ANL-7803

RDE SITE OFFICE
ANL, Idaho
Rec. 11-9
File library

GPTHEORY: A FORTRAN Computer Program for Determining Molecular Symmetry Properties

Thomas D. Bouman and Gordon L. Goodman

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95

ANL-7803
Mathematics and Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

GPTHEORY: A FORTRAN Computer Program for Determining Molecular Symmetry Properties

by

Thomas D. Bouman* and Gordon L. Goodman

Chemistry Division

April 1971

*Faculty of Chemistry, Southern Illinois University, Edwardsville

2

TABLE OF CONTENTS

	Page
ABSTRACT	7
I. INTRODUCTION	7
II. THEORY AND TECHNIQUES	11
A. Basic Group Theory	11
B. Generation of Standard Representatives for the Symmetry Species of the Finite Point Group	12
1. The Totally Symmetric Operator	12
2. Choice of Basis Functions	13
3. Construction of Crude Symmetry Vectors	14
4. Case of Inversion Symmetry	15
5. Imposition of a Standard Phase Choice	16
6. Example: The Octahedral Group O_h	17
C. Symmetry-adaptation of an Arbitrary Closed Set of Functions	18
1. Allowed Function Classes	18
2. Example: Atomic Basis Set	19
3. Symmetrization Procedure	20
D. Finite Point-group Vector-coupling Coefficients	21
1. Definition and Properties	21
2. Method of Calculation	22
3. Antisymmetric Direct Products	24
4. Extensions and Uses	24
E. Phase Alignment of Sets of Functions Carrying the Same Symmetry Species	25
III. PROGRAM DESCRIPTIONS AND ALGORITHMS	26
A. Design Philosophy	26
B. Logic Flow	26
C. Descriptions of the Subprograms and Their Algorithms	30
1. Driver and Initialization	30
2. Input and Molecular-geometry Analysis	30
3. Routines Common to More Than One Major Segment	31
4. Determination of Point-group Properties	32

TABLE OF CONTENTS

Page

5. Symmetry-adaptation of an Atomic Basis Set 34
6. Finite Point-group Vector-coupling Coefficients 36

D. Definition of Variables in Common Blocks 37

1. Glossary 37
2. Blocks Containing Output from the Program 37
3. Blocks Containing Information Internal to the Program . 40

IV. INSTRUCTIONS FOR USE 44

A. Description of Input 44

B. Built-in Basis Sets 44

C. Sample Input and Program Output 48

D. Table Limits and Error Messages 49

APPENDIX: Source-code Listings 67

ACKNOWLEDGMENTS 106

REFERENCES 107

LIST OF FIGURES

No.	Title	Page
1.	Disposition of the Set of Points S at Which the Operator $\mathcal{P}$ of Eq. 5 Is Defined, Shown for an Octahedral MX_6 Molecule.	12
2.	Structure of Matrix $\underset{\sim}{M}$ from Eq. 10, Showing Off-diagonal Blocks Connecting Sets of Functions Belonging to the Same Symmetry Species	15
3.	Computer Plots of Real, Symmetry-adapted Spherical Harmonics Centered at the Origin Giving the Simplest Occurrence of Each Even-parity Representation of O_h	18
4.	Computer Plots of the Simplest Occurrence of Each Odd-parity Irreducible Representation of O_h	18
5.	Symmetry-adapted Prototype Functions (L = 6) Spanning All the Even-parity Representations of O_h.	19
6.	Symmetry-adapted Prototype Functions (L = 9) Spanning All the Odd-parity Representations of O_h	19
7.	Flowchart of Overall Logic Flow	27
8.	Flowchart for Molecular-geometry Analysis Section	28
9.	Flowchart for Determination of Point-group Properties and Prototype Functions	28
10.	Flowchart for Symmetry-adaptation of Atomic Basis Set.	29
11.	Flowchart for Calculation of Vector-coupling Coefficients. . . .	29
12.	A Possible Overlay Structure for the GPTHEORY Program. . .	29
13.	Example of Input Deck Including a Basis Set Supplied by the User	46
14.	Example of Input Deck with Basis Set Selected by the Program	46
15.	Sample Output from the Program for an Octahedral KrF_6 Molecule, Quantized about the Fourfold Axis.	51

LIST OF TABLES

No.	Title	Page
I.	Symmetry Species of the Point Group O_h	17
II.	L-values Used in the Prototype Functions for the Various Point Groups	33
III.	Description of Input Deck	45
IV.	Prestored Minimal Valence-shell Atomic Basis Set	46
V.	Examples of Double-zeta Slater Basis Sets	48
VI.	Summary of Input and Output for the Computer Program as Currently Implemented	48

GPTHEORY: A FORTRAN Computer Program for Determining Molecular Symmetry Properties

by

Thomas D. Bouman and Gordon L. Goodman

ABSTRACT

Many simplifications and insights into quantum-mechanical calculations on molecules are possible if one uses the techniques and theorems of molecular point-group theory. However, because of the difficulty in translating the standard textbook methods of group theory into a form amenable to use on a computer, these simplifications have not been generally exploited in large-scale machine computations. Despite avoiding explicit consideration of the nonnumeric operations of the point group, we have succeeded in implementing a method in FORTRAN IV which allows one to generate all the relevant symmetry information about the molecule, requiring as input only the type of nuclei, their spatial coordinates, and parameters characterizing the atomic basis set. The method depends on the construction of a simple, artificial operator having the symmetry of the nuclear configuration and the generation of a set of prototype functions spanning all the irreducible representations of the group. These may be used to generate various types of symmetry information, including a symmetry-adapted atomic basis set, vibrational symmetry coordinates, and finite point-group vector-coupling coefficients. Listings of the computer program, known as GPTHEORY, and complete instructions for use are given.

I. INTRODUCTION

The theorems and techniques of group theory as applied to molecules with symmetry have long been recognized as a powerful tool for the theoretical chemist.[1-3] Use of symmetry-based selection rules, for example, has shed much light on the interpretation of atomic and molecular spectra[4,5] and has led to many simplifications in quantum-mechanical calculations.[6] Methods developed by Racah[7,8] for atomic spectroscopy and extended to molecules by Tanabe and co-workers[9,10] and by Griffith[11,12] have made possible the determination of quantitative as well as qualitative relationships from the symmetry properties of a problem.

Until fairly recently, however, molecular symmetry properties have been generally ignored in organizing problems in molecular quantum mechanics for solution on large, high-speed digital computers. The resulting type of procedure, in which one allows the computer to calculate and manipulate all quantities in the same way, even those that group theory would say must vanish, may be termed a "brute-force" approach. The tendency to use this type of approach arises because of difficulties in translating the nonnumeric operations and manipulations of molecular point-group theory to a form that can be conveniently handled on a digital computer. Nevertheless, quantum chemists have recently come to appreciate again that an investigator who ignores symmetry not only gives up an important means of alleviating computational bottlenecks, but also foregoes insights that symmetry can give to the results of a large calculation. Thus, increased attention is currently being paid to techniques for implementing symmetry aspects for machine calculations.

Several different lines of attack have been followed in efforts to take account of symmetry in designing machine programs for quantum chemistry; a few examples, which are by no means exhaustive, will serve to illustrate these lines of attack. One method is to design a computer program around the specific symmetry properties of a particular point group, and thus to take full advantage of the qualitative simplifications occurring in a restricted class of molecules. Examples of this approach appear in the work of Joshi[13] on hydrides with C_{3v} symmetry, that of Wahl[14] on homonuclear diatomic molecules (symmetry $D_{\infty h}$), and that of McLean and Yoshimine[15] on linear molecules in general.

