSAV(1),G(1),NC(4)
     DATA PI/3.1415926536D0/
     TWOPI=PI+PI
C
     CALL PTHCRD(X,PATH,NVAR)
     CALL GETXYZ(C,VI)
     IF(NTRAN.EQ.1) THEN
      DO 10 I=1,3
      DO 10 J=1,NUMAT
        CC(I,J,3)=CC(I,J,2)
   10   CC(I,J,2)=C(I,J)
      P(3)=P(2)
      NTYPE=2
```

```
      NTRAN=2
      NTRN1=3
     ELSE
      DO 20 I=1,3
      DO 20 J=1,NUMAT
  20     CC(I,J,2)=C(I,J)
     ENDIF
C
C INTERPOLATE A STRUCTURE OF THIS PATH COORD  COMPUTE ERROR IN
C PATH COORD OF NEW STRUCTURE TO ADJUST 'AIM' OF INTERPOLATION
     P(2)=PATH
     FRAC=PATH
     TARG=PATH
     CALL STRTP(FRAC,0)
     CALL GETXYZ(C,VI)
     CALL VDIST(C,CC(1,1,1),W,S,VF,VL,DF,NVC,NV,1)
     CALL VDIST(C,CC(1,1,3),W,S,VF,VL,DL,NVC,NV,1)
     PATH=0.5D0*(P(1)+P(3)+(P(3)-P(1))*(DF**2-DL**2)/DRP**2)
     ERR2=TARG-PATH
     ERR =FRAC-PATH
     DFDT=0.D0
     IF(ERR2.NE.0.D0) DFDT=ERR/ERR2
     KNT=0
     FRLST=FRAC
     FRAC=FRAC+ERR
     IF(DABS(ERR).LE.0.002D0) GOTO 40
  30  CALL STRTP(FRAC,0)
      PTH=PATH
      CALL GETXYZ(C,VI)
      CALL VDIST(C,CC(1,1,1),W,S,VF,VL,DF,NVC,NV,1)
      CALL VDIST(C,CC(1,1,3),W,S,VF,VL,DL,NVC,NV,1)
      PATH=0.5D0*(P(1)+P(3)+(P(3)-P(1))*(DF**2-DL**2)/DRP**2)
      ERR2=TARG-PATH
      ERR =FRAC-PATH
      KNT=KNT+1
     IF(DABS(ERR2).LT.DMAX1(0.001D0,0.05D0*DABS(ERR)))GOTO 40
      IF(KNT.GT.6) STOP
      IF(PATH.NE.PTH) DFDT=(DFDT*2.D0+(FRAC-FRLST)/(PATH-PTH))/3.D0
      FRLST=FRAC
      FRAC=FRAC+ERR2*DFDT
      GOTO 30
C INCREMENT SHOULD BE LARGER THAN ERROR, BUT NO LARGER THAN 0.1
  40 DELP=DMIN1(0.1D0,DMAX1(DELP,1.1D0*DABS(ERR)))
     IF(TARG+DELP.GE.P(3)) DELP=P(3)-TARG-1.D-5
     IF(TARG-DELP.LE.P(1)) DELP=TARG-P(1)+1.D-5
     DELP=DMAX1(0.015D0,DELP)
     NBDORD=0
     DO 50 I=1,NVAR
```

```
      XI=X(I)
      XSAV(I)=XI
      LV=LOCVI(I)
      IF(NVC(LV) GT 0) GOTO 50
        NBDORD=NBDORD+1
        XI=DMAX1(1 D 6 XI)
        XI=R0(NBDORD)+AA(NBDORD)*DLOG10(XI)
 50   VI(LV)=XI/S(LOC(2.I))
    CALL GETXYZ(C.VI)
    P(2)=PATH
    DO 65 I=1.3
    DO 65 J=1.NUMAT
 65   CC(I.J.2)=C(I.J)
C
    EX=ENERGY
    IWADE=0
    CALL ENIMAX(4.VF.VL.EPC.1)
    IWADE=1
    DO 60 I=1.3
    DO 60 J=1.NUMAT
 60   CC(I.J.2)=C(I.J)
    ENERGY=EX
    PATH=FRAC
    NBDORD=0
    XRMS=0 D0
    DO 80 I=1.NVAR
      VAR=VI(LOCVI(I))*S(LOC(2.I))
      IF(NVC(LOCVI(I)) GT 0) GOTO 75
        NBDORD=NBDORD+1
        VAR=10.D0**((VAR-R0(NBDORD))/AA(NBDORD))
 75   DX=VAR-XSAV(I)
      IF(LOC(2.I).GT.1) THEN
        IF(DX.GT. PI) DX=DX-TWOPI
        IF(DX.LT.-PI) DX=DX+TWOPI
      ENDIF
      XRMS=XRMS+DX**2
 80   X(I)=VAR
    XRMS=DSQRT(XRMS/DFLOAT(NVAR))
    JSKPA=1
    CALL DXYZ
    CALL DRTP(G)
    CALL ELECPC(EPC)
    CALL PTHCRD(X.P(2).NVAR)
C
C SET INCREMENT FOR NEXT CALL TO AVERAGE OF LAST INCREMENT AND
C CHANGE IN PATH COORD FOR THIS CALL. WITH (MIN. MAX) = (0 02. 0 1)
C
    DELP=0 5D0*(DELP+DABS(P(2)-PATH))
```

```
      PATH=P(2)
      GRMS=0 D0
      GZ=0 D0
      GMAX=0 D0
      DO 90 I=1 NVAR
        GZ=GZ+G(I)*Z(I)
        GMAX=DMAX1(GMAX DABS(G(I)))
        GG(I)=G(I)
   90   GRMS=GRMS+G(I)**2
      GRMS=DSQRT(GRMS/DFLOAT(NVAR))
      RETURN
      END
C
C ******************
C
      SUBROUTINE CPHF(YSAV Y NVAR EPC DEPCDY DEPCDZ)
      IMPLICIT REAL*8 (A-H O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16 NUMORB=34)
      PARAMETER (I3NX=3*NUMATM NDIE=6*NUMORB*NUMATM
     MPAK=(NUMORB*(NUMORB+1))/2 NBDIM=NUMORB**2/4)
      PARAMETER (NUMLMN=NUMORB*MPAK)
      COMMON
     /ERG   / ENERG GTOT(I3NX)
     /UNITS / IN IO JO LINES LITOT
     /BNDATA/ BETAS(18) BETAP(37) ALPA(55) ZS(18) ZP(18)
     /PARM1 / VAR(3 NUMATM) NATMD ND(NUMATM 5) LOC(2 I3NX) LREACT(2)
     /SYNCOM/ DUM(I3NX 5) S(3) PCS(8) NVC(I3NX) JD(14)
     /MMSYNC/ NDHA NEWDHA NTRAN NTRN1 IJKL(2 NUMATM)
     /ATOMS / CORD(I3NX) NATMS NAT(NUMATM) N1ST(NUMATM) NLST(NUMATM)
     /INTVAR/ VF(I3NX) VL(I3NX) Z(I3NX) BF(MPAK) BL(MPAK) DFL
           LOCVI(I3NX) MAX
     /HALFE / IMULT NOCCO NCLOSE IOPEN IODD JODD XIIII XJJJJ XIJIJ
     /REP   / GSS(18) GPP(18) GSP(18) GP2(18) HSP(18) HP2(18)
     /DIELRE/ W(NDIE)
     /ORBITS/ NUMB NORBS C(NUMORB NUMORB) DSY(MPAK) DSZ(MPAK)
           E1ELN(NUMORB) Q(NUMORB 2)
     /ATDATA/ YYDUM(90) CORE(18) ZZDUM(36)
     /RHO   / P(MPAK) B(MPAK-10) AA(5) R0(5)
C    /DDSDDR/ DDS(MPAK*3) DR(NDIE) DRY(NDIE) DRZ(NDIE)
     /DDSDDR/ DDS(MPAK*3) DR(NDIE*3)
     DIMENSION YSAV(1) Y(1) B0(NBDIM*30) ALMNW(NUMLMN)
           AIBNY(NUMORB) AIBNZ(NUMORB) DRY(NDIE) DRZ(NDIE)
             AIMNW(MPAK) AIBNW(NUMORB)
           CY(I3NX) CZ(I3NX)
C     EQUIVALENCE (B0(NBDIM*3+1) ALMNW(1))
C
COMPUTE LENGTH OF Y VECTOR & XYZ COORDS OF NEW POINT
```

```
C
      TIMEI=CLOCK(I)
      NBDORD=0
      YLEN=0 D0
      DO 10 I=1.NVAR
        DY=0 1D0*Y(I)
        YLEN=YLEN+DY**2
        VV=YSAV(I)+DY
        IF(NVC(LOCVI(I)).GT 0) GOTO 10
          NBDORD=NBDORD+1
          VV=DMAX1(1 D-6 VV)
          VV=R0(NBDORD)+AA(NBDORD)*DLOG10(VV)
   10   VAR(LOC(2.I).LOC(1.I))=VV
      CALL GMETRY(CY.-1)     =VV
C MAKE SHIFT ALONG Z OF EQUAL LENGTH TO THE ONE ALONG Y
      DZ=DSQRT(YLEN)
      NBDORD=0
      DO 20 I=1.NVAR
        VV=YSAV(I)+DZ*Z(I)
        IF(NVC(LOCVI(I)).GT.0) GOTO 20
          VV=DMAX1(1 D-6.VV)
          NBDORD=NBDORD+1
          VV=R0(NBDORD)+AA(NBDORD)*DLOG10(VV)
   20   VAR(LOC(2.I).LOC(1.I))=VV
      CALL GMETRY(CZ.-1)
C RESET VAR & XYZ TO THOSE OF STARTING POINT
      NBDORD=0
      DO 30 I=1.NVAR
        VV=YSAV(I)
        IF(NVC(LOCVI(I)) GT.0) GOTO 30
          NBDORD=NBDORD+1
          VV=R0(NBDORD)+AA(NBDORD)*DLOG10(VV)
   30   VAR(LOC(2.I).LOC(1.I))=VV
      CALL GMETRY(CORD.-1)
COMPUTE Y & Z VECTORS OVER XYZ COORDS
      N3=3*NATMS
      DO 40 I=1.N3
        CY(I)=CY(I)-CORD(I)
   40   CZ(I)=CZ(I)-CORD(I)
C
C LOOP OVER ATOMS  COMPUTE ONE CENTER CONTRIBUTIONS TO A  THEN
C LOOP OVER ATOM PAIRS  THE XYZ DERIVATIVES OF THE ONE ELECTRON
C HAMILTONIAN INTEGRALS. AND THE TWO CENTER REPULSION INTEGRALS
C WERE WRITTEN TO UNIT 3 DURING THE ENERGY GRADIENT CALCULATION
C COMPUTE 2 CENTER CONTRIBUTIONS TO A. AND TO B0 FOR Y AND Z VECTORS
C
      IH=0
      IR=0
```

```
      IHX=0
      IJW=0
      ICX=1
      NINT=0
      DO 80 I=2,NATMS
        ICX=ICX+3
        IF=N1ST(I)
        IL=NLST(I)
        IFL=1+IL-IF
        NINT=NINT+IFL*(IFL+1)/2
        IMINUS=I-1
        JCX=-2
      DO 80 J=1,IMINUS
        JCX=JCX+3
        JF=N1ST(J)
        JL=NLST(J)
        JFL=1+JL-JF
        IIH=IFL*JFL
        IH2=IIH+IIH
        NJ=JFL*(JFL+1)/2
        IIR=IFL*(IFL+1)*NJ/2
        IR2=IIR+IIR
C     COMPUTE SUM OVER X Y & Z OF DS AND OF DR TIMES THE Y & Z VECTORS
        DXY=CY(ICX)-CY(JCX)
        DXZ=CZ(ICX)-CZ(JCX)
        DYY=CY(ICX+1)-CY(JCX+1)
        DYZ=CZ(ICX+1)-CZ(JCX+1)
        DZY=CY(ICX+2)-CY(JCX+2)
        DZZ=CZ(ICX+2)-CZ(JCX+2)
        DO 70 K=IF,IL
          LR=K-IL-1
        DO 70 L=IF,K
          LR=LR+IL-L+1
        DO 70 M=JF,JL
          IF(K.EQ.L) THEN
            IH=IH+1
            IHX=IHX+1
      DSY(IH)=-1.D1*(DDS(IHX)*DXY+DDS(IHX+IIH)*DYY+DDS(IHX+IH2)*DZY)
      DSZ(IH)=-1.D1*(DDS(IHX)*DXZ+DDS(IHX+IIH)*DYZ+DDS(IHX+IH2)*DZZ)
          ENDIF
          NR=M-JL
        DO 70 N=JF,M
          NR=NR+JL-N+1
          IR=IR+1
          IRX=IJW+LR*NJ+NR
          DRY(IR)=DR(IRX)*DXY+DR(IRX+IIR)*DYY+DR(IRX+IR2)*DZY
   70     DRZ(IR)=DR(IRX)*DXZ+DR(IRX+IIR)*DYZ+DR(IRX+IR2)*DZZ
        IHX=IHX+IH2
```