A second category includes computer programs which are designed to accommodate the simplifications arising in a wider class of point groups, but which require the specification of various amounts of group-theoretical information as input. The IBMOL program system of Clementi and coworkers,[16] for example, uses selection rules in eliminating the handling of zeros; however, the user must provide the program with a symmetry-adapted basis set and certain indices characterizing the point group. The POLYATOM system[17] generates lists of nonvanishing integrals by extensive manipulation of input tables of symmetry operations and their effects on atoms and atomic basis functions. Symmetry- and spin-adapted Slater determinantal configuration functions for molecules with no higher than two-fold degeneracies may be generated automatically by the projection operator method of Gershgorn and Shavitt,[18] if a symmetry-adapted atomic basis set is provided. A semiempirical calculation organized for symmetry has been programmed by Glarum,[19] who requires the user to supply the irreducible representation matrices for the point group under consideration. Again, no degeneracies higher than two are permitted. A final example in this category is the automation by several investigators of problems in molecular vibration and rotation.[20] The methods have in common the requirement that the user provide the transformation from internal to symmetry coordinates.

Several attempts have been made to generate the required group-theoretical information on the computer. Possibly the first such method to be implemented was developed by Flodmark and Blokker.[21] These authors concentrated on the detailed properties of the group itself, such as the actual irreducible representation matrices, and generated symmetry-adapted functions with an automated version of the same techniques and formalism one employs in discussing the symmetry aspects of such a problem in a group-theory textbook. The user must supply the program with the multiplication table of the group. Gabriel[22,23] has developed and is implementing a computer-oriented method for finding symmetry-adapted functions which avoids explicit use of the representation matrices, but makes use of the generators for the group and their commutator algebra. His work may be regarded as providing a rigorous theoretical justification for the method we have developed.

An approach much closer in spirit to the one we have implemented has been proposed by Moccia.[24] Rather than working with the symmetry-group operations and irreducible representations themselves, he uses instead group-theoretical theorems which basis functions and operators must satisfy if they are to be symmetry-adapted. Standard techniques of calculating quantum-mechanical integrals and diagonalizing matrices are employed to generate a symmetrized basis set in a numerical way well suited to the capabilities of a digital computer. Moccia's method, however, is dependent on the assumption that the unsymmetrized atomic basis functions comprise sets of "equivalent orbitals"[25] which are permuted among themselves by the symmetry operations of the point group. Thus, a pre-transformation of the atomic basis functions into spatially equivalent orbitals must be performed before the method can be used; this may be a nontrivial task, especially if extended basis sets are employed. Furthermore, the construction of spatially equivalent orbitals in some cases requires hybridization of the atomic basis functions; this involves assumptions about the nature of the basis functions which go beyond those strictly determined by symmetry.

Chung and Goodman[26,27] have extended Moccia's work to obviate the need for using "equivalent orbitals" and have succeeded in producing canonical symmetry-adapted atomic basis sets from the atomic coordinates and initial Slater-type atomic orbitals alone. Their work and the present work have influenced each other.

To obtain symmetry coordinates for their work on vibrational spectra of polymers and other molecules, Gussoni and Zerbi[28] have implemented a method similar in spirit to Moccia's. By setting up the $\underset{\sim}{G}$ matrix for a set of internal nuclear-displacement coordinates and diagonalizing it, they produced a transformation that takes the original internal coordinates into symmetry coordinates.

Our present work is motivated in part by the desire to develop an efficient and systematic way of introducing symmetry into large-scale computing efforts, and also by the wish to explore new algorithms for solving standard problems--algorithms suggested by the numerical capabilities of digital computers and their associated scientific programming languages. If such a method is to be widely used, it should fulfill the requirements of speed, generality, simplicity of input, and transferability from one type of problem to another.

In the following sections, we describe a method of generating group-theoretic information on a machine in a form that is readily usable for subsequent computation. As we show, it is in fact possible to obtain virtually all relevant symmetry information, both qualitative and quantitative, for a quantum-mechanical problem without recourse to any standard textbook representation for, or explicit consideration of, the elements forming the point group. We adopt the point of view that what really interests us is not rotations, reflections, etc., nor group multiplication tables and irreducible representation matrices, nor even how functions transform under the symmetry operations. We are interested rather only in numerical relationships, such as whether a matrix element is zero or nonzero, and what ratios exist between nonzero elements. Hence we work with matrix elements from the start.

In the method to be described, we rely substantially on the following theorem in group theory, which we state without proof:

> Theorem. If a matrix has a structure embodying the full symmetry of the particular nuclear geometry, its eigenvectors form bases for irreducible representations of the point group of the molecule, provided that there are no accidental degeneracies in the eigenvalues.

A matrix having this structure is the matrix of an operator that itself possesses the symmetry of the nuclear framework. In the statement of this theorem, "a structure embodying the full symmetry of the particular nuclear geometry" means a structure that is left unaltered by each operation of the point group to which the nuclear framework belongs, but one that is altered by any operation under which the nuclear framework is not invariant.

In our method, two artificial operators totally symmetric with respect to the nuclear framework are constructed, and a set of functions is chosen to span all the symmetry species occurring for the point group.* These functions then comprise a standard set against which the symmetry aspects of any other functions can be aligned, and with which other group-theoretic information can be obtained.

---

*It is clear at this point that linear molecules should be excluded from this method, since their possible symmetry species are not finite in number. However, this class of molecules is already well endowed with computer-oriented techniques taking advantage of symmetry.

II. THEORY AND TECHNIQUES

A. Basic Group Theory

We begin this section with a brief review of representation theory. Let $\underset{\sim}{\varphi}$ be a set of functions $\varphi_1, \varphi_2, \ldots$, associated in some way with a molecule belonging to a point group G. We assume that $\underset{\sim}{\varphi}$ is complete in the sense that any symmetry operation $\underline{R}$ of G transforms each of the functions, φ_j, say, into a linear combination of the other functions in the set

$$\varphi_j' \equiv \underline{R}\varphi_j = \sum_k \varphi_k A_{kj}(\underline{R}), \tag{1}$$

or, in matrix-vector notation,

$$\underset{\sim}{\varphi}' = \underset{\sim}{\varphi}\underset{\sim}{A}(\underline{R}). \tag{2}$$

Thus, each such operation $\underline{R}$ has a matrix $\underset{\sim}{A}(\underline{R})$ associated with it. One can easily show (see, e.g., Ref. 6) that the set of matrices $\underset{\sim}{A}(\underline{R})$ forms a <u>representation</u> of the group G. Although this representation is in general not the simplest possible, we can find a reduction matrix $\underset{\sim}{U}$, corresponding to a unitary transformation of $\underset{\sim}{\varphi}$ that generates a new set of representation matrices $\underset{\sim}{B}(\underline{R})$ which factor into the simplest possible or <u>irreducible representations</u> of G,

$$\underset{\sim}{B}(\underline{R}) = \underset{\sim}{U}^{-1}\underset{\sim}{A}(\underline{R})\underset{\sim}{U}. \tag{3}$$

The same transformation applied to the functions $\underset{\sim}{\varphi}$ converts them to a set of functions $\underset{\sim}{\psi}$ which form bases for irreducible representations of G:

$$\underset{\sim}{\psi} = \underset{\sim}{\varphi}\underset{\sim}{U}. \tag{4}$$

This new set of functions is said to be <u>symmetry-adapted</u> to the group G, and satisfies all the usual symmetry-selection rules and other relationships.