```
 80   IJW=IJW+IIR+IR2
   NMAI=(NORBS-NCLOSE)*NCLOSE
   NMAI2=NMAI+NMAI
   MAI=0
   NNORBS=NORBS*(NORBS+1)/2
   INT1=(NLST(1)-N1ST(1)+1)*(NLST(1)-N1ST(1)+2)/2
   KMN0=NINT
   NINT=NINT+INT1
   REWIND 3
   DO 200 MA=NCLOSE+1,NORBS
     DO 90 I=1,NUMLMN
 90    ALMNW(I)=0.D0
     DO 100 I=1,NORBS
       AIBNY(I)=0.D0
100    AIBNZ(I)=0.D0
     EIGA=E1ELN(MA)
     IOL=0
     ISP=0
     KMNS=KMN0
     INTS=INT1
     DO 140 I=2,NATMS
       IF=N1ST(I)
       IL=NLST(I)
       IFL=IL-IF+1
       IW=IFL*(IFL+1)/2
       NII=NAT(I)
       COREI=-CORE(NII)
       MKLS=0
       INTS=INTS+IW
       DO 130 J=1,I-1
         JF=N1ST(J)
         JL=NLST(J)
         JFL=JL-JF+1
         JW=JFL*(JFL+1)/2
         COREJ=-CORE(NAT(J))
         LMNS=KMNS
         KMN=KMNS
         MKLS=MKLS+JFL*(INTS-IW-JW)
       DO 125 K=IF,IL
         CK=C(K,MA)
         KK=K*(K-1)/2
         PKK=P(KK+K)
         LMN=LMNS
         KMNS=KMN
       DO 125 L=IF,K
         CL=C(L,MA)
         LL=L*(L-1)/2
         PKL=2.D0*P(KK+L)
```

```
            MKLS=MKLS+1
            KMN=KMNS
            MKL=MKLS-IW
          DO 125 M=JF,JL
            CM=C(M,MA)
            MM=M*(M-1)/2
            PMM=P(MM+M)
            PKM=-0.5D0*P(MAX0(KK+M,MM+K))
            PLM=-0.5D0*P(MAX0(LL+M,MM+L))
            MKL=MKL+IW
            IF(M.EQ.JF) GOTO 120
            NKL=MKLS-IW
            DO 110 N=JF,M-1
              ISP=ISP+1
              KMN=KMN+1
              LMN=LMN+1
              NKL=NKL+IW
              CN=C(N,MA)
              NW=IW*(N-1)
              NN=N*(N-1)/2
              PMN=2.D0*P(MM+N)
              PKN=-0.5D0*P(MAX0(KK+N,NN+K))
              IF(K.EQ.L) THEN
C (KK MN)
                PKKI=PKK
                IF(K.EQ.IF) PKKI=PKKI+COREI
                COEF=CK*PMN+CM*PKN+CN*PKM
                AIBNY(K)=AIBNY(K)+DRY(ISP)*COEF
                AIBNZ(K)=AIBNZ(K)+DRZ(ISP)*COEF
                COEF=CK*PKN+CN*PKKI
                AIBNY(M)=AIBNY(M)+DRY(ISP)*COEF
                AIBNZ(M)=AIBNZ(M)+DRZ(ISP)*COEF
                COEF=CK*PKM+CM*PKKI
                AIBNY(N)=AIBNY(N)+DRY(ISP)*COEF
                AIBNZ(N)=AIBNZ(N)+DRZ(ISP)*COEF
                ALMNW(KMN)=ALMNW(KMN)+CK*W(ISP)
                ALMNW(NKL)=ALMNW(NKL)+CM*W(ISP)
                ALMNW(MKL)=ALMNW(MKL)+CN*W(ISP)
              ELSE
C (KL MN)
                PLN=-0.5D0*P(MAX0(LL+N,NN+L))
                COEF=CL*PMN+CM*PLN+CN*PLM
                AIBNY(K)=AIBNY(K)+DRY(ISP)*COEF
                AIBNZ(K)=AIBNZ(K)+DRZ(ISP)*COEF
                COEF=CK*PMN+CM*PKN+CN*PKM
                AIBNY(L)=AIBNY(L)+DRY(ISP)*COEF
                AIBNZ(L)=AIBNZ(L)+DRZ(ISP)*COEF
                COEF=CK*PLN+CL*PKN+CN*PKL
```

```
            AIBNY(M)=AIBNY(M)+DRY(ISP)*COEF
            AIBNZ(M)=AIBNZ(M)+DRZ(ISP)*COEF
            COEF=CK*PLM+CL*PKM+CM*PKL
            AIBNY(N)=AIBNY(N)+DRY(ISP)*COEF
            AIBNZ(N)=AIBNZ(N)+DRZ(ISP)*COEF
            ALMNW(KMN)=ALMNW(KMN)+CL*W(ISP)
            ALMNW(LMN)=ALMNW(LMN)+CK*W(ISP)
            ALMNW(MKL)=ALMNW(MKL)+CN*W(ISP)
            ALMNW(NKL)=ALMNW(NKL)+CM*W(ISP)
           ENDIF
 110        CONTINUE
 120      ISP=ISP+1
         KMN=KMN+1
         IF(K EQ L) THEN
C (KK MM)
          PKKI=PKK
          PMMJ=PMM
          IF(K.EQ.IF) PKKI=PKKI+COREI
          IF(M.EQ.JF) PMMJ=PMMJ+COREJ
          IOL=IOL+1
          AIBNY(K)=AIBNY(K)+(CK*PMMJ+CM*PKM)*DRY(ISP)+CM*DSY(IOL)
          AIBNZ(K)=AIBNZ(K)+(CK*PMMJ+CM*PKM)*DRZ(ISP)+CM*DSZ(IOL)
          AIBNY(M)=AIBNY(M)+(CM*PKKI+CK*PKM)*DRY(ISP)+CK*DSY(IOL)
          AIBNZ(M)=AIBNZ(M)+(CM*PKKI+CK*PKM)*DRZ(ISP)+CK*DSZ(IOL)
          ALMNW(KMN)=ALMNW(KMN)+CK*W(ISP)
          ALMNW(MKL)=ALMNW(MKL)+CM*W(ISP)
         ELSE
C (KL MM)
          PMMJ=PMM
         LMN=LMN+1
          IF(M.EQ.JF) PMMJ=PMMJ+COREJ
          COEF=CL*PMMJ+CM*PLM
          AIBNY(K)=AIBNY(K)+DRY(ISP)*COEF
          AIBNZ(K)=AIBNZ(K)+DRZ(ISP)*COEF
          COEF=CK*PMMJ+CM*PKM
          AIBNY(L)=AIBNY(L)+(CK*PMMJ+CM*PKM)*DRY(ISP)
          AIBNZ(L)=AIBNZ(L)+(CK*PMMJ+CM*PKM)*DRZ(ISP)
          AIBNY(M)=AIBNY(M)+(CK*PLM+CL*PKM+CM*PKL)*DRY(ISP)
          AIBNY(M)=AIBNY(M)+(CK*PLM+CL*PKM+CM*PKL)*DRY(ISP)
          AIBNZ(M)=AIBNZ(M)+(CK*PLM+CL*PKM+CM*PKL)*DRZ(ISP)
          ALMNW(KMN)=ALMNW(KMN)+CL*W(ISP)
          ALMNW(LMN)=ALMNW(LMN)+CK*W(ISP)
          ALMNW(MKL)=ALMNW(MKL)+CM*W(ISP)
         ENDIF
 125      CONTINUE
         KMNS=KMN
 130      MKLS=MKL+(NINT-INTS)*JFL
 140     KMNS=KMN+(NINT-INTS)*IFL
```

```
      DO 200 MI=1 NCLOSE
        MAI=MAI+1
        DO 53 MN=1 NNORBS
   53     AIMNW(MN)=0 D0
        AY=0 D0
        AZ=0 D0
        LMN=0
        DO 160 I=1 NATMS
          IF=N1ST(I)
          IL=NLST(I)
          IFF=IF*(IF-1)/2
          NII=NAT(I)
          SSSS=2 D0*GSS(NII)
          IF(IL GT IF) THEN
            PPPP=2 D0*GPP(NII)
            SSPP=4 D0*GSP(NII)-2 D0*HSP(NII)
            SPSP=3 D0*HSP(NII)-GSP(NII)
            PPP2P2=4 D0*GP2(NII)-2 D0*HP2(NII)
            PP2PP2=3 D0*HP2(NII)-GP2(NII)
          ENDIF
        DO 160 J=1 NATMS
          JF=N1ST(J)
          JL=NLST(J)
          JFF=JF*(JF-1)/2
          WKKKK=SSSS
          LL=IFF
        DO 160 L=IF IL
          CLI=C(L MI)
          KL=LL+IF-1
          LL=LL+L
          IF(I EQ J) THEN
            AY=AY+CLI*AIBNY(L)
            AZ=AZ+CLI*AIBNZ(L)
            CLA=C(L MA)
            CLAI=CLA*CLI
            IF(L EQ IF) GOTO 48
            KK=IFF
COMPUTE (SS PP), (SP SP), (PP P*P*), AND (PP* PP*) CONTRIBUTIONS
            WKKLL=SSPP
            WKLKL=SPSP
            DO 54 K=IF L-1
              KL=KL+1
              KK=KK+K
              AIMNW(LL)=AIMNW(LL)+C(K MA)*C(K MI)*WKKLL
              AIMNW(KK)=AIMNW(KK)+CLAI*WKKLL
              AIMNW(KL)=AIMNW(KL)+(C(K MA)*CLI+C(K MI)*CLA)*WKLKL
              WKKLL=PPP2P2
   54         WKLKL=PP2PP2
```

```
COMPUTE (SS SS)=GSS OR (PP PP)=GPP CONTRIBUTION TO A
  48       AIMNW(LL)=AIMNW(LL)+CLAI*WKKKK
           WKKKK=PPPP
           GOTO 160
         ENDIF
         MN=JFF
         LLMN=LL-L
         DO 150 M=JF,JL
           CMI=C(M,MI)
           LM=MAX0(LLMN+M,M*(M-1)/2+L)
           MN=MN+JF-1
         DO 150 N=JF,M
           MN=MN+1
           LMN=LMN+1
           AIMNW(LM)=AIMNW(LM)-C(N,MI)*ALMNW(LMN)
           IF(M EQ N) GOTO 150
           LN=MAX0(LLMN+N,N*(N-1)/2+L)
           AIMNW(LN)=AIMNW(LN)-CMI*ALMNW(LMN)
 150         AIMNW(MN)=AIMNW(MN)+4 D0*CLI*ALMNW(LMN)
 160       CONTINUE
       B0(MAI)=AY/(E1ELN(MI)-EIGA)
       B0(MAI+NMAI)=AZ/(E1ELN(MI)-EIGA)
       MBJ=NMAI2
       DO 195 MB=NCLOSE+1,NORBS
         DO 155 I=2,NORBS
 155       AIBNW(I)=0 D0
         MN=1
         AIBNW(1)=C(1,MB)*AIMNW(1)
         DO 175 M=2,NORBS
           CM=C(M,MB)
           DO 170 N=1,M-1
             MN=MN+1
             CN=C(N,MB)
             AIBNW(M)=AIBNW(M)+CN*AIMNW(MN)
 170           AIBNW(N)=AIBNW(N)+CM*AIMNW(MN)
           MN=MN+1
 175         AIBNW(M)=AIBNW(M)+CM*AIMNW(MN)
       DO 195 MJ=1,NCLOSE
         AW=0 D0
         DO 190 N=1,NORBS
 190       AW=AW+C(N,MJ)*AIBNW(N)
         MBJ=MBJ+1
 195     B0(MBJ)=AW
 200   WRITE(3)(B0(IR),IR=NMAI2+1,MBJ)
C
C
C SOLVE MATRIX EQUATIONS  ( 1 - A ) B = B0 . WHERE
C A  IS THE MATRIX OF ELECTRON REPULSION INTEGRALS.
```