Our interest therefore centers on the determination of the reduction matrix $\underset{\sim}{U}$ for a given set of functions, in a computer-oriented way. We identify three major stages in our procedure. First, the functions are transformed to a set decomposed according to definite symmetry types; second, particularly simple functions are extracted from these initial symmetrized functions; finally, a standard phase choice is imposed to yield canonical symmetrized functions.

B. Generation of Standard Representatives for the Symmetry Species of the Finite Point Group

1. The Totally Symmetric Operator

We address ourselves first to the choice of an operator P which possesses the full symmetry of any given nuclear geometry. The nuclear-attraction potential field and, of course, the complete Hamiltonian operator for a molecule are examples of such operators which normally occur in a quantum-mechanical calculation. We want to construct, however, an artificial operator for which the matrix elements are almost trivial to compute, and which has no more and no less invariance under rotations and reflections than do these naturally occurring ones.

A simple operator having these symmetry properties, one that avoids placing undue weight on symmetry-dictated nodal surfaces occurring for various functions, is the operator that annihilates a function everywhere except at a finite number of points that are members of the set S:

$$P(\underset{\sim}{r}) = \sum_{\nu \in S} \delta(\underset{\sim}{r} - \underset{\sim}{r}_\nu) \tag{5}$$

such that the points in S lie on circles defining a plane perpendicular to the line joining each nucleus to the center of nuclear charge of the molecule of interest, as shown in Fig. 1. Evaluation of the matrix elements of P then reduces to summing the values of the integrand at the points in S.

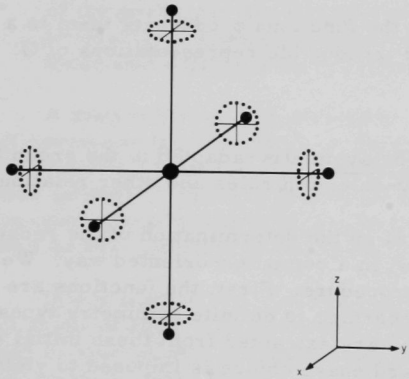

Fig. 1

Disposition of the Set of Points S at Which the Operator P of Eq. 5 Is Defined, Shown for an Octahedral MX_6 Molecule. ANL Neg. No. 121–4299.

This set of points defining P may be viewed either as one that goes identically into itself under all the operations of the point group G, or as an approximation to a set of symmetrically disposed circles (where the summation in Eq. 5 is replaced by a sum of line integrals). In practice,

the two interpretations appear to be equivalent, in that no practical accuracy is lost if the points do not exactly transform into one another. Transformation coefficients good to 11 significant figures are generally obtained with 24 equally spaced points on each circle, even in cases in which the circle encloses a fivefold symmetry axis. Our choice of operator has the advantage not only of making the computations simple, but also of freeing as much as possible the determination of symmetry properties from other details of any problem in which they are to be used.

2. Choice of Basis Functions

In choosing a "standard set" of functions to symmetry-adapt, we require that each symmetry species occur at least once in the representation spanned by these functions. A particularly convenient choice is the set of 2L + 1 orbital-angular-momentum eigenfunctions, $Y_{LM}(\theta,\varphi)$, of appropriately high L-value. In a spherically symmetric environment, these functions span a (2L + 1)-fold-degenerate irreducible representation of the three-dimensional full rotation group SO(3). As the symmetry of the environment is lowered, this erstwhile irreducible representation splits into the various irreducible representations appropriate to the finite point group. By choosing L high enough, therefore, we are assured that each possible new representation will be spanned at least once.[2,29,30] If a center of inversion is present, two distinct L values are in fact required: an even one, L_g, for the representations that are even with respect to inversion, and an odd one, L_u, for those of odd parity. (The selection of the proper set of functions for any particular case of chemical significance, i.e., for any point group having no higher than an eightfold rotation axis, is built into the program. It can be shown that L = 16 is the highest one needs to span all the symmetry species that occur in these cases.)

For convenience of computation and ease of visualization, we choose the following real combinations of the Y_{LM}:[11]

$$\left.\begin{aligned} S_0^{(L)} &= Y_{LO}; \\ S_{2M}^{(L)} &= \frac{i}{\sqrt{2}}(Y_{L,-M} - Y^*_{L,-M}); \\ S_{2M+1}^{(L)} &= \frac{1}{\sqrt{2}}(Y_{L,-M} + Y^*_{L,-M}); \end{aligned}\right\} \qquad (6)$$

where

$$Y_{L,-M} = (-1)^M Y^*_{LM}.^{\dagger}$$

---

†See Ref. 8, where Eqs. 2.5.6 and 2.5.29 provide the definitions used here for the $Y_{L,M}$.

This entails no loss of generality. We index the functions sequentially by the integer K, K = 1, 2, ..., 2L + 1. Let this set be denoted $S_K^{(L)}$, or simply $\underset{\sim}{S}^{(L)}$; a simple radial dependence, $R(r) = r^{-1}$, is assumed a part of the functions $\underset{\sim}{S}^{(L)}$ in order to avoid accidental degeneracies. The functions are centered at the center of nuclear charge, which is chosen as the origin of the master coordinate system.

3. Construction of Crude Symmetry Vectors

By the Theorem in Sec. I, a unitary transformation $\underset{\sim}{W}$ which diagonalizes a matrix $\underset{\sim}{P}$, totally symmetric with respect to the point-group operations,

$$\underset{\sim}{W}^{-1}\underset{\sim}{P}\underset{\sim}{W} = \underset{\sim}{P}_D, \tag{7}$$

produces eigenvectors that belong to definite symmetry types. We therefore compute the matrix of $\mathcal{P}$ over the set $S^{(L)}$ and diagonalize it by means of a unitary matrix $\underset{\sim}{W}$.* This transformation defines a symmetry-adapted set of functions $\underset{\sim}{F}^{(L)}$ through

$$\underset{\sim}{F}^{(L)} = \underset{\sim}{S}^{(L)}\underset{\sim}{W}. \tag{8}$$

These new functions $\underset{\sim}{F}^{(L)}$ do already transform irreducibly, but they are not yet in a sufficiently well-defined form to be suitable for subsequent applications.

In particular, we must first find out how many different symmetry species are spanned by the set $\underset{\sim}{F}$, and identify which functions belong to which species. To do this, we sort the $\underset{\sim}{F}$ first according to the degeneracies of their associated eigenvalues. We still must distinguish among species with the same degeneracy, but with different symmetry behavior. For this purpose, we choose a second totally symmetric operator, $\mathcal{P}'$, of the type of Eq. 5, but defined on a different set of points S', with matrix elements

$$P'_{\mu\nu} = \left\langle S_\mu^{(L)} \middle| \mathcal{P}' \middle| S_\nu^{(L)} \right\rangle. \tag{9}$$

In general, the matrix $\underset{\sim}{W}$ that diagonalized $\underset{\sim}{P}$ does not completely diagonalize $\underset{\sim}{P}'$, but produces a matrix,

*The choice of algorithm for diagonalization in this method is not a matter of indifference. If degeneracies are present, the Givens–Householder technique introduces mixing in the eigenvectors where there is none in the matrix, corresponding to an arbitrary rotation of the coordinate axes. The Jacobi method, for one, gives eigenvectors that retain the simple structure of the original matrix, and is therefore preferable.

$$\underset{\sim}{W}^{-1}\underset{\sim}{P}'\underset{\sim}{W} = \underset{\sim}{M}, \tag{10}$$

having the structure shown in Fig. 2. The nonzero off-diagonal elements of $\underset{\sim}{M}$ connect functions F_i belonging to the same symmetry species, so we can now assign a unique symmetry label to each function.

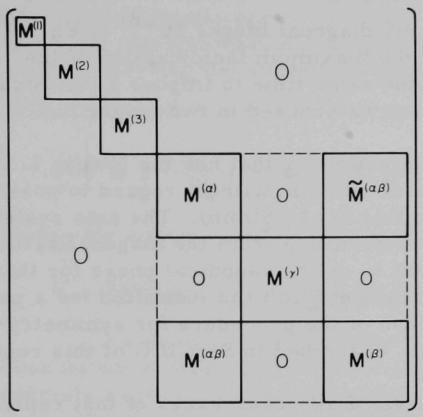

Fig. 2

Structure of Matrix $\underset{\sim}{M}$ from Eq. 10, Showing Off-diagonal Blocks Connecting Sets of Functions Belonging to the Same Symmetry Species. The blocks on the diagonal are themselves diagonal, with equal diagonal elements within a block. ANL Neg. No. 121-4451.

4. Case of Inversion Symmetry

A molecule possessing inversion symmetry will require that the above procedure be repeated for the ungerade representations. One then has two sets of functions, which together span all the irreducible representations of the point group. Since the symmetry behavior of functions in one set differs from that of corresponding members of the other set only in their inversion parity, it is useful to establish a correspondence between the two sets of symmetry labels that have been assigned. Any matrix element of P between members of the sets $\underset{\sim}{F}^{(L_g)}$ and $\underset{\sim}{F}^{(L_u)}$ must vanish because of the parity difference. However, if one first forms a new set of functions

$$Y_{L''M''} = \sum_{MM'} (L''M''|LML'M')Y_{LM}Y_{L'M'}, \tag{11}$$

where $L'' = L_u$, $L = L'$, and the quantities $(L''M''|LML'M')$ are the usual Clebsh-Gordan coefficients,[8] these new functions must perforce have even parity, as the product of two sets of like parity. Since the L'' and M'' values determine the behavior of the functions toward rotations and reflections, the coefficients $\underset{\sim}{W}$ that symmetry-adapt the real analogues of the $Y_{L''M''}$ are the same as those that symmetrize $\underset{\sim}{S}^{(L_u)}$. The nonzero matrix elements

$$\left\langle F_\mu^{(L'')} \middle| P \middle| F_\nu^{(L_g)} \right\rangle$$

then identify symmetrized functions differing only in their behavior toward inversion.

5. Imposition of a Standard Phase Choice

If all sets of functions belonging to the same symmetry species have the same phase, the individual off-diagonal blocks $M^{(\alpha\beta)}$ of Eq. 10 are also diagonal, and we have achieved the maximum factorization of the matrix. To accomplish this, and at the same time to impose a canonical phase choice on all symmetry vectors, we proceed in two stages.

First, we find the set of functions $\underset{\sim}{S}$ that has the lowest L value capable of spanning a particular representation, without regard to possible inversion-parity distinction. Denote this set by $\underset{\sim}{S}(\min)$. The axis system and phase evinced by the eigenvectors resulting from the diagonalization of the matrix of P over this set are taken as the canonical phase for that representation.[30] (The way in which the set $\underset{\sim}{S}(\min)$ is identified for a particular representation is a special case of the procedure for symmetry-adapting an arbitrary set of functions, described in Sec. II.C of this report.)

Second, we align the phases of all occurrences of that representation in the set $\underset{\sim}{F}$ with the canonical phase by means of a unitary transformation $\underset{\sim}{T}$, described in Sec. II.E. If inversion symmetry is present, the procedure is best elucidated through an example. Suppose that the set of spherical harmonics with L = 1 is the smallest set capable of spanning a representation labeled T_{1u} in the group under consideration. Then the phases of all occurrences of T_{1u} in the set $\underset{\sim}{F}^{(L_u)}$ are aligned with that of the L = 1 set. This phase choice is then also imposed on all occurrences of T_{1g} in $\underset{\sim}{F}^{(L_g)}$. The problem of parity is circumvented by using an A_{1u} operator to connect the two sets of functions: One identifies the A_{1u} function in the set $\underset{\sim}{F}^{(L_u)}$ and multiplies P by it, thereby forming a new odd-parity operator,

$$P_u = F(A_{1u}) \times P. \tag{12}$$

The off-diagonal blocks of matrix elements of P_u are made diagonal by the method given in Sec. II.E.

The transformations just described may be collected into a matrix $\underset{\sim}{T}$ which, when applied to the $\underset{\sim}{F}^{(L)}$, defines a new set of functions $\underset{\sim}{\psi}'^{(L)}$:

$$\underset{\sim}{\psi}'^{(L)} = \underset{\sim}{F}^{(L)}\underset{\sim}{T} = \underset{\sim}{S}^{(L)}\underset{\sim}{W}\underset{\sim}{T} \doteq \underset{\sim}{S}^{(L)}\underset{\sim}{V}'. \tag{13}$$

These functions now have all the well-defined symmetry behavior and phase choices that are desirable for use in calculations, and as such they could be used as our standard set of functions spanning all possible symmetry types in the point group. If the set $\underset{\sim}{\psi}'$ contains two or more representatives of a given symmetry species, however, these vectors will mix together in a way determined by the operators we have used, and hence in a way that bears no interpretive significance. To "undo" this arbitrary mixing, we may apply a final transformation $\underset{\sim}{B}$ which maximizes the number of zeroes in $\underset{\sim}{V}'$ while preserving the symmetry properties. The construction of $\underset{\sim}{B}$ is described in detail in Ref. 26. The functions $\underset{\sim}{\psi}^{(L)}$,

$$\underset{\sim}{\psi}^{(L)} = \underset{\sim}{\psi}'^{(L)}\underset{\sim}{B} = \underset{\sim}{S}^{(L)}\underset{\sim}{V}'\underset{\sim}{B} = \underset{\sim}{S}^{(L)}\underset{\sim}{V}, \tag{14}$$

comprise the standard set, called <u>prototype functions</u>.

6. Example: The Octahedral Group O_h

As an example of the prototype functions and the standard phase choices we have implemented, we consider the octahedral group O_h. The irreducible representations of O_h and their degeneracies are indicated in Table I. The lowest L values that span all the even- and odd-parity representations are $L_g = 6$ and $L_u = 9$, respectively. If $\Gamma(L = 6)$ and $\Gamma(L = 9)$ are the corresponding representations, then under the operations making up the group O_h they reduce as follows:

$$\left.\begin{aligned} \Gamma(L = 6) &\rightarrow A_{1g} + A_{2g} + E_g + T_{1g} + 2T_{2g}; \\ \Gamma(L = 9) &\rightarrow A_{1u} + A_{2u} + E_u + 3T_{1u} + 2T_{2u}. \end{aligned}\right\} \tag{15}$$

TABLE I. Symmetry Species of the Point Group O_h

Symbol		
Even Parity	Odd Parity	Degeneracy
A_{1g}	A_{1u}	1
A_{2g}	A_{2u}	1
E_g	E_u	2
T_{1g}	T_{1u}	3
T_{2g}	T_{2u}	3

Figures 3 and 4 show the simplest representatives of each symmetry species in O_h, derived from real spherical harmonics, and indicate the standard phase choices against which the prototype functions are aligned.