```
C B0 IS THE SUM OF THE DERIVATIVES OF THE REPULSION INTEGRALS AND
C   OF THE ONE ELECTRON HAMILTONIAN INTEGRALS  AND
C B  IS THE OCCUPIED-VIRTUAL ELEMENTS OF U  FOR THE VECTORS Y & Z
C
    CALL SCPHF(B0,B0(NMAI+1),NMAI,NBDIM,KY,KZ)
C
    DEPCDY=0.D0
    DEPCDZ=0.D0
    IJ=0
    DO 796 I=2,NATMS
      IF=N1ST(I)
      IL=NLST(I)
      IJ=IJ+1
      IM1=I-1
    DO 796 J=1,IM1
      JF=N1ST(J)
      JL=NLST(J)
      IJ=IJ+1
      DB=BL(IJ)-BF(IJ)
    DO 796 K=IF,IL
      KK=K*(K-1)/2
    DO 796 L=JF,JL
      PKL=DB*P(MAX0(KK+L,L*(L-1)/2+K))
      MAI=0
      DPY=0.D0
      DPZ=0.D0
      DO 794 MA=NCLOSE+1,NORBS
      DO 794 MI=1,NCLOSE
        MAI=MAI+1
        COEF=C(K,MA)*C(L,MI)+C(K,MI)*C(L,MA)
        DPY=DPY+COEF*B0(MAI)
794       DPZ=DPZ+COEF*B0(MAI+NMAI)
      DEPCDY=DEPCDY+DPY*PKL
796   DEPCDZ=DEPCDZ+DPZ*PKL
    FACT=4.D0*(PCS(NTRN1)-PCS(1))/DFL
    DEPCDY=FACT*DEPCDY
    DEPCDZ=FACT*DEPCDZ
    TIME=6.D1*(CLOCK(I)-TIME1)
    WRITE(6,440) TIME,KY,KZ
440   FORMAT(F9.4,' SECONDS TO SOLVE THE CPHF EQUATIONS ',I4,
     ' ITERATIONS FOR THE Y VECTOR ',I3,' FOR Z ')
    RETURN
    END
C
C ******************
C
    SUBROUTINE SCPHF(Y,Z,M,NBDIM,KY,KZ)
C
```

```
C SOLVES THE MATRIX EQUATIONS  (1 A) (Y Z) = (Y1 Z1)
C WHERE A IS AN M*M MATRIX  Y Y1 Z & Z1 ARE M COMPONENT VECTORS
C Y & Z ARE TO BE DETERMINED  CLOSED SHELL CASES ONLY
C
      IMPLICIT REAL*8(A H O Z)
      PARAMETER (NUMORB=34)
      PARAMETER (MPAK=(NUMORB*(NUMORB+1))/2)
      COMMON
     /HALFE / IMULT,NOCCO,NO,IOPEN,IODD,JODD,XIIII,XJJJJ,XIJIJ
     /ORBITS/ NUMB,NORBS,C(NUMORB,NUMORB),B(MPAK),F(MPAK),
           EV(NUMORB),Q(NUMORB,2)
      DIMENSION Y(1),YLYK(14,14),YSQUAR(15),W(14)
           Z(1),ZLZK(14,14),ZSQUAR(15)
      DATA KMAX/14/
      DATA ZERO,ONE,SMALL,CONV/0.D0,1.D0,1.D-24,5.D-18/
C
      IPRINT=0
      RM=DFLOAT(M)
      M2=M+M
      DO 1 K=1,KMAX
      DO 1 L=1,KMAX
        YLYK(L,K)=ZERO
    1   ZLZK(L,K)=ZERO
C
C NORMALIZE THE INITIAL VECTOR
C
      TEMPY=ZERO
      TEMPZ=ZERO
      DO 10 I=1,M
        TEMPY=TEMPY+Y(I)**2
   10   TEMPZ=TEMPZ+Z(I)**2
C
C GIVE WARNING IF ONE OF THE INITIAL VECTOR IS ZERO
C SET IT TO ONE SO THAT THE ROUTINE DOES NOT ABORT
C
      TESTY=TEMPY/RM
      TESTZ=TEMPZ/RM
      IF(TESTY.LT.SMALL) THEN
       WRITE(6,2003) 'Y'
       TEMPY=ONE
      ENDIF
      IF(TESTZ.LT.SMALL) THEN
       WRITE(6,2003) 'Z'
       TEMPZ=ONE
      ENDIF
      SCALEY=DSQRT(TEMPY)
      SCALEZ=DSQRT(TEMPZ)
      TEMPY=ONE/SCALEY
```

```
      TEMPZ=ONE/SCALEZ
      DO 20 I=1,M
        Y(I)=Y(I)*TEMPY
   20   Z(I)=Z(I)*TEMPZ
      YSQUAR(1)=ONE
      ZSQUAR(1)=ONE
C
C FORM A SEQUENTIALLY ORTHOGONALIZED SET OF VECTORS X(K)
C XSQUAR(K) CONTAINS THE SQUARE MAGNITUDE OF X(K)
C
      ISIZE=2*(KMAX+1)*NBDIM
      NUMA=M**2
      NAFST=ISIZE-NUMA
      K1=0
C
      DO 100 K=1,KMAX
COMPUTE Y(K+1)=A*Y(K)  AND  Z(K+1)=A*Z(K)
        K0=K1
        K1=K1+M2
        MAXA=ISIZE-K1-M2
        IF(MAXA.LT.M) WRITE(6,802)
        IF(MAXA.LT.M) STOP
        IF(K1+M2.GT.NAFST) NAFST=ISIZE-MAXA
        NALST=ISIZE
        NA=ISIZE
        IF(K.GT.1.AND.NUMA.LE.MAXA) NA=0
        IR=0
        REWIND 3
        DO 28 MA=NO+1,NORBS
        DO 28 MI=1,NO
          IF(NA.GE.MAXA) THEN
            NA=0
            IF(M*(M-IR).LE.MAXA) NALST=NAFST+(M-IR)*M
            DO 443 II=NAFST,NALST-1,M
  443         READ(3)(Y(II+IC),IC=1,M)
C           READ(3)(Y(IC),IC=NAFST+1,NALST)
          ENDIF
          TEMPY=0.D0
          TEMPZ=0.D0
          DO 27 IC=1,M
            NA=NA+1
            TEMPY=TEMPY+Y(NAFST+NA)*Y(K0+IC)
   27       TEMPZ=TEMPZ+Y(NAFST+NA)*Z(K0+IC)
          IR=IR+1
          Y(K1+IR)=TEMPY/(EV(MI)-EV(MA))
   28     Z(K1+IR)=TEMPZ/(EV(MI)-EV(MA))
C
C FORM ELEMENTS OF MATRIXES YLYK & ZLZK
```

```
C NOTE THAT YLYK(L,K) REFERS TO (Y(L) A Y(K))
C
      LM=-M2
      DO 40 L=1,K
        LM=LM+M2
        TEMPY=ZERO
        TEMPZ=ZERO
        DO 30 I=1,M
          TEMPY=TEMPY+Y(LM+I)*Y(K1+I)
 30       TEMPZ=TEMPZ+Z(LM+I)*Z(K1+I)
        YLYK(L,K)=TEMPY
 40     ZLZK(L,K)=TEMPZ
      IF(K.GT.1) THEN
       YLYK(K,K-1)=YSQUAR(K)
       ZLZK(K,K-1)=ZSQUAR(K)
      ENDIF
C
C ORTHOGONALIZE Y(K+1) & Z(K+1) TO PREVIOUS Y & Z VECTORS
C
      LM=-M2
      DO 50 L=1,K
        LM=LM+M2
        TEMPY=YLYK(L,K)/YSQUAR(L)
        TEMPZ=ZLZK(L,K)/ZSQUAR(L)
      DO 50 I=1,M
        Y(K1+I)=Y(K1+I)-TEMPY*Y(LM+I)
 50     Z(K1+I)=Z(K1+I)-TEMPZ*Z(LM+I)
C
C FIND SQUARE MAGNITUDE
C
      TEMPY=ZERO
      TEMPZ=ZERO
      DO 70 I=1,M
        TEMPY=TEMPY+Y(K1+I)**2
 70     TEMPZ=TEMPZ+Z(K1+I)**2
      YSQUAR(K+1)=TEMPY
      ZSQUAR(K+1)=TEMPZ
C
C FIND IF THE NEW VECTOR IS ZERO. IF SO, CPHF HAS CONVERGED
C IN ORDER FOR THE ROUTINE NOT TO BLOW UP, SET XSQUAR(K+1) TO ONE
C
      SUMSQY=TEMPY/RM
      SUMSQZ=TEMPZ/RM
C IF NEW X(K+1) IS SUFFICIENTLY SMALL, EXIT LOOP  SAVE VALUE OF K
      IF(SUMSQY.GT.CONV) KY=K
      IF(SUMSQZ.GT.CONV) KZ=K
      IF(KY.LT.K.AND.KZ.LT.K) GOTO 200
 100 CONTINUE
```

```
C ABORT IF CPHF DOESNT CONVERGE
      IF(KY GE KMAX) WRITE(6 2005)  Y  SUMSQY
      IF(KZ GE KMAX) WRITE(6 2005)  Z  SUMSQZ
      STOP
C
C FIND COEFFICIENTS BY SOLVING SIMULTANEOUS EQUATIONS
  200 KY=KY+1
      KZ=KZ+1
      YSQUAR(KY+1)=ONE
      ZSQUAR(KZ+1)=ONE
C
      DO 210 I=1 KY
  210   YLYK(I I)=YLYK(I I)-YSQUAR(I)
      CALL INV(YLYK KY KMAX)
CALL ? MINV(YLYK KY KMAX)
      DO 220 I=1 KY
  220   W(I)=-YLYK(I 1)*SCALEY
C PLACE ANSWER IN FIRST M LOCATIONS OF ARRAY Y
      DO 230 I=1 M
  230   Y(I)=Y(I)*W(1)
      IS=M
      DO 240 J=2 K
        IS=IS+M
      DO 240 I=1 M
        IS=IS+1
  240   Y(I)=Y(I)+W(J)*Y(IS)
      IF(IPRINT GT 0) WRITE(6 2001) (W(I) I=1 KY)
      IF(IPRINT GT 1) WRITE(6 2002) (Y(I) I=1 M)
C
      DO 250 I=1 KZ
  250   ZLZK(I I)=ZLZK(I I)-ZSQUAR(I)
      CALL INV(ZLZK KZ KMAX)
C
      DO 260 I=1 KZ
  260   W(I)=-ZLZK(I 1)*SCALEZ
      DO 270 I=1 M
  270   Z(I)=Z(I)*W(1)
      IS=M
      DO 280 J=2 K
        IS=IS+M
      DO 280 I=1 M
        IS=IS+1
  280   Z(I)=Z(I)+W(J)*Z(IS)
      IF(IPRINT GT 0) WRITE(6 2001) (W(I) I=1 K)
      IF(IPRINT GT 1) WRITE(6 2002) (Z(I) I=1 M)
      RETURN
2001 FORMAT(' COEFF  1P 10E12 4)
2002 FORMAT(' B  12F10 6)
```

```
 2003 FORMAT(' CPHF INPUT A2  VECTOR IS ZERO')
 2005 FORMAT(' CPHF DID NOT CONVERGE FOR A2  CONVERGENCE = 1PE13.5)
  802 FORMAT(// 5X,'***** NOT ENOUGH ROOM FOR THE A MATRIX IN SC
     1 'PHF  INCREASE SIZE OF B0 IN SUBROUTINE CPHF *****')
      END
C
C ******************
C
      SUBROUTINE INV(A,N,KMAX)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C GAUSS-JORDAM ALGORITHM TO INVERT SQUARE MATRIX A
C
      DIMENSION A(KMAX,KMAX),IS(2,14),IAD1(14),IAD2(14),D(14)
      DATA ZERO,ONE,SMALL/0.D0,1.D0,1.D-18/
C
      DO 10 L=1,N
      IS(1,L)=0
   10 IS(2,L)=0
      DO 90 IMA=1,N
        B=ZERO
        DO 30 L=1,N
        DO 30 M=1,N
          IF(IS(1,L).EQ.1.OR.IS(2,M).EQ.1) GOTO 30
          E=DABS(A(L,M))
          IF(E.LT.B) GOTO 20
          I=L
          K=M
   20     B=DMAX1(B,E)
   30   CONTINUE
        IS(1,I)=1
        IS(2,K)=1
        IAD1(K)=I
        IAD2(I)=K
        B=A(I,K)
C PIVOT
        IF(DABS(B).GT.SMALL) GOTO 40
        WRITE(6,160) B
        STOP
C
   40   A(I,K)=ONE/B
        DO 50 L=1,N
   50   IF(L.NE.K) A(I,L)=-A(I,L)/B
        DO 70 L=1,N
          IF(L.EQ.I) GOTO 70
          DO 60 M=1,N
   60     IF(M.NE.K) A(L,M)=A(L,M)+A(L,K)*A(I,M)
   70   CONTINUE
```

```
      DO 80 L=1,N
   80   IF(L.NE.I) A(L,K)=A(L,K)/B
   90 CONTINUE
C
      DO 120 L=1,N
        DO 100 J=1,N
  100   D(J)=A(IAD1(J),L)
        DO 110 J=1,N
  110   A(J,L)=D(J)
  120 CONTINUE
C
      DO 150 L=1,N
        DO 130 J=1,N
  130   D(J)=A(L,IAD2(J))
        DO 140 J=1,N
  140   A(L,J)=D(J)
  150 CONTINUE
      RETURN
  160 FORMAT(' MATRIX IS SINGULAR IN SUB INV. PIVOT =',1PD12.4)
      END
C
C ******************
C
      SUBROUTINE AX(A,B0,B,NBDIM)
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR.DIM/NOLIST'
      PARAMETER (NUMORB=34)
      PARAMETER (MPAK=(NUMORB*(NUMORB+1))/2)
      COMMON
     /HALFE / IMULT,NOCCO,NO,IOPEN,IODD,JODD,XIIII,XJJJJ,XIJIJ
     /ORBITS/ NUMB,NORBS,C(NUMORB,NUMORB),BF(MPAK,2),
              EV(NUMORB),Q(NUMORB,2)
      DIMENSION A(NBDIM,NBDIM),B0(NBDIM),B(NBDIM)
C
COMPUTE MATRIX PRODUCT  B = A B0
C
      NV=NORBS-NO
      NONV=NO*NV
      DO 20 IR=1,NONV
        TEMP=0.D0
        DO 10 IC=1,NONV
   10   TEMP=TEMP+A(IR,IC)*B0(IC)
   20 B(IR)=TEMP
C
C DIVIDE B BY DIFFERENCE IN EIGENVALUES
C
C     NB=0
C     DO 60 II=NO+1,NORBS
```