The behavior of the symmetry-adapted prototype functions for $L = L_g$ and $L = L_u$ is shown in Figs. 5 and 6. These figures were generated and plotted by the computer as a program option, and illustrate a type of incidental application for which the program might be useful. The transformation coefficients making up the matrix $\underset{\sim}{V}$ in Eq. 14 are real combinations of those given in Appendix 2 of Ref. 11.

C. Symmetry-adaptation of an Arbitrary Closed Set of Functions

1. Allowed Function Classes

We have placed a good deal of emphasis thus far on the construction of the prototype functions and their phases. This effort was well spent, however, since it now becomes an easy task to symmetry-adapt virtually any set of functions associated with the molecule, that is, to find the reduction matrix $\underset{\sim}{U}$ of Eq. 4. For this to be possible, it is usually necessary to

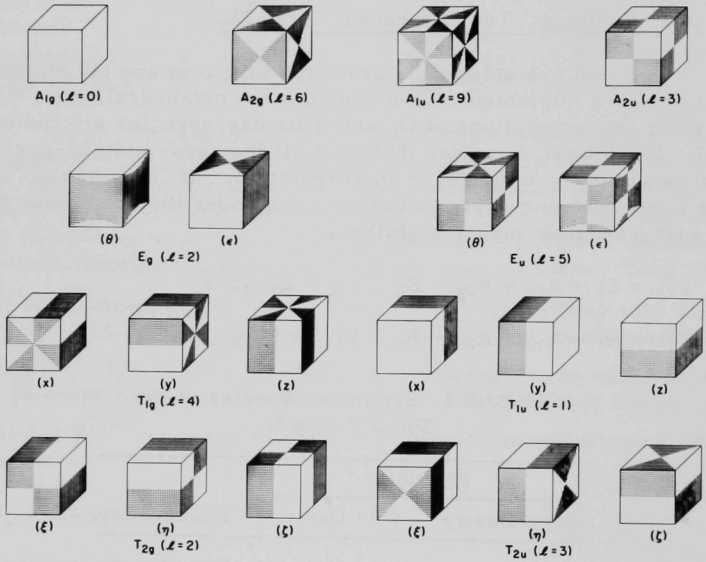

Fig. 3

Computer Plots of Real, Symmetry-adapted Spherical Harmonics Centered at the Origin Giving the Simplest Occurrence of Each Even-parity Representation of O_h. The functions are evaluated on the surface of a unit cube, and the shaded areas represent negative function values. Subspecies labels are those used in Ref. 11. ANL Neg. No. 121-4440.

Fig. 4

Computer Plots of the Simplest Occurrence of Each Odd-parity Irreducible Representation of O_h. ANL Neg. No. 121-4438.

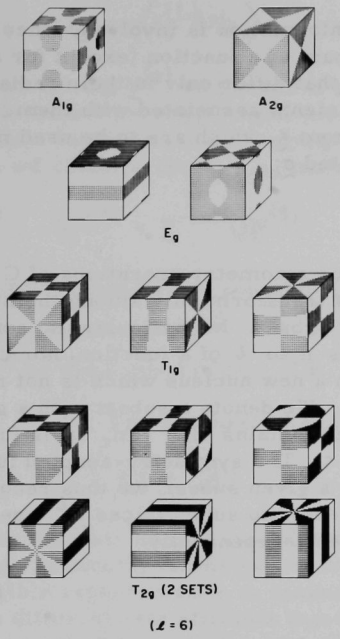

Fig. 5

Symmetry-adapted Prototype Functions (L = 6) Spanning All the Even-parity Representations of O_h. ANL Neg. No. 121-4439.

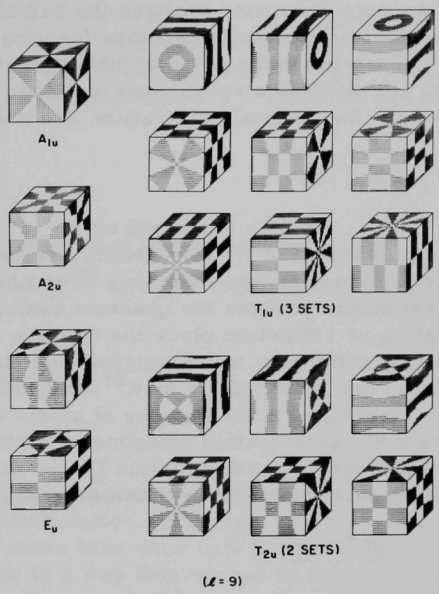

Fig. 6

Symmetry-adapted Prototype Functions (L = 9) Spanning All the Odd-parity Representations of O_h. ANL Neg. No. 121-4437.

require only that the set be closed, in the sense that Eq. 1 is valid. These functions may be of various types: For example, in an ab initio calculation of electronic properties, they may be atomic basis functions centered on the individual atoms. For problems in molecular dynamics, they may be the Cartesian displacements of the nuclei. To illustrate an important case, we will carry through the example of determining a symmetry-adapted atomic basis set.

2. Example: Atomic Basis Set

Let $\underset{\sim}{X}$ be the set of all atomic orbital basis functions to be used in a subsequent calculation; a typical member is $\chi^{A}_{n\ell m}(R_A, \theta_A, \varphi_A)$, whose quantum numbers are n, ℓ, and m, and whose origin is at atom A. We may choose the functions to be real without infringing on the generality of the method. From $\underset{\sim}{X}$ we may extract a smaller set $\underset{\sim}{\chi}$ which contains all values of the angular indices, ℓ and m, represented in $\underset{\sim}{X}$, but with only one representative radial part for each ℓ value within a class of symmetry-related

atoms, or atom type.* Again, no loss of information is involved, since no point-group operation changes the radial part of a function (except for a change of origin), and therefore functions that differ only in their radial parts have the same transformation coefficients associated with them. The final set of symmetry-adapted functions from $\underset{\sim}{\chi}$, which are to be used in a quantum-mechanical calculation, are denoted $\underset{\sim}{\sigma}$:

$$\underset{\sim}{\sigma} = \underset{\sim}{\chi}\underset{\sim}{U}. \tag{16}$$

As $\underset{\sim}{\chi}$ is subjected to the various symmetry operations of G, certain subsets of its constituent functions transform only among themselves and never into members of another subset. In particular, no rotation or reflection changes the quantum numbers n or ℓ of a function, nor can a rotation or reflection place the function on a new nucleus which is not related by symmetry to the original nucleus. We denote a subset with a given ℓ-value and atom type τ by $\chi^{\ell\tau}$; the subset contains $(2\ell+1)\, n_\tau$ basis functions, where n_τ is the number of atoms in τ. The symmetry-adapted functions we desire contain components within a given subset; we thus recognize that the transformation matrix $\underset{\sim}{U}$ also factors into submatrices $\underset{\sim}{U}^{\ell\tau}$, permitting each subset to be symmetry-adapted independently:

$$\underset{\sim}{\sigma}^{\ell\tau} = \underset{\sim}{\chi}^{\ell\tau}\underset{\sim}{U}^{\ell\tau}. \tag{17}$$

Each subset in $\underset{\sim}{\chi}$ is treated the same way; in the remainder of this section, therefore, the superscripts will be omitted for brevity. Further, since the actual radial dependence factors out of the symmetry transformation, we ignore it and replace it by $R(r) = r^{-1}$ for ease of computation; the only requirement on $R(r)$ is that it be sufficiently distinctive to avoid accidental degeneracies.