```
C       EI=EV(II)
C     DO 60 IA=1 NO
C       NB=NB+1
C  60 B(NB)=B(NB)/(EI-EV(IA))
C
      RETURN
      END
C
C ==================================
C
      SUBROUTINE BONDS
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFORDIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
C
C  ROUTINE TO COMPUTE & STORE THE BONDS IN THE INPUT STRUCTURES
C
      INTEGER*2 IAT1,IAT2
      COMMON/UNITS/IN,IO,JO,LINES,LITOT
     /SYNCOM/ COORD(3,NUMATM,3),W(I3NX),VI(I3NX),S(3),P(3),DELP,DRP,
           FR,TARGET,SCALE,NVC(I3NX),NV,NR,NT,NP,NCALC,NTYPE,
           IOPT,IWRITE,NDENSE,INVAR,IMPROS,NRWT,MODIFY,NATMS
     /MMSYNC/ NDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
     /BNDTAB/ NBNDS,IAT1(NUMATM),IAT2(NUMATM)
     /ORBITS/ NUMB,NN,COEF(NUMORB,NUMORB),H(MPAK),F(MPAK),
           E1ELN(NUMATM),Q(NUMORB,2)
     /INTVAR/ VFLZ(I3NX,3),BF(MPAK),BL(MPAK),DFL,LV(I3NX),MAX2
     /FLAGS / SCFCRT,RFLAGS(13),IFLAGS(16)
      INTEGER*2 NBOR(NUMATM,NUMATM),IAT1S(NUMATM),IAT2S(NUMATM)
      REAL BNDRD(NUMATM),BNDRDS(NUMATM)
      EQUIVALENCE (IAT1S(1),NBOR(1,1)),(IAT2S(1),NBOR(1,2))
C
C FIND BONDING IN FIRST STRUCTURE AND SAVE IN ARRAYS
C INCREASE CONVERGENCE CRITERION FOR PATH DEFINING STRUCTURES
C
      SCFCRT=0.1D0*SCFCRT
      MOL=1
      CALL CONNEC(COORD(1,1,MOL),BF,BNDRD,MOL)
      NBONDS=NBNDS
      DO 55 IL=1,NBNDS
        BNDRDS(IL)=BNDRD(IL)
        IAT1S(IL)=IAT1(IL)
   55   IAT2S(IL)=IAT2(IL)
C
C NOW FIND BONDING IN OTHER STRUCTURES AND COMPARE LISTS
C
      DO 130 MOL=2,NTRN1
```

```
      CALL CONNEC(COORD(1 1 MOL) BL BNDRD MOL)
      IF(NBONDS GE NATMS-1) GOTO 130
CHECK FOR SIMILAR BONDING
      DO 125 IL=1 NBNDS
    DO 110 IS=1 NBONDS
    IF(IAT1(IL) EQ IAT1S(IS) AND IAT2(IL) EQ IAT2S(IS))GOTO 125
    IF(IAT1(IL) EQ IAT2S(IS) AND IAT2(IL) EQ IAT1S(IS))GOTO 125
 110 CONTINUE
C
C IF REACHED HERE  NEW BOND IS NOT ON OLD LIST  ADD TO END
C
      NBONDS=NBONDS+1
      IF(NBONDS LE NUMATM)GOTO 120
    WRITE(IO.92) NUMATM
  92 FORMAT(' ERROR -- MORE THAN ' I4 ' BONDS IN COMPOSITE SET ')
    STOP
C
C BOND LIST MUST BE SORTED BY BOND ORDER
C 'BUBBLE' NEW BOND TO CORRECT POSITION
C
 120   NBND=NBONDS
      DO 124 I=2.NBONDS
        IF(BNDRDS(NBND-1) LT BNDRD(IL)) GOTO 122
        BNDRDS(NBND)=BNDRD(IL)
        IAT1S(NBND)=IAT1(IL)
        IAT2S(NBND)=IAT2(IL)
        GOTO 125
 122      BNDRDS(NBND)=BNDRDS(NBND-1)
        IAT1S(NBND)=IAT1S(NBND-1)
        IAT2S(NBND)=IAT2S(NBND-1)
 124      NBND=NBND-1
C IF LOOP FINISHES. NEW BOND HAD HIGHEST ORDER  PUT IN 1ST POSITION
      BNDRDS(1)=BNDRD(IL)
      IAT1S(1)=IAT1(IL)
      IAT2S(1)=IAT2(IL)
C
 125   CONTINUE
 130 CONTINUE
    SCFCRT=10.D0*SCFCRT
C
C SAVE COMPOSITE LIST IN IAT1  IAT2 ARRAYS & COMPUTE DFL
C
    NBNDS=NBONDS
    DFL=0.D0
    DO 140 IL=1.NBNDS
      IAT1(IL)=IAT1S(IL)
      IAT2(IL)=IAT2S(IL)
 140   DFL=DFL+(BF(IL)-BL(IL))**2
```

```
      LAST=NATMS*(NATMS+1)/2
      K=NBNDS+1
      DO 145 I=K LAST
  145   DFL=DFL+(BF(I) BL(I))**2
C
CHECK TO SEE IF STRUCTURE IS FULLY LINKED  FIRST ZERO OUT TABLE
C
      DO 200 NA=2 NATMS
      DO 200 NB=1 NATMS
        NBOR(NA NB)=0
  200   NBOR(NB NA)=0
C
C  NOW MAKE NOTE OF BONDS ALREADY DECLARED
C
      DO 210 N=1 NBNDS
        NA=IAT1(N)
        NB=IAT2(N)
        NBOR(NA NB)=1
  210   NBOR(NB NA)=1
C
C  TEST TO SEE IF BONDS FORM A LINKED STRUCTURE
C
      LOOP=0
  220 LINK=0
      LOOP=LOOP+1
C
      DO 250 NA=2 NATMS
        NAM1=NA-1
      DO 250 NB=1 NAM1
        IF(NBOR(NA NB) EQ 1) GOTO 240
C
        DO 230 NC=1 NATMS
C IF A IS BONDED TO C AND B IS BONDED TO C  A AND B ARE CONNECTED
          IF(NBOR(NA NC) GT 0 AND NBOR(NB NC) GT 0) NBOR(NA NB)=2
  230     NBOR(NB NA)=NBOR(NA NB)
        GOTO 250
C
  240   DO 245 NC=1 NATMS
          IF(NBOR(NA NC) NE 1  OR  NBOR(NB NC) NE 1) GOTO 245
C
CHECK TO SEE WHETHER BOND LENGTH A-B SHOULD BE KEPT IN SET  SINCE
C-A AND C-B ARE ALSO CONSIDERED AS BONDED
C
      RAB=DSQRT((COORD(1 NA NTRN1)-COORD(1 NB NTRN1))**2+(COORD(2
     1 NA NTRN1)-COORD(2 NB NTRN1))**2+(COORD(3 NA NTRN1)-COORD(3
     2 N3 NTRN1))**2)
      RAC=DSQRT((COORD(1 NA NTRN1)-COORD(1 NC NTRN1))**2+(COORD(2
     1 NA NTRN1)-COORD(2 NC NTRN1))**2+(COORD(3 NA NTRN1)-COORD(3
```

```
     2 NC,NTRN1))**2)
      RBC=DSQRT((COORD(1,NB,NTRN1)-COORD(1,NC,NTRN1))**2+(COORD(2
     1 ,NB,NTRN1)-COORD(2,NC,NTRN1))**2+(COORD(3,NB,NTRN1)-COORD(3,
     2 ,NC,NTRN1))**2)
C
          IF(RAB.LE.1.25D0*DMAX1(RAC,RBC)) GOTO 245
C
C  NO - DELETE THE A-B BOND LENGTH FROM THE SET
          NBOR(NA,NB)=2
          NBOR(NB,NA)=2
  245   CONTINUE
C
  250   IF(NBOR(NA,NB).LE.0) LINK=LINK+1
C
C  IF LINK IS ZERO, ALL ATOMS ARE CONNECTED TO ALL OTHER ATOMS
C  THROUGH AT LEAST ONE DIRECTLY BONDED PATHWAY
C
      IF(LINK.EQ.0) RETURN
      IF(LINK.EQ.LAST) GOTO 260
      LAST=LINK
      GOTO 220
C  THE STRUCTURE IS NOT LINKED  DECLARE ADDITIONAL ATOM PAIRS
C  AS BONDED
  260 RMIN=1.D06
      NAB=0
      DO 270 NA=2,NATMS
        NAM1=NA-1
      DO 270 NB=1,NAM1
        IF(NBOR(NA,NB).NE.0) GOTO 270
        NAB=NAB+1
      RAB=DSQRT((COORD(1,NA,NTRN1)-COORD(1,NB,NTRN1))**2+(COORD(2,
     1 NA,NTRN1)-COORD(2,NB,NTRN1))**2+(COORD(3,NA,NTRN1)-COORD(3,
     2 NB,NTRN1))**2)
        RMIN=DMIN1(RAB,RMIN)
  270 CONTINUE
      IF(NAB.GE.1) GOTO 290
      WRITE(IO,285)
  285 FORMAT(///5X,'**** STRUCTURE CANNOT BE LINKED  RUN ABORTED ****')
      STOP
C
  290 RMIN=RMIN+5.D-2*SCALE
      DO 300 NA=2,NATMS
        NAM1=NA-1
      DO 300 NB=1,NAM1
        IF(NBOR(NA,NB).NE.0) GOTO 300
      RAB=DSQRT((COORD(1,NA,NTRN1)-COORD(1,NB,NTRN1))**2+(COORD(2,
     1 NA,NTRN1)-COORD(2,NB,NTRN1))**2+(COORD(3,NA,NTRN1)-COORD(3,
     2 NB,NTRN1))**2)
```

```
      IF(RAB GT RMIN) GOTO 300
      NBOR(NA,NB)=1
      NBOR(NB,NA)=1
      NBNDS=NBNDS+1
      IAT1(NBNDS)=NA
      IAT2(NBNDS)=NB
      WRITE(IO,195)NA,NB
  195 FORMAT(/,5X,'THE BOND BETWEEN ATOMS ',I3,' AND ',I3
     1,' HAS BEEN ADDED TO LINK THE STRUCTURE ')
  300 CONTINUE
C   NOW GO BACK TO SEE WHETHER STRUCTURE IS NOW LINKED
      GOTO 220
      END
C
C   ****************
C
      SUBROUTINE CONNEC(CRD,BCRD,BNDRD,MOL)
      IMPLICIT REAL*8 (A-H,O-Z)
C     INCLUDE 'NDFOR.DIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
      INTEGER*2 IAT1,IAT2
      COMMON/UNITS/IN,IO,JO,LITOT(2)
     /SYNCOM/ DUM1(I3NX,4),VI(I3NX),DM2(10),SCALE,NVC(I3NX),IDUM1(14)
     /FCOMN / CORE(NUMATM),WDS(NUMATM),NORBS(NUMATM)
     /BNDTAB/ NBNDS,IAT1(NUMATM),IAT2(NUMATM)
     /RHO   / P(MPAK),B(MPAK)
     /ORBITS/ NUMB,NNORBS,COEF(NUMORB,NUMORB),H(MPAK),F(MPAK),
           E1ELN(NUMATM),Q(NUMORB,2)
     /ENERGS/ E(3)
     /FORCE1/ ADUM(20),ENG(I3NX),BDUM(I3NX),NPTS
     /FUNCT / ENSCF,DUM(3)
     /ATOMS / C(3,NUMATM),NUMAT,NAT(NUMATM),NFIRST(NUMATM),
           NLAST(NUMATM)
     /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
           KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
           ITIME,NTYP,KRESET,IWADE,NTO,NREM,MAXEND
      REAL BNDRD(1)
      DIMENSION CRD(3,1),BCRD(1),LEGAL(12)
C
C THIS IS A LIST OF THE ATOM TYPES WHICH THIS ROUTINE WILL BOND
C
      DATA LEGAL/1,5,6,7,8,9,14,15,16,17,35,53/
C
C TEST FOR ILLEGAL ATOM TYPES, AND COPY XYZ FOR ENERGY CALC
C
      SC=1.D0/SCALE
      DO 20 I=1,NUMAT
```