3. Symmetrization Procedure

The initial stages of symmetry-adaptation proceed as outlined previously: the matrix of P over the subset $\underset{\sim}{\chi}$ is calculated and diagonalized by a transformation $\underset{\sim}{W}$. A new set of functions $\underset{\sim}{\rho}$ is formed from $\underset{\sim}{\chi}$,

$$\underset{\sim}{\rho} = \underset{\sim}{\chi}\underset{\sim}{W}. \tag{18}$$

To determine what representations are contained in $\underset{\sim}{\rho}$ and to line up phases, we reintroduce the real harmonics $\underset{\sim}{S}^{(L)}$ and calculate a connection matrix $\underset{\sim}{Q}$,

---

*Operationally, we define an atom type to include all atoms of a given chemical element that are "equidistant" from the center of nuclear charge. According to the looseness of the criterion applied to determine "equidistance," a user may optionally include atoms that are not strictly related by a symmetry operation. In studies of small distortions from an equilibrium geometry, this option allows one to avoid abrupt changes of symmetry adaptation as the nuclear geometry changes continuously.

$$Q_{ij} = \left\langle S_i^{(L)} \middle| P \middle| \chi_j \right\rangle . \tag{19}$$

Then, letting $\underset{\sim}{V}^{(\alpha)}$ be the set of prototype transformation vectors associated with the αth prototype eigenvalue, of degeneracy n_α, and $\underset{\sim}{W}^{(\beta)}$ the set of atomic basis-function eigenvectors from eigenvalue β, of degeneracy $n_\beta = n_\alpha$, we compute the $n_\alpha \times n_\alpha$ matrix

$$\underset{\sim}{M}^{(\alpha\beta)} = \underset{\sim}{V}^{-1(\alpha)} \underset{\sim}{Q} \underset{\sim}{W}^{(\beta)} . \tag{20}$$

If $\underset{\sim}{M}^{(\alpha\beta)}$ contains any nonzero elements, then $\underset{\sim}{W}^{(\beta)}$ carries the same symmetry species as $\underset{\sim}{V}^{(\alpha)}$. The phase of $\underset{\sim}{W}^{(\beta)}$ is made to correspond to that of $V^{(\alpha)}$ by the method of Sec. II.E.

The new basis functions $\underset{\sim}{\sigma}'$,

$$\underset{\sim}{\sigma}' = \underset{\sim}{P} \underset{\sim}{T} = \underset{\sim}{\chi} \underset{\sim}{W} \underset{\sim}{T} = \underset{\sim}{\chi} \underset{\sim}{U}' , \tag{21}$$

have the well-defined transformation properties necessary to induce maximum symmetry factorization in a calculation. However, if the same irreducible representation is spanned more than once in a set of functions $\underset{\sim}{\sigma}'$, the different sets still mix together in a way determined by the operator P. The method described in Ref. 26 determines a final transformation $\underset{\sim}{B}'$, which produces a simple set of coefficients in $\underset{\sim}{U}$:

$$\underset{\sim}{\sigma} = \underset{\sim}{\sigma}' \underset{\sim}{B}' = \underset{\sim}{\chi} \underset{\sim}{U}' \underset{\sim}{B}' = \underset{\sim}{\chi} \underset{\sim}{U} . \tag{22}$$

The procedure is repeated for each set $\underset{\sim}{\chi}^{\ell\tau}$.

D. Finite Point-group Vector-coupling Coefficients

1. Definition and Properties

Let $\underset{\sim}{F}_\Gamma$ and $\underset{\sim}{G}_{\Gamma'}$ be two sets of functions forming bases for the irreducible representations Γ and Γ', respectively, of the point group. Consider a new set $\underset{\sim}{H}$ with $n_\Gamma \times n_{\Gamma'}$ members formed by taking all possible products of pairs $F_{\Gamma\gamma} \times G_{\Gamma'\gamma'}$, where γ and γ' label the different subspecies in Γ and Γ'. The elements in $\underset{\sim}{H}$ provide a new basis set for representations of the point group. A representation Γ'' is contained in the direct product $\Gamma \times \Gamma'$, if symmetry-adapted functions $H_{\Gamma''\gamma''}$ can be constructed from $\underset{\sim}{H}$ by means of a suitable unitary transformation:[12]

$$H_{\Gamma''\gamma''} = \sum_\gamma \sum_{\gamma'} (\Gamma''\gamma'' | \Gamma\gamma\Gamma'\gamma') F_{\Gamma\gamma} G_{\Gamma'\gamma'} . \tag{23}$$

The coefficients $(\Gamma''\gamma''|\Gamma\gamma\Gamma'\gamma')$ in Eq. 23 are called vector-coupling coefficients. In the full rotation group, these coefficients occur in the coupling of two angular momenta to form a third:

$$Y_{\ell''m''} = \sum_m \sum_{m'} (\ell''m''|\ell m\ell' m')Y_{\ell m}Y_{\ell' m'}. \tag{24}$$

In this case, the coefficients are also called Clebsh-Gordan coefficients.[8] The vector-coupling coefficients are defined to be orthonormal in the sense:

$$\left.\begin{aligned} &\sum_\gamma \sum_{\gamma'} (\Gamma_1\gamma_1|\Gamma\gamma\Gamma'\gamma')(\Gamma\gamma\Gamma'\gamma'|\Gamma_2\gamma_2) = \delta_{\Gamma_1\Gamma_2}\delta_{\gamma_1\gamma_2}\delta(\Gamma,\Gamma',\Gamma_1); \\ &\sum_{\Gamma''} \sum_{\gamma''} (\Gamma\gamma_1\Gamma'\gamma_1'|\Gamma''\gamma'')(\Gamma''\gamma''|\Gamma\gamma_2\Gamma'\gamma_2') = \delta_{\gamma_1\gamma_2}\delta_{\gamma_1'\gamma_2'}; \end{aligned}\right\} \tag{25}$$

where $\delta(\Gamma,\Gamma',\Gamma_1)$ is unity if $\Gamma \times \Gamma'$ contains Γ_1, and is zero otherwise. Because vector-coupling coefficients completely describe the geometric relationships between various basis functions, any quantum-mechanical matrix element can be factored into a part which is determined by only geometric properties and a second part which depends on rotationally invariant properties of the particular wave functions and operator:

$$\left\langle F_{\alpha''\Gamma''\gamma''}\middle|A_{\Gamma\gamma}\middle|G_{\alpha'\Gamma'\gamma'}\right\rangle = (\Gamma''\gamma''|\Gamma\gamma\Gamma'\gamma')\left\langle F_{\alpha''\Gamma''}\middle\|A_\Gamma\middle\|G_{\alpha'\Gamma'}\right\rangle. \tag{26}$$

The second factor on the right is called the <u>reduced matrix element</u>, and the theorem that assures this factorization is called the Wigner-Eckart theorem.[12] If Γ, Γ', and Γ'' are degenerate irreducible representations, considerable labor may be saved in calculating matrix elements among the members of the degenerate sets by calculating the reduced matrix element only once.