```
      C(1,I)=CRD(1,I)*SC
      C(2,I)=CRD(2,I)*SC
      C(3,I)=CRD(3,I)*SC
      LAT=MOD(NORBS(I),100)
      DO 10 J=1,12
   10    IF(LAT.EQ.LEGAL(J))GOTO 20
      WRITE(IO,30)I,NAT(I)
   20  CONTINUE
   30 FORMAT(' BONDS CANNOT BE FOUND TO ATOM ',I2,' ATOMIC NUMBER ',I4)
C COMPUTE ENERGY & STORE IN ARRAY FOR PRINTING IN SUB SYNCAL. THEN
C COMPUTE APS BOND ORDERS AND STORE BONDED ATOM PAIRS
      CALL AXIS
      CALL GUESSP(P,KHARGE,NNORBS)
      IWADE=1
      CALL FMTRX
      IWADE=0
      E(MOL)=ENSCF
      I3N=3*NUMAT
      GRMS=0.D0
      DO 38 I=1,I3N
   38   GRMS=GRMS+ENG(I)**2
      VI(MOL)=DSQRT(GRMS/I3N)*23.0606D0
      CALL OAOPOP(P,B,F,F(NUMAT+1),F(2*NUMAT+1),1)
C ZERO OUT DIAGONAL TERMS IN BOND INDICE ARRAY
      N=0
      DO 45 I=1,NUMAT
        N=N+I
   45   B(N)=0.D0
      WRITE(6,41)MOL
   41 FORMAT(/5X,'BOND ORDER MATRIX FOR STRUCTURE ',I2)
      CALL VECPRT(B,NUMAT)
C COPY BOND INDICE ARRAY
      LAST=NUMAT*(NUMAT+1)/2
      DO 43 I=1,LAST
   43   BCRD(I)=B(I)
C
      KNT=0
C FIND THE NUMAT LARGEST BOND INDICES
      DO 70 K=2,NUMAT
        BOND=0.D0
        DO 50 I=2,LAST
          IF(B(I).LE.BOND) GOTO 50
            BOND=B(I)
            IBND=I
   50     CONTINUE
        IF(BOND.LT.0.2D0) GOTO 80
C RESET BOND INDICE TO ZERO  SO IT WONT BE FOUND AGAIN
         B(IBND)=0.D0
```

```
C FIND WHICH ATOMS FORM BOND
      N=1
      DO 60 I=2,NUMAT
        N=N+I
        IF(N.LT.IBND) GOTO 60
          KNT=KNT+1
          IAT1(KNT)=I
          IAT2(KNT)=IBND-N+I
          BNDRD(KNT)=BOND
          GOTO 70
 60     CONTINUE
 70   CONTINUE
C
 80 NBNDS=KNT
    RETURN
    END
C
C  ****************
C
    SUBROUTINE FILNVC
    IMPLICIT REAL*8 (A-H,O-Z)
C    INCLUDE 'NDFOR.DIM/NOLIST'
    PARAMETER (NUMATM=16,NUMORB=34)
    PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
    INTEGER*2 IJKL
    COMMON/UNITS/IN,IO,JO,LINES,LITOT
    /MMSYNC/ NDHA,NEWDHA,NTRAN,NTRN1,IJKL(4,NUMATM)
    /PARM1 / A(I3NX),NATMD,NI(NUMATM),NA(NUMATM),NB(NUMATM),
         NC(NUMATM),NN(NUMATM),LOC(2,I3NX),NNN(2)
    /SYNCOM/ C(3,NUMATM,3),W(I3NX),D(I3NX),DDUM(11),NVC(I3NX),NV,NR,
         NT,NP,NCALC,NTYPE,JDUM(8)
    /INTVAR/ VF(I3NX),VL(I3NX),Z(I3NX),BF(MPAK),BL(MPAK),DFL,
         LOCVI(I3NX),MAX2
    /DFP   / X(I3NX),NVAR
    /RHO   / P(MPAK),B(MPAK-10),AA(5),R0(5)
    CHARACTER*8 BREAK
C
C THE IJKL ARRAY IS THE SAME AS NI,A,B,C  I000 IA00 IAB0 IABC
C ADD ALL VARIABLES TO NVC ARRAY TO DEFINE VI ARRAY
C ADD COMPLETE SET TO NI,NA,NB,NC ARRAYS TO DEFINE THE A ARRAY
C FILL LOC   TO GET FROM OPTIMIZATION SET TO A  ARRAY
C FILL LOCVI TO GET FROM OPTIMIZATION SET TO VI ARRAY
C IF BOND IS FORMING, SET NVC=-NVC AS FLAG TO OPTIMIZE BOND ORDER
C
C TRANSFER THE BONDS
C
    NR=0
    NT=NDHA-1
```

```
NP=NT+NDHA-2
NVAR=0
NATM1=NUMATM+1
NBDORD=0
II=IJKL(1,1)
NI(1)=II
NA(II)=0
NB(II)=0
NC(II)=0
L=NTRN1
DO 20 I=2,NDHA
  NR=NR+1
  W(NR)=1 D0
  II=IJKL(1,I)
  IA=IJKL(2,I)
  IB=IJKL(3,I)
  IC=IJKL(4,I)
  NI(I)=II
  NA(II)=IA
  NB(II)=IB
  NC(II)=IC
  NVC(NR)=II+IA*NATM1
  K=MAX0(II*(II-1)/2+IA,IA*(IA-1)/2+II)
  ITEST=0
  BTEST=DMIN1(BF(K),BL(K))
  IF(BTEST LT 0 02D0) ITEST=1
  DB=DABS(BL(K)-BF(K))
  IF((BF(K) GT 0 5D0 AND BL(K) GT 0 5D0)  OR  DB LT 0 2D0
     OR  NTYPE LT 0) GOTO 10
   RF=DSQRT((C(1,II,1)-C(1,IA,1))**2+(C(2,II,1)-C(2,IA,1))**2+
         (C(3,II,1)-C(3,IA,1))**2)
   RL=DSQRT((C(1,II,L)-C(1,IA,L))**2+(C(2,II,L)-C(2,IA,L))**2+
         (C(3,II,L)-C(3,IA,L))**2)
   IF(BF(K) GT BL(K)) THEN
     BMAX=BF(K)
     BMIN=BL(K)
     BREAK= BREAKING
     DR=RF-RL
     RR=RF
   ELSE
     BMAX=BL(K)
     BMIN=BF(K)
     BREAK= FORMING
     DR=RL-RF
     RR=RL
   ENDIF
   IF(LINES+4 GT LITOT) CALL PAGE
   IF(ITEST NE 0  AND  LINES+7 GT LITOT) CALL PAGE
```

```
      LINES=LINES+4
      WRITE(IO,80) II,IA BREAK
      IF(ITEST.NE.0) THEN
        LINES=LINES+3
        WRITE(IO,90) BTEST
      ENDIF
      WRITE(IO,100) BF(K),BL(K)
      NVC(NR)=-NVC(NR)
      NBDORD=NBDORD+1
      AA(NBDORD)=DR/DLOG10(BMAX/BMIN)
      R0(NBDORD)=RR-AA(NBDORD)*DLOG10(BMAX)
   10 NVAR=NVAR+1
      LOC(1,NVAR)=II
      LOC(2,NVAR)=1
      LOCVI(NVAR)=NR
      IF(I.LT.3) GOTO 20
        NT=NT+1
        W(NT)=1.D0
        NVAR=NVAR+1
        LOC(1,NVAR)=II
        LOC(2,NVAR)=2
        LOCVI(NVAR)=NT
        NVC(NT)=II+(IA+IB*NATM1)*NATM1
      IF(I.LT.4) GOTO 20
        NP=NP+1
        W(NP)=1.D0
        NVAR=NVAR+1
        LOC(1,NVAR)=II
        LOC(2,NVAR)=3
        LOCVI(NVAR)=NP
        NVC(NP)=II+(IA+(IB+IC*NATM1)*NATM1)*NATM1
   20 CONTINUE
      NV=NP
      NP=NP-NT
      NT=NT-NR
      NTYPE=IABS(NTYPE)
      RETURN
   80 FORMAT(//,10X,'THE BOND BETWEEN ATOMS ',I3,' AND ',I3,' IS ',A8)
  100 FORMAT(10X,'THE BOND ORDER WILL BE FOLLOWED DURING THE '
     *,'OPTIMIZATION AND PATH MAXIMUM SEARCHES. LIMITING VALUES ARE '
     *,F8.5,' AND ',F8.5)
   90 FORMAT(/,4X,'***** THE BOND ORDER GOES TO ',F9.5,' BETTER RESU'
     *,'LTS MIGHT BE OBTAINED IF THE BOND IS MORE FULLY FORMED *****',/)
      END
C
C ************************
C
      SUBROUTINE INPXYZ
```

```
      IMPLICIT REAL*8 (A-H O Z)
C     INCLUDE 'NDFOR.DIM/NOLIST'
      PARAMETER (NUMATM=16)
      PARAMETER (I3NX=3*NUMATM)
C
C  ROUTINE TO READ INITIAL ATOMIC COORDINATES
C
      COMMON/UNITS/IN IO JO LINES,LITOT
      /FCOMN / CORE(NUMATM),WDS(NUMATM),NORBS(NUMATM)
      /SYNCOM/ COORD(3,NUMATM,3),W(I3NX),VI(I3NX),S(3),P(3),DELP,DRP,
           FR,TARGET,SCALE,NVC(I3NX),NV,NR,NT,NP,NCALC,NTYPE,
           MAX,IWRITE,NDENSE,INVAR,IMPROS,NRWT,MODIFY,NATMS
      /MMSYNC/ NDHA,NTRAN,NTRN1,IJKL(4,NUMATM)
      /TITLE / MTITLE(20)
      /ATOMS / CORDS(3,NUMATM),NUMAT,NAT(NUMATM),NFIRST(NUMATM),
           NLAST(NUMATM)
      /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
           KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
           ITIME,NTYP,KRESET,IWADE,NTO,NREM,MAXEND
      /PARM1 / A(3,NUMATM),NATMD,NI(NUMATM),NC(NUMATM),NB(NUMATM),
           NA(NUMATM),NN(NUMATM),LOC(2,I3NX),LREACT(2)
      INTEGER*2 IJKL
C
C  READ TITLE
C
      READ(IN,10,END=90)(MTITLE(I),I=1,20)
      WRITE(JO,10)(MTITLE(I),I=1,20)
   10 FORMAT(20A4)
C
C  INITIALIZATION CALL TO SUBROUTINE PAGE
C
      CALL PAGE
      IF(MIDDLE.EQ.1) NATMS=NUMAT
      IF(MIDDLE.EQ.1) RETURN
      READ(IN,20)NATMS,NDIS,IMPROS
      WRITE(JO,20)NATMS,NDIS,IMPROS
   20 FORMAT(3I5)
      IF(IABS(NATMS).GT.NUMATM)THEN
       IF(LINES+3.GT.LITOT)CALL PAGE
       WRITE(IO,30) NUMATM,NATMS
   30  FORMAT(// 5X,'*** ERROR ***  ONLY ',I3,' ATOMS ALLOWED ',I10)
       STOP
      ENDIF
      IF(NATMS.LT.0)THEN
       NATMS=-NATMS
       READ(5,40)((IJKL(I,J),I=1,4),J=1,NATMS)
      WRITE(7,40)((IJKL(I,J),I=1,4),J=1,NATMS)
   40  FORMAT(4I2)
```

```
      NDHA=NATMS
      ENDIF
C
      SCALE=1.D0
      IF(NDIS.NE.0) SCALE=1.D0/0.529167D0
      S(1)=SCALE
      S(2)=1.745329252D-2
      S(3)=S(2)
      NTRN1=NTRAN+1
      NTYPE=NTRAN
      NUMAT=NATMS
      NATMD=NATMS
C
C  READ STRUCTURE
C
      READ(IN,60,END=90)(NORBS(N),CORE(N),(COORD(I,N,1),
     1 I=1,3),WDS(N),N=1,NATMS)
      WRITE(JO,60)(NORBS(N),CORE(N),(COORD(I,N,1),I=1,3),
     1 WDS(N),N=1,NATMS)
      DO 50 I=1,NATMS
      NN(I)=CORE(I)
   50 NAT(I)=NN(I)
      IFLG=0
      DO 70 MOL=2,NTRN1
      DO 70 N=1,NATMS
      READ(IN,60,END=90) NORB,COR,(COORD(I,N,MOL),I=1,3)
      WRITE(JO,60) NORB,COR,(COORD(I,N,MOL),I=1,3),WDS(N)
   60 FORMAT(2X,I3,F6.1,4X,3F10.5,5X,A4)
      IF(NORB.NE.NORBS(N).OR.COR.NE.CORE(N)) IFLG=1
   70 CONTINUE
      IF(IFLG.EQ.0)RETURN
C
      WRITE(IO,80)
   80 FORMAT(//,5X,'*** FATAL ERROR ***  --- CORE CHARGES',
     1 ' OR NUMBER OF ORBITALS DO NOT MATCH IN THE STRUCTURES')
      STOP
C
C  END OF FILE FOUND
C
   90 WRITE(IO,100)
  100 FORMAT(' ERROR --- END OF FILE --- MISSING INFORMATION',/,
     1 10X,'*** FATAL ERROR ***')
      STOP
      END
C
C  **********
C
      SUBROUTINE SYNCAL(C)
```

```
      IMPLICIT REAL*8 (A-H O-Z)
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16 NUMORB=34)
      PARAMETER (I3NX=3*NUMATM MPAK= (NUMORB*(NUMORB+1))/2)
C
C THIS IS THE MAIN DRIVER ROUTINE IN SYNCHRONOUS TRANSIT CALCULATION
C
      COMMON/UNITS/IN IO JO LINES LITOT
      /FCOMN / CORE(NUMATM) WDS(NUMATM) NORBS(NUMATM)
      /SYNCOM/ COORD(3 NUMATM 3) W(I3NX) VI(I3NX) S(3) P(3) DELP DRP
            FR TARGET SCALE NVC(I3NX) NV NR NT NP NCALC NTYPE
            MAX IWRITE NDENSE INVAR IMPROS NRWT MODIFY NATMS
      /MMSYNC/ NDHA NTRAN NTRN1 IJKL(2 NUMATM)
      /FUNCT / ENSCF DUM(3)
      /INTVAR/ VF(I3NX) VL(I3NX) Z(I3NX) BFL(MPAK 2) DFL
            LV(I3NX) MAX2
      /ENERGS/ E(3)
      REAL*8 F(3 4) DISCO(3) DSOS(3) C(3 NUMATM)
      REAL DIST ATU ANGS
      DATA ATU ANGS/4H AU  4HANGS /
C
      DIST=ATU
      IF(SCALE EQ 1 D0) DIST=ANGS
      IF(NCALC GT 1) GOTO 320
C  DETERMINE INTERNAL COORDINATE SET AND ASSIGN WEIGHTS
      CALL BONDS
      DO 15 N=1 NATMS
      JL=0
      DO 15 L=2 NTRN1
      JMAX=L-1
      DO 15  J=1 JMAX
      JL=JL+1
      IF(N EQ 1) DSOS(JL)=0 D0
      DO 151  M=1 3
  151 DSOS(JL)=DSOS(JL)+(COORD(M N L)-COORD(M N J))**2
   15 DISCO(JL)=DSQRT(DSOS(JL)/DFLOAT(NATMS))
      IF(LINES+3 GT LITOT) CALL PAGE
      LINES=LINES+3
      IF(NTRAN EQ 1) WRITE(IO 17) DISCO(1) DIST
   17 FORMAT(//10X 'INITIAL CARTESIAN COORDINATE INDEX = ' F12 5 1X A4)
      IF(NTRAN EQ 2) WRITE(IO 18) DISCO DIST
   18 FORMAT(//10X 43HINITIAL CARTESIAN COORDINATE INDICES  1 2 = F12 5
     1 4X 5H1 3 = F12 5 4X 5H2 3 = F12 5 1X A4)
C PRINT ENERGIES AND GRADIENTS WHICH WERE CALCULATED IN SUB CONNEC
C WHEN APS BOND ORDERS WERE DETERMINED
      IF(LINES+4 GT LITOT)CALL PAGE
      LINES=LINES+4
      WRITE(IO 97)(E(M) M=1 NTRN1)
```

```
      WRITE(IO,98)(VI(M) M=1 NTRN1)
   97 FORMAT(/' HEATS OF FORMATION  2(F16 5  KCALS/MOLE  11X)
     1 F16 5  KCALS/MOLE )
   98 FORMAT(/' GRADIENTS  F25 5  KCALS/XYZ  2(F28 5  KCALS/XYZ ))
      IF(LINES+7 GT LITOT) CALL PAGE
      LINES=LINES+7
      WRITE(IO,100)
  100 FORMAT(//10X,72HNOTE THAT STRUCTURE 2 HAS BEEN TRANSLATED TO ORIGI
     1N OF COORDINATE SYSTEM )
      IF(NTRAN EQ 1) WRITE(IO,110)
      IF(NTRAN EQ 2) WRITE(IO,120)
  110 FORMAT(10X,97HSTRUCTURE 1 HAS BEEN TRANSLATED AND ROTATED TO BRING
     1 IT INTO MAXIMUM COINCIDENCE WITH STRUCTURE 2  )
  120 FORMAT(10X,108HSTRUCTURES 1 AND 3 HAVE BEEN TRANSLATED AND ROTATED
     1 TO BRING EACH INTO MAXIMUM COINCIDENCE WITH STRUCTURE 2  )
      IF(NTRAN EQ 1) WRITE(IO,130) DISCO(1),DIST
  130 FORMAT(//10X,30HINDEX AT MAXIMUM COINCIDENCE = F12 5,1X,A4)
      IF(NTRAN EQ 2) WRITE(IO,140) DISCO,DIST
  140 FORMAT(//10X,39HINDICES AT MAXIMUM COINCIDENCE   1,2 = F12 5,
     1 4X,5H1,3 = F12 5,4X,5H2,3 = F12,5,1X,A4)
C  DETERMINE INTERNAL VARIABLE SET
      CALL SPAN
      CALL VDIST(COORD,COORD(1,1,2),W,S,VF,VI,DISCO(1),NVC,NV,1)
      IF(LINES+6 GT LITOT) CALL PAGE
      LINES=LINES+6
      IF(NTYPE EQ 2) GOTO 170
      WRITE(IO,150) DISCO(1)
  150 FORMAT(//10X,40HINTERNAL VARIABLE DISCOINCIDENCE INDEX = F10 5)
      WRITE(IO,160) P(1),P(2)
  160 FORMAT(/10X,39HPATH COORDINATES FOR STRUCTURES 1 AND 2  2F10 5)
      DRP=DISCO(1)
      GOTO 210
  170 CALL VDIST(COORD,COORD(1,1,3),W,S,VF,VL,DISCO(3),NVC,NV,1)
      CALL VDIST(COORD(1,1,2),COORD(1,1,3),W,S,VI,VL,DISCO(2),NVC,NV),NVC,NV,1)
      P(2)=0.5D0*(P(1)+P(3)+(P(3)-P(1))*
     1 (DISCO(1)**2-DISCO(2)**2)/DISCO(3)**2)
      DRP=DISCO(3)
      WRITE(IO,180) DISCO
  180 FORMAT(//10X,' INTERNAL VARIABLE DISCOINCIDENCE INDICES     1
     1 ,2 = ,F10 5,4X,5H2,3 = ,F10 5,4X,5H1,3 = ,F10 5 )
      WRITE(IO,190) P(1),P(3),P(2)
  190 FORMAT(/10X,' PATH COORDINATES FOR STRUCTURES 1 AND 3  2F10 5/
     1 /10X,84HPATH COORDINATE FOR STRUCTURE 2 ASSIGNED FROM DISCOINCIDE
     2NCE INDICES (1,2) AND (2,3)  ,F10 5)
      WRITE(JO,200) P(2)
  200 FORMAT(/5X,33HPATH COORDINATE FOR STRUCTURE 2 =   F10 5)
C  TEST APPROPRIATENESS OF QST PATHWAY
      CALL QCHECK
```

```
  210 CONTINUE
C   CHECK TO SEE WHETHER ANY ATOM PAIR REVERSALS WOULD IMPROVE THE
C   MATCHING OF STRUCTURES -- THIS MIGHT INDICATE THAT THE PATH
C   LIMITING STRUCTURES HAD BEEN DEFINED IMPROPERLY
      NSWTCH=0
      DO 290 L=1,NTRN1,2
      DO 280 NA=2,NATMS
      NM1=NA-1
      DO 270 NB=1,NM1
      IF(CORE(NA) .NE. CORE(NB)) GOTO 270
      D0=0.D0
      D3=0.D0
      DO 220 M=1,3
      D0=(COORD(M,NA,2)-COORD(M,NA,L))**2+(COORD(M,NB,2)-COORD(M,NB,L))
     1**2+D0
      D3=(COORD(M,NA,2)-COORD(M,NB,L))**2+(COORD(M,NB,2)-COORD(M,NA,L))
     1**2+D3
  220 CONTINUE
      IF(D0.LE.D3) GOTO 270
      NSWTCH=NSWTCH+1
      IL=1
      IF(NSWTCH.EQ.1) IL=6
      IF(LINES+IL.GT.LITOT) CALL PAGE
      LINES=LINES+IL
      IF(NSWTCH.EQ.1) WRITE(IO,230)
  230 FORMAT(///5X,68H*** THE DEFINITION OF THE PATH-LIMITING STRUCTURES
     1 SHOULD BE CHECKED ***/)
      WRITE(IO,240) NA,NB,L
  240 FORMAT( 5X,76H*** MAXIMUM COINCIDENCE WOULD BE IMPROVED BY SWITCH
     1ING COORDINATES FOR ATOMS,I3,4H AND,I3,13H IN STRUCTURE,I2 )
  270 CONTINUE
  280 CONTINUE
  290 CONTINUE
      IF(NSWTCH.EQ.0) GOTO 310
      WRITE(JO,300) NSWTCH
  300 FORMAT(14X,I4,38H PAIRS OF ATOM SWITCHES ARE INDICATED )
      WRITE(JO,230)
  310 CONTINUE
C RETURN IF ONLY PATH COORDINATE ASSIGNMENT IS WANTED
      IF(NCALC.EQ.0) RETURN
  320 CONTINUE
      READ(IN,330) FR,IWRITE,NDENSE,MAX,DELP
      WRITE(JO,330)FR,IWRITE,NDENSE,MAX,DELP
  330 FORMAT(F10.5,3I5,F10.5)
  340 CONTINUE
C   INTERPOLATE INTERNAL VARIABLES
      LIN=0
      CALL STRTP(FR,1)
```

```
      IF(NCALC GT 1) GOTO 347
C PRINT INTERNAL VARIABLES FOR INPUT STRUCTURES
        DO 345 L=1 NTRN1
          IF(LINES+NV/5+6  GT LITOT AND LINES GT 40) CALL PAGE
          LINES=LINES+4
          WRITE(IO 380) P(L)
          IF(L EQ 1)CALL VPRINT(VF COORD(1 1 1) 0 0)
          IF(L EQ 2 AND NTRAN EQ 2)CALL VPRINT(VF COORD(1 1 2) 1 0)
  345     IF(L EQ NTRN1)CALL VPRINT(VL COORD(1 1 L) 0 0)
C   INTERPOLATE CARTESIAN COORDINATES
  347 CALL GETXYZ(C VI)
      IF(LINES+NATMS+6 GT LITOT AND LINES GT 40) CALL PAGE
      LINES=LINES+6
      WRITE(IO 350) FR NDENSE
  350 FORMAT(//10X 'INITIAL COORDINATE ESTIMATES FOR FRACTIONAL POINT '
     1 F10 5 5X 'NDENSE = ' I3//10X 'ATOM' 11X 'X' 12X 'Y' 12X 'Z'/)
      DO 360 N=1 NATMS
      IF(LINES+1 GT LITOT) CALL PAGE
      LINES=LINES+1
  360 WRITE(IO 370) N NORBS(N) CORE(N) C(1 N) C(2 N) C(3 N) WDS(N)
  370 FORMAT(2I3 F6 1 4X 3F10 5 5X A4)
      IF(LINES+NV/5+6  GT LITOT AND LINES GT 40) CALL PAGE
      LINES=LINES+4
      WRITE(IO 380) FR
  380 FORMAT(//10X 'INTERPOLATED INTERNAL VARIABLES FOR FRACTIONAL '
     1 'POINT ' F10 5/)
      CALL VPRINT(VI C 0 0)
C   EVALUATE DISCOINCIDENCE INDICES OF GENERATED STRUCTURE WITH
C   THE PATH-DEFINING POINTS
      DO 410 L=1 NTRN1
  410 CALL VDIST(C COORD(1 1 L) W S VF VL DISCO(L) NVC NV 1)
      IF(LINES+5 GT LITOT) CALL PAGE
      LINES=LINES+5
      WRITE(IO 420) (DISCO(L) L=1 NTRN1)
  420 FORMAT(//10X 'INTERNAL VARIABLE DISCOINCIDENCE INDICES ' 3F12 5)
C   COMPUTE VALUE OF THE PATH COORDINATE FROM THE DISCOINCIDENCE
C   INDICES WITH RESPECT TO THE PATH END POINTS FOR THIS CALCULATION
      L=NTRN1
      TARGET=0 5D0*(P(1)+P(L)+(P(L)-P(1))*
     1 (DISCO(1)**2-DISCO(L)**2)/DRP**2)
      WRITE(IO 430) L TARGET
  430 FORMAT(/10X 75HPATH COORDINATE CALCULATED FROM DISCOINCIDENCE INDI
     1CES TO STRUCTURES 1 AND  I2 2H = F12 5)
      TIME=6 D1*CLOCK(MN)
      WRITE(JO 440) TARGET TIME
  440 FORMAT(/2X 'PATH COORDINATE = ' F9 5 23X 'TIME = ' F7 3 ' SEC')
      RETURN
      END
```