2. Method of Calculation

To compute the vector-coupling coefficients for finite point groups, we use the fact that the real spherical harmonics in terms of which we expand prototype functions for the finite group are themselves symmetry-adapted to the full rotation group. In showing the derivation, we find it convenient to revert to the complex functions Y_{LM}, which are then related to the prototype functions $\underset{\sim}{\psi}^{(L)}$ of Eq. 14 by a complex transformation $\underset{\sim}{C}^{(L)}$:

$$\underset{\sim}{\psi}^{(L)} = \underset{\sim}{Y}^{(L)}\underset{\sim}{C}^{(L)}, \tag{27}$$

or

$$\psi^{(L)}_{\alpha\Gamma\gamma} = \sum_{M=-L}^{L} Y_{LM} C^{\alpha\Gamma\gamma}_{LM}. \tag{28}$$

The coefficients $C^{\alpha\Gamma\gamma}_{LM}$ are related to the elements of the matrix $\underset{\sim}{V}$ of Eq. 14 in the following way: First, let the matrix $\underset{\sim}{C}$ be written

$$\underset{\sim}{C} = \underset{\sim}{A} + i\underset{\sim}{B},$$

where $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are real matrices. Then,

$$\left.\begin{aligned} A^{\alpha\Gamma\gamma}_{L0} &= V^{\alpha\Gamma\gamma}_{L1}; & B^{\alpha\Gamma\gamma}_{L0} &= 0; \\ A^{\alpha\Gamma\gamma}_{LM} &= \frac{(-1)^M}{\sqrt{2}} V^{\alpha\Gamma\gamma}_{L,2M+1}; & B^{\alpha\Gamma\gamma}_{LM} &= \frac{(-1)^M}{\sqrt{2}} V^{\alpha\Gamma\gamma}_{L,2M}; \\ A^{\alpha\Gamma\gamma}_{L,-M} &= (-1)^M A^{\alpha\Gamma\gamma}_{LM}; & B^{\alpha\Gamma\gamma}_{L,-M} &= (-1)^{M+1} B^{\alpha\Gamma\gamma}_{LM}. \end{aligned}\right\} \tag{29}$$

Now consider the matrix element

$$\left\langle \psi^{(L'')}_{\alpha''\Gamma''\gamma''} \middle| \psi^{(L')}_{\alpha'\Gamma'\gamma'} \middle| \psi^{(L)}_{\alpha\Gamma\gamma} \right\rangle.$$

In view of Eq. 26, this can be written

$$\left\langle \psi^{(L'')}_{\alpha''\Gamma''\gamma''} \middle| \psi^{(L')}_{\alpha'\Gamma'\gamma'} \middle| \psi^{(L)}_{\alpha\Gamma\gamma} \right\rangle = (\Gamma''\gamma'' | \Gamma\gamma\Gamma'\gamma') \left\langle \psi^{(L'')}_{\alpha''\Gamma''} \middle\| \psi^{(L')}_{\alpha'\Gamma'} \middle\| \psi^{(L)}_{\alpha\Gamma} \right\rangle. \tag{30}$$

By Eq. 28, however, we can also write

$$\left\langle \psi^{(L'')}_{\alpha''\Gamma''\gamma''} \middle| \psi^{(L')}_{\alpha'\Gamma'\gamma'} \middle| \psi^{(L)}_{\alpha\Gamma\gamma} \right\rangle =$$

$$\sum_{M''} \sum_{M'} \sum_{M} C^{*\alpha''\Gamma''\gamma''}_{L''M''} C^{\alpha'\Gamma'\gamma'}_{L'M'} C^{\alpha\Gamma\gamma}_{LM} \left\langle Y_{L''M''} \middle| Y_{L'M'} \middle| Y_{LM} \right\rangle. \tag{31}$$

Equating these two expressions and applying the Wigner-Eckart theorem to the matrix element over spherical harmonics, we obtain the final expression:

$$(\Gamma''\gamma''|\Gamma\gamma\Gamma'\gamma') = \frac{(-1)^{L''}\left\langle Y_{L''}\middle\|Y_{L'}\middle\|Y_{L}\right\rangle}{\left\langle \psi^{(L'')}_{\alpha''\Gamma''}\middle\|\psi^{(L')}_{\alpha'\Gamma'}\middle\|\psi^{(L)}_{\alpha\Gamma}\right\rangle}$$

$$\sum_{M'}\sum_{M} C^{*\alpha''\Gamma''\gamma''}_{L'',M+M'} C^{\alpha'\Gamma'\gamma'}_{L'M'} C^{\alpha\Gamma\gamma}_{LM} (-1)^{M+M'} \begin{pmatrix} L & L' & L'' \\ M & M' & -M-M' \end{pmatrix}. \tag{32}$$

The quantity in front of the summation signs in Eq. 32 is not calculable directly in our method. Its magnitude is determined, however, by the condition of Eq. 25 on the vector-coupling coefficients. We are left with an arbitrary choice of sign for each trio of representations $(\Gamma, \Gamma', \Gamma'')$. In the absence of a well-defined means of imposing other sign choices, we choose the quantity

$$\frac{(-1)^{L''}\left\langle Y_{L''}\middle\|Y_{L'}\middle\|Y_{L}\right\rangle}{\left\langle \psi^{(L'')}_{\alpha''\Gamma''}\middle\|\psi^{(L')}_{\alpha'\Gamma'}\middle\|\psi^{(L)}_{\alpha\Gamma}\right\rangle}$$

to be <u>positive</u> for all representations and prototype L values. Since the vector-coupling coefficients are not dependent upon the inversion parity of the representations involved, we may take $L = L' = L'' = L_g$, the value that spans all the even representations of the point group.

3. Antisymmetric Direct Products

Equation 32 fails for some combinations $(\Gamma, \Gamma', \Gamma'')$ if all three sets of spherical harmonics have even L values, because of the behavior of the 3-j symbols under permutation of L values. This condition arises when two of the Γ's are the same and the third belongs to the set of antisymmetric squares (see, e.g., Ref. 11). The summation may be made not to vanish if one of the sets of Y_{LM} is antisymmetric. Let the functions $Y_{L''M''}$, Y_{LM}, $Y_{L'M'}$ of Eq. 24 be chosen so that $L = L' = L_g$, and $L'' = L_u$. The functions $Y_{L''M''}$ are then not usual spherical harmonics, because they are antisymmetric with respect to interchange of L and L', have <u>even</u> inversion parity (see Eq. 11), and in fact are multiplied throughout by $i \equiv \sqrt{-1}$. The coefficients $C^{\alpha''\Gamma''\gamma''}_{L''M''}$ that symmetry-adapt the $Y_{L''M''}$ can be inserted into Eq. 32 to obtain the desired antisymmetric vector-coupling coefficients.