```
C
C ****************
C
      SUBROUTINE SYNTRN
      IMPLICIT REAL*8 (A-H O Z)
C     INCLUDE 'NDFOR:DIM/NOLIST'
      PARAMETER (NUMATM=16,NUMORB=34)
      PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
C
C  DRIVER ROUTINE FOR SYNCHRONOUS TRANSIT CALCULATIONS
C
      COMMON/UNITS/IN,IO,JO,LINES,LITOT
     /SYNCOM/ COEFF(3,NUMATM,3),W(I3NX),VI(I3NX),S(3),P(3),DELP,DRP,
           FR,TARGET,SCALE,NVC(I3NX),NV,NR,NT,NP,NCALC,NTYPE,
           MAX,IWRITE,NDENSE,INVAR,IMPROS,NRWT,MODIFY,NATMS
     /FCOMN / CORE(NUMATM),WDS(NUMATM),NORBS(NUMATM)
     /FUNCT / ENSCF,DUM(3)
     /CYCLES/ NPBI,NPETER
     /ATOMS / C(3,NUMATM),NNUM,IDUM(I3NX)
     /INTVAR/ VF(I3NX),VL(I3NX),Z(I3NX),BF(MPAK),BL(MPAK),DFL,
           LOCVI(I3NX),MAX2
     /RHO   / PP(MPAK),B(MPAK-10),AA(5),R0(5)
     /DFP   / X(I3NX),NVAR
     /PARM1 / A(3,NUMATM),NATMD,NI(NUMATM),NA(NUMATM),NB(NUMATM),
           ND(NUMATM),NN(NUMATM),LOC(2,I3NX),IDDDD(2)
     /FLAGS / SCFCRT,COMENT(5),TITLE(6),SECADD,TIME1,KITSCF,
           KHARGE,KOUTPT,IDUMMY,KGEOM,KOUNT,KSYM,KDEP,MIDDLE,
           ITIME,NTYP,KRESET,IWADE,NTO,NREM,MAXEND
     /MMSYNC/ NDHA,NTRAN,NTRN1,IJKL(2,NUMATM)
      DIMENSION NC(4)
C
      PI=3.1415926536D0
      TWOPI=PI+PI
      NPETER=0
      IF(LINES.GT.15) CALL PAGE
      READ(IN,10) NTYPE,NCALCS,IPRINT,INVAR,MODIFY,NRWT,IMPROS
      WRITE(JO,10) NTYPE,NCALCS,IPRINT,INVAR,MODIFY,NRWT,IMPROS
   10 FORMAT(7I5)
      IF(IABS(NTYPE).EQ.NTRAN) GOTO 30
      WRITE(IO,20) NTYPE
   20 FORMAT(////10X,'*** ILLEGAL NTYPE = ',I10,' IN SYNCHRONOUS TRAN
     1'SIT  RUN ABORTED ****')
      STOP
   30 READ(IN,40) P(1),P(2)
      WRITE(JO,40) P(1),P(2)
   40 FORMAT(2F10.5)
      P(NTRN1)=P(2)
      NCALC=0
```

```
C  SET NCALC = -1 IF ONLY MAXIMUM COINCIDENCE AND PATH COORDINATE
C  ASSIGNMENT IS TO BE DONE
      IF(NCALCS LE 0) NCALC= 1
      IF(NCALCS GT 0) GOTO 80
      IF(LINES+5 GT LITOT)CALL PAGE
      LINES=LINES+5
      WRITE(IO,60)
   60 FORMAT(// 5X, 62H* NO POINTS ON SYNCHRONOUS-TRANSIT PATHWAY HAVE B
     1EEN REQUESTED //9X,97HONLY MAXIMUM COINCIDENCE AND (IF QST) ASSIGN
     2MENT OF PATH COORDINATE FOR STRUCTURE 2 WILL BE DONE  )
   80 IF(NTRAN EQ 1) WRITE(IO,90)
   90 FORMAT(//29X,38HLINEAR SYNCHRONOUS TRANSIT CALCULATION//10X,
     1 43HCOORDINATES OF INITIAL AND FINAL STRUCTURES//7X,4HATOM,
     2 10X,'X',10X,'Y',10X,'Z',15X,'X',10X,'Y',10X,'Z'/ )
      IF(NTRAN EQ 2) WRITE(IO,100)
  100 FORMAT(//28X,'QUADRATIC SYNCHRONOUS TRANSIT CALCULATION'//10X,
     1 'COORDINATES OF INITIAL, INTERMEDIATE, AND FINAL STRUCTURES'//
     2 7X,'ATOM',10X,'X',10X,'Y',10X,'Z',15X,'X',10X,'Y',10X,'Z',
     3 15X,'X',10X,'Y',10X,'Z'/)
      LINES=LINES+8
      DO 105 J=1,NATMS
      IF(LINES+1 GT LITOT) CALL PAGE
      LINES=LINES+1
  105 WRITE(IO,106) J,CORE(J),((COEFF(I,J,M),I=1,3),M=1,NTRN1)
  106 FORMAT(I8,F5.1,F12.5,2F11.5,2(5X,3F11.5))
  110 CONTINUE
      NCALC=NCALC+1
      IF(NCALC.GT.NCALCS) GOTO 210
C  EXECUTE SYNCHRONOUS TRANSIT GEOMETRY CALCULATION
      CALL SYNCAL(C)
      IF(NCALCS LE 0) RETURN
C
C  EXECUTE SCF AND ANY AUXILIARY CALCULATIONS SPECIFIED BY THE
C  EXECUTION CONTROL FLAGS
      IWADE=0
      CALL SCF
      CALL ELECPC(EPC)
      NPETER=NPETER+1
      IF (LINES+3 GT LITOT) CALL PAGE
      LINES=LINES+3
      WRITE(IO,125) ENSCF
  125 FORMAT(//5X,'HEAT OF FORMATION',9X,F15.5,' KCAL/MOLE')
C  PRINT PATH COORDINATE
      IF(LINES+2 GT LITOT)CALL PAGE
      LINES=LINES+2
      WRITE(IO,130) TARGET,EPC
  130 FORMAT(/10X,'PATH COORDINATES  GEOMETRIC = ',F10.5,
     1 '  ELECTRONIC = ',F10.5)
```

```
C
C  LOCATE ST PATH MAXIMUM
C    SCFCRT=10 D0*SCFCRT
     IF(MAX GT 0) CALL ENIMAX(4 VF VL EPC 0)
C    SCFCRT=0 1D0*SCFCRT
     GOTO 110
C
C GET XYZ COORDS FOR FLANKING POINTS IN GMETRY S ORIENTATION
C AND FILL A X ARRAYS FOR OPTIMIZATION
C
  210 CALL GETXYZ(COEFF(1 1 1) VF)
     CALL GETXYZ(COEFF(1 1 NTRN1) VL)
     MAX2=MAX
C FILL A THE OPTIMIZATION SET AND Z ARRAY
     NBDORD=0
     ZLEN=0 D0
     DO 230 I=1 NVAR
       K=LOC(2 I)
       L=LOC(1 I)
       M=LOCVI(I)
       VAR=VI(M)*S(K)
       A(K L)=VAR
       DV=(VL(M)-VF(M))*S(K)
       IF(LOC(2 I)-2) 222 225 224
  222    IF(NVC(M) GT 0) GOTO 225
         NBDORD=NBDORD+1
         VAR=10 D0**((VAR-R0(NBDORD))/AA(NBDORD))
         IA=MAX0(L*(L-1)/2+NA(L) NA(L)*(NA(L)-1)/2+L)
         DV=BL(IA)-BF(IA)
         GOTO 225
  224    IF(DV GT PI) DV=DV-TWOPI
       IF(DV LT -PI) DV=DV+TWOPI
  225    Z(I)=DV
       X(I)=VAR
  230    ZLEN=ZLEN+W(M)*DV**2
     ZLEN=DSQRT(ZLEN)
     DO 400 I=1 NVAR
  400    Z(I)=Z(I)/ZLEN
     RETURN
  240 FORMAT(//10X 'ATOMIC COORDINATES FOR THIS CALCULATION'//
    1  10X 'ATOM' 10X 'X' 12X 'Y' 12X 'Z'/)
  250 FORMAT(8X I5 F15 5 2F13 5)
     END
C
C  **********************
C
     SUBROUTINE VCANON(CX FN VAL N NC NMAX IFV)
     IMPLICIT REAL*8 (A-H O-Z)
```

```
C     INCLUDE 'NDFOR DIM/NOLIST'
      PARAMETER (NUMATM=16)
      PARAMETER (I3NX=3*NUMATM)
C
C FIRST  DECODE CANONICAL ORDER INDEX  THEN  IF
C IFV=1  COMPUTE INTERNAL VARIABLE FROM XYZ
C
      COMMON/UNITS/IN IO JO LINES LITOT
     /SYNCOM/ COORD(3 NUMATM 3) W(I3NX) VI(I3NX) S(3) P(3) DELP DRP
           FR TARGET SCALE NVC(I3NX) NV NR NT NP NCALC NTYPE
           MAX IWRITE NDENSE INVAR IMPROS NRWT MODIFY NATMS
      DIMENSION CX(3 1) C(3 4) D(3 4) FN(3 4) IC(4) NC(4)
      DATA IC TOLER DEL/1 2 3 4 0 99999999D0 5 D 3/
C DECODE THE CANONICAL ORDER INDEX
      NATM1=NUMATM+1
      NATM2=NATM1**2
      NATM3=NATM2*NATM1
      NMAX=4
      NVCN=IABS(NVC(N))
      NC(4)=NVCN/NATM3
      IF(NC(4) EQ 0) NMAX=3
      NC(3)=(NVCN-NATM3*NC(4))/NATM2
      IF(NC(3) EQ 0) NMAX=2
      NC(2)=(NVCN-NATM3*NC(4)-NATM2*NC(3))/NATM1
      NC(1)= NVCN-NATM3*NC(4)-NATM2*NC(3)-NATM1*NC(2)
C   RETURN IF ONLY THE DECODING WAS DESIRED
      IF(IFV EQ 0) RETURN
C   CALCULATE THE BOND LENGTH  ANGLE  OR DIHEDRAL ANGLE
        DO 30 I=1 NMAX
          J=NC(I)
        DO 30 M=1 3
   30     C(M I)=CX(M J)
      IF(NMAX-3) 35 40 45
C   BOND LENGTH
   35 CALL BNDIST(FN C IC VALUE)
      VAL=VALUE
      RETURN
C BOND ANGLE OR DIHEDRAL ANGLE
   40 CALL BNDANG(FN C IC VALUE CANG)
      GOTO 50
   45 CALL DIHDRL(FN C IC VALUE CANG)
C   CHECK FOR UNDEFINED DIHEDRAL ANGLE
   50 IF(CANG LT 1 5D0) GOTO 55
C   SET INDICATOR
      NMAX=-4
      RETURN
   55 VAL=VALUE
      IF(DABS(CANG) LT TOLER) RETURN
```

```
C  SET WARNING FLAG FOR LINEAR A B C ANGLE
     IF(NMAX EQ 3) NMAX= 3
     RETURN
     END
C
C **********
C
     SUBROUTINE VDIST(CA,CB,W,S,VA,VB,DIST,NVC,NV,IFV)
     IMPLICIT REAL*8 (A-H,O-Z)
     PARAMETER (NUMATM=16,NUMORB=34)
     PARAMETER (I3NX=3*NUMATM,MPAK=(NUMORB*(NUMORB+1))/2)
     COMMON/RHO   / PP(MPAK),B(MPAK-10),AA(5),R0(5)
C THIS SUBROUTINE COMPUTES DISTANCE BETWEEN STRUCTURES REPRESENTED
C BY COORDINATES CA AND CB OR BY INTERNAL VARIABLES VA AND VB
     DIMENSION F(3,4),NC(4)
     DIMENSION W(1),CA(3,1),CB(3,1),VA(1),VB(1),S(1),NVC(1)
     DIST=0.D0
     WT=0.D0
     NBDORD=0
     DO 30 N=1,NV
       CALL VCANON(CA,F,V1,N,NC,NMAX,IFV)
CHECK FOR UNDEFINED DIHEDRAL ANGLE
       IF(IFV.EQ.1) THEN
         IF(NMAX.LE.-4) GOTO 30
         CALL VCANON(CB,F,V2,N,NC,NMAX,1)
         IF(NMAX.LE.-4) GOTO 30
       ELSE
         V1=VA(N)
         V2=VB(N)
         IF(V1.GT.1.D05.OR.V2.GT.1.D05) GOTO 30
       ENDIF
       IF(NVC(N).LT.0) THEN
         NBDORD=NBDORD+1
         V1=10.D0**((V1-R0(NBDORD))/AA(NBDORD))
         V2=10.D0**((V2-R0(NBDORD))/AA(NBDORD))
       ENDIF
       DV=V1-V2
       IF(DV.GT. 180.D0) DV=DV-36.D1
       IF(DV.LT.-180.D0) DV=DV+36.D1
       DIST=DIST+W(N)*(DV*S(NMAX-1))**2
       WT=WT+W(N)
  30   CONTINUE
     DIST=DSQRT(DIST/WT)
     RETURN
     END
```

APPENDIX B

The structure of the modified MNDO program is illustrated in the listing below of the calling sequence. This listing was made by my thesis advisor, Dr. Thomas A. Halgren, with the aid of the Program Structure Analyzer written at the Merck Sharp and Dohme Research Laboratories by Dr. Joseph D. Andose. To avoid redundancies and produce a compact listing, replicate calls to a reused used code section "n" are indicated symbolicly by the notation "<n>..."; the full call sequence appears following the notation "<n>" in the PARENT REFERENCE column.

```
CALL SEQUENCE                                   PARENT REFERENCE   REPLICATE REFERENCE
-------------------------------------------------------------------------------------

MNDO$MAIN
.   ERRSET
.   CLOCK1
.   START
.   .   OBFAC
.   .   METHOD
.   .   INPUT
.   .   .   PUNIN
.   .   .   INPXYZ
.   .   .   .   PAGE                                                  <28>...
.   .   .   ESTIM
.   .   .   .   RDXYZ
.   .   .   .   .   BNDANG                                            <26>...
.   .   .   .   .   DIHDRL                                            <27>...
.   .   .   .   SYMTRY                                                <12>...
.   .   .   GMETRY
.   .   .   AXIS                                        <1>
.   .   .   .   RSP                                     <2>
.   .   .   .   .   TRED3
.   .   .   .   .   TQLRAT
.   .   .   .   .   TQL2
.   .   .   .   .   TRBAK3
.   .   .   SETUP
.   .   .   GUESSP
.   SYNTRN
.   .   PAGE                                                           <28>...
.   .   SYNCAL
.   .   .   BONDS
.   .   .   .   CONNEC
.   .   .   .   .   AXIS                                              <1>....
.   .   .   .   .   GUESSP
.   .   .   .   .   FMTRX                                             <13>...
.   .   .   .   .   DAOPOP                                            <25>...
.   .   .   .   .   VECPRT
.   .   .   PAGE                                                      <28>...
```

CALL SEQUENCE	PARENT REFERENCE	REPLICATE REFERENCE
. . . SPAN		
. . . . TABINT		
. . . . . ERROR		
. . . . PATH		
. . . . . NXTATM		
. . . . . . ERROR		
. . . . APPEND		
. . . . FRSTAB		
. . . . . ERROR		
. . . . AUSED		
. . . . . ERROR		
. . . . NXTTAB		
. . . . . ERROR		
. . . . . TBSRCH		
. . . . . . ERROR		
. . . . FILNVC		
. . . . . PAGE		<28>...
. . . VDIST	<3>	
. . . . VCANON	<4>	
. . . . . BNDIST		
. . . . . BNDANG		<26>...
. . . . . DIHDRL		<27>...
. . . QCHECK		
. . . . VCANON		<4>....
. . . . PAGE		<28>...
. . . STRTP	<5>	
. . . . INTERP		
. . . . VCANON		<4>....
. . . . PAGE		<28>...
. . . VPRINT	<6>	
. . . . PAGE		<28>...
. . . . VCANON		<4>....
. . . GETXYZ		<9>....
. . . CLOCK		
. . SCF		<10>...
. . ELECPC	<7>	
. . . OAOPOP		<25>...
. . ENIMAX	<8>	
. . . PAGE		<28>...
. . . PTHADJ		
. . . . PAGE		<28>...
. . . . STRTP		<5>....
. . . . GETXYZ		<9>....
. . . . VDIST		<3>....
. . . . GUESSP		
. . . . SCF		<10>...
. . . . ELECPC		<7>....

CALL SEQUENCE	PARENT REFERENCE	REPLICATE REFERENCE
. . . POLY		
. . . . APCH		
. . . . APFS		
. . . . PAGE		<28>...
. . . . BACKCH		
. . . . PVAL		
. . . . PDER		
. . . INTERP		
. . . VPRINT		<6>....
. . GETXYZ	<9>	
. . . GMETRY		
. SCF	<10>	
. . CLOCK		
. . PAGE		<28>...
. . PUNOUT	<11>	
. . . SYMTRY	<12>	
. . . . HADDON		
. . . . . DEPVAR		
. . . CLOCK		
. . . PAGE		<28>...
. . FMTRX	<13>	
. . . SALVAG		
. . . CLOCK		
. . . GS	<14>	
. . . . HCORE	<15>	
. . . . . BONTYP		
. . . . . OVERLP		
. . . . . ROTATE		
. . . . . REPP		
. . . . . ITER		
. . . . . RSP		<2>....
. . . . . BORDER		
. . . . . ESCF		
. . . . . VECPRT		
. . . . . SPCG		
. . . . . OPCI		
. . . . . SPCG		
. . . . . CIORB		
. . . . . ETYPE0		
. . . . . MATCH		
. . . . . ETYPE1		
. . . . . MATCH		
. . . . . ETYPE2		
. . . . . MATCH		
. . . . . SQUIRF(ETYPE0)		
. . . . . MATCH		

CALL SEQUENCE	PARENT REFERENCE	REPLICATE REFERENCE
. . . . . EO(ETYPEO)		
. . . . . . MATCH		
. . . . . RSP		<2>....
. . . SKP	<16>	
. . . . HCORE		<15>...
. . . UHF	<17>	
. . . . HCOREO		
. . . . . BONTYP		
. . . . . OVERLP		
. . . . . ROTATO		
. . . . . REPP		
. . . . . ITERO		
. . . . . BIRST		<18>...
. . . . . RSP		<2>....
. . . . . UBORD		
. . . . . ESCF		
. . . . . VECPRT		
. . . . BIRST	<18>	
. . . . . HCORE		<15>...
. . . . . RSP		<2>....
. . . . . UBORD		
. . . VALUE		
. . . . GS		<14>...
. . . . SKP		<16>...
. . . . UHF		<17>...
. . . . SALVAG		
. . . . FITP		
. . . DOT		
. . . DXYZ	<19>	
. . . . DEROVR		
. . . . DELRI		
. . . . DELMOL		
. . . . . ROTAT		
. . . MATOUT		
. . . FRCE		
. . . . RAMS		
. . . . FGMTRX		
. . . . . RSP		<2>....
. . . . . MATOUT		
. . DRTP	<20>	
. . . DSINV	<21>	
. . . . DMFSD		
. OPT		
. . DFPMIN		
. . . PTHCRD	<22>	
. . . . VDIST		<3>....

CALL SEQUENCE	PARENT REFERENCE	REPLICATE REFERENCE
. . . PAGE		<28>...
. . . CLOCK		
. . . ELECPC		<7>....
. . . COMPFG	<23>	
. . . . SYMTRY		<12>...
. . . . GMETRY		
. . . . SCF		<10>...
. . . CPHF		
. . . . CLOCK		
. . . . GMETRY		
. . . . SCPHF		
. . . . . INV		
. . . PUNOUT		<11>...
. . . ORTHOG		
. . . . PTHCRD		<22>...
. . . ZSHFT		
. . . . PTHCRD		<22>...
. . . . COMPFG		<23>...
. . . . ELECPC		<7>....
. . . . PAGE		<28>...
. . . . CLOCK		
. . . . PTHMAX		<24>...
. . . DSINV		<21>...
. . . PTHMAX	<24>	
. . . . PTHCRD		<22>...
. . . . GETXYZ		<9>....
. . . . STRTP		<5>....
. . . . VDIST		<3>....
. . . . ENIMAX		<8>....
. . . . DXYZ		<19>...
. . . . DRTP		<20>...
. . . . ELECPC		<7>....
. . . HDIAG		
. . . . DSINV		<21>...
. . . . EIGEN7		
. . . . PAGE		<28>...
. FINAL		
. . PRTDFP		
. . . SYMTRY		<12>...
. . . GMETRY		
. . . VECPRT		
. . PRTCIS		
. . . VECPRT		
. . . MATOUT		
. . PRTSCF		
. . . VECPRT		

CALL SEQUENCE	PARENT REFERENCE	REPLICATE REFERENCE
. . . MATOUT		
. . . DIPOLE		
. . OAOPOP	<25>	
. . . VECPRT		
. . PRTLST		
. . . GMETRY		
. . . VECPRT		
. . NABOR		
. . . BNDANG	<26>	
. . . . DARCOS		
. . . DIHDRL	<27>	
. . . . DARCOS		
. CLOCK		
. PAGE	<28>	
. . CLOCK		

References

1. O'Neil, S. V., Pearson, P. K., III, H. F. Schaefer, and Bender, C. F., *J. Chem. Ph s.*, vol. 58, p. 1126, 1973.
2. Shavitt, I., Stevens, R. M., Minn, F. L., and Karplus, M., *J. Chem. Ph s.*, vol. 48, p. 2700, 1968.
3. Ransil, B. J., *J. Chem. Ph s.*, vol. 26, p. 971, 1957.
4. Dewar, M. J. S. and Kirschner, S., *J. Am. Chem. Soc.*, vol. 93, p. 4291, 1971.
5. Dewar, M. J. S. and Kirschner, S., *J. Am. Chem. Soc.*, vol. 96, p. 5244, 1974.
6. Dewar, M. J. S. and Kirschner, S., *J. Am. Chem. Soc.*, vol. 96, p. 6809, 1974.
7. Rothman, M. J. and Lohr, L. L., *Chem. Ph s. Lett.*, vol. 70, p. 405, 1980.
8. Halgren, T. A. and Lipscomb, W. N., "The Synchronous-Transit Method for Determining Reaction Pathways and Locating Molecular Transition States," *Chem. Ph s. Lett.*, vol. 49, pp. 225-232, 1977.
9. McIver, J. W. and Komornicki, A., *J. Am. Chem. Soc.*, vol. 94, p. 2625, 1972.
10. Peckham, G., *Comput. J.*, vol. 13, p. 418, 1970.
11. Fletcher, R., *Practical Methods of Optimization: Unconstrained Optimization*, 1, Wiley, New York, 1980.
12. Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, vol. 13, pp. 317-322, 1970.
13. Powell, M. J. D., "Recent Advances in Unconstrained Optimization," *Mathematical Programming 1*, vol. 1, pp. 26-57, North-Holland Publishing Company, 1971.
14. Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method For Minimization," *Computer Journal*, vol. 6, pp. 163-168, 1963.
15. Davidon, W. C., *Computer Journal*, vol. 10, p. 406, 1968.
16. Pulay, P., Forarasi, G., Pang, F., and Boggs, J. E., *J. Am. Chem. Soc.*, vol. 101, p. 2250, 1979.
17. Schlegel, H. B., *J. Comput. Chem.*, vol. 3, p. 214, 1982.
18. Halgren, T. A. and Lipscomb, W. N., "Approximations to Self-Consistent Field Molecular Wavefunctions," *Proc. Nat. Acad. Sci. USA*, vol. 69, pp. 652-656, 1972.
19. Halgren, T. A., *(personal communication)*.
20. Baker, "An Algorithm for the Location of Transition States," *J. of Comp. Chem.*, vol. 7, pp. 385-395, 1986.
21. Bell, S., Crighton, J. S., and Fletcher, R., *Chem. Ph s. Lett.*, vol. 82, p. 122, 1981.
22. Bell, S. and Crighton, J. S., "Locating Transition States," *J. Chem. Ph s.*, vol. 80, pp. 2464-2475, 1984.

23. Dewar, M. J. S. and Thiel, W., "Ground States of Molecules. The MNDO Method. Approximations and Parameters," *J. Am. Chem. Soc.*, vol. 99, pp. 4899-4907, 1977.

24. Dewar, M. J. S. and Yamaguchi, Y., "Analytical First Derivatives of the Energy in MNDO," *Computers & Chemistr* , vol. 2, pp. 25-29, Pergamon Press Ltd., 1978.

25. Pople, J. A., Krishnan, R., Schlegel, H. B., and Binkley, J. S., "Derivative Studies in Hartree-Fock and Moller-Plesset Theories," *International Journal of Quantum Chemistr* , vol. 13, pp. 225-241, John Wiley & Sons, Inc., 1979.

26. Armstrong, D. R., Perkins, R. G., and Stewart, J. J. P., *J. Chem. Soc. Dalton Trans.*, vol. 838, 1973.

27. Pauling, Linus, *The Nature of The Chemical Bond*, Cornell University Press, Ithica, NY, 1960.

28. Schroder, Stefan and Thiel, Walter, "Comparison of Semiempirical and AB INITIO Transition States for Organic Reactions," *Journal of Molecular Structure (Theochem)*, vol. 138, pp. 141-150, Elsevier Sciencer Publishers B.V., Amsterdam, 1986.

29. Schroder, Stefan and Thiel, Walter, "Comparison of Semiempirical and AB INITIO Transition States," *J. Am. Chem. Soc.*, vol. 107, pp. 4422-4430, 1985.