4. Extensions and Uses

Vector-coupling coefficients are produced by the method described in this report for any finite point group, even for one such as the icosahedral group which has two linearly independent sets of coefficients for certain combinations of irreducible representations.[11] With proper

attention to phase choices, it should be possible also to generate the V, W, and X coefficients of Griffith,[12] which are the respective analogues of the Wigner 3-j, 6-j, and 9-j symbols.[8]

E. Phase Alignment of Sets of Functions Carrying the Same Symmetry Species

Let $\underset{\sim}{V}^{(\alpha)}$ and $\underset{\sim}{W}^{(\beta)}$ be the sets of transformation vectors associated with eigenvalues α, β, respectively, of degeneracy $n_\alpha = n_\beta$. Here $\underset{\sim}{V}^{(\alpha)}$ and $\underset{\sim}{W}^{(\beta)}$ need not arise from the same set of functions, but $\underset{\sim}{V}^{(\alpha)}$ is assumed to have the standard phase choice. We seek a unitary transformation $\underset{\sim}{T}^{(\alpha\beta)}$,

$$\underset{\sim}{W}'^{(\beta)} = \underset{\sim}{W}^{(\beta)}\underset{\sim}{T}^{(\alpha\beta)}, \tag{33}$$

such that

$$\underset{\sim}{M}^{(\alpha\beta)}\underset{\sim}{T}^{(\alpha\beta)} = \underset{\sim}{M}_D^{(\alpha\beta)}, \tag{34}$$

where $\underset{\sim}{M}^{(\alpha\beta)}$ is the matrix connecting $\underset{\sim}{V}^{(\alpha)}$ and $\underset{\sim}{W}^{(\beta)}$, and $\underset{\sim}{M}_D^{(\alpha\beta)}$ is diagonal, with equal diagonal elements λ. $\underset{\sim}{M}^{(\alpha\beta)}$ is defined by

$$\underset{\sim}{M}^{(\alpha\beta)} = \underset{\sim}{V}^{-1(\alpha)}\underset{\sim}{P}\underset{\sim}{W}^{(\beta)}, \tag{35}$$

where $\underset{\sim}{P}$ is the matrix of a totally symmetric operator over the basis functions represented in $\underset{\sim}{V}^{(\alpha)}$ and $\underset{\sim}{W}^{(\beta)}$.

Suppressing the superscripts and rewriting, we obtain

$$\underset{\sim}{M} = \underset{\sim}{M}_D\underset{\sim}{T}^{-1}, \tag{36}$$

or

$$m_{ij} = \sum_k \lambda_{ik}\delta_{ik}t_{kj}^{-1} = \lambda t_{ij}^{-1}. \tag{37}$$

Since $\underset{\sim}{T}$ is unitary, we may write

$$\sum_j |m_{ij}|^2 = \lambda^2 \sum_j |t_{ij}^{-1}|^2 = \lambda^2 \cdot 1. \tag{38}$$

Then

$$\lambda = \sqrt{\sum_i |m_{ij}|^2},$$

and $t_{ij}^{-1} = t_{ji} = m_{ij}^{*}/\lambda$. These small transformations $\underset{\sim}{T}^{(\alpha\beta)}$ are all that need be calculated, although formally they may be collected in one unitary transformation $\underset{\sim}{T}$ applied to the entire matrix $\underset{\sim}{W}$.

III. PROGRAM DESCRIPTIONS AND ALGORITHMS

A. Design Philosophy

The program described in this report (which we call GPTHEORY) is designed with a view to interfacing with any of several types of applications programs, as well as being a "stand-alone" program in and of itself. To facilitate interfacing, the output from the program is organized into block-common storage areas whose contents may optionally be dumped onto an external storage device. Input to the program is purposely kept to a bare minimum, namely, the number and types of atoms, their Cartesian coordinates in some convenient axis system, and "something" that characterizes the basis set. The latter may be as little as a control integer, since the program contains several prestored Slater-type atomic basis sets, from which it will extract the necessary information. Section IV of this report contains complete descriptions of the input and the output.

A potential user of this program may well envision applications that have not been specifically provided for, and may therefore need to modify the structure of the program. Hence, we have tried to make the package as modular as possible, well-defined tasks being relegated to subroutines. Furthermore, "sockets" have been provided for the user to "plug in" his own applications module with minimum trauma.

Extensive matrix manipulations and inner products, such as are used in this method, tend to involve accumulation of roundoff errors that can seriously degrade the discrimination of "zero" and "nonzero" quantities vital to our approach. This problem is handled quite well by the introduction of threshold parameters: Matrix elements and certain other quantities are compared with the zero threshold at frequent stages in the computation, and set equal to zero if they fall below. In this way, cumulative errors are minimized, and up to 15-digit accuracy is maintained in the transformation coefficients.

A good deal of intermediate printout, which is suppressed under normal operation, may be invoked as a diagnostic tool in case of trouble. Indeed, if termination occurs under conditions allowing the program to remain in control, a restart is made with intermediate output to aid in troubleshooting.

B. Logic Flow

The program is divided into several sections that are logically almost independent of one another. Overall control is therefore vested in a

short driver program, MAIN, which calls the appropriate segments, controls restarting in case of program-detected errors, and provides an interface to applications programs.

Figures 7-11 show flowcharts of the several major parts of the program. This version of the program is not organized into an overlay structure; if space is a limitation, however, the control sections (subroutines and COMMON blocks) can be put into overlays as shown in Fig. 12. (The tree structure will, of course, differ somewhat for different machines: It is shown here for the IBM 360 Linkage Editor.)

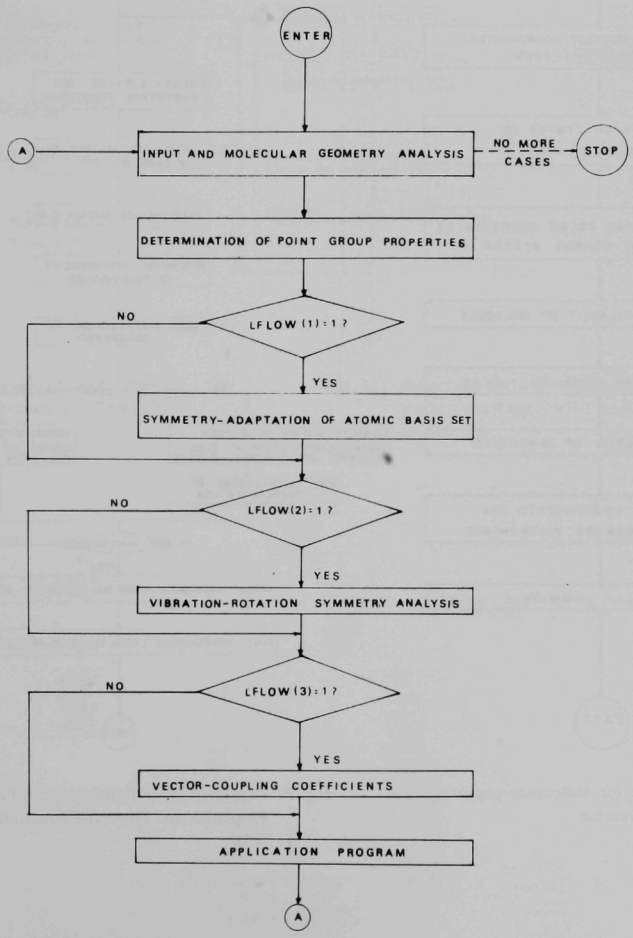

Fig. 7. Flowchart of Overall Logic Flow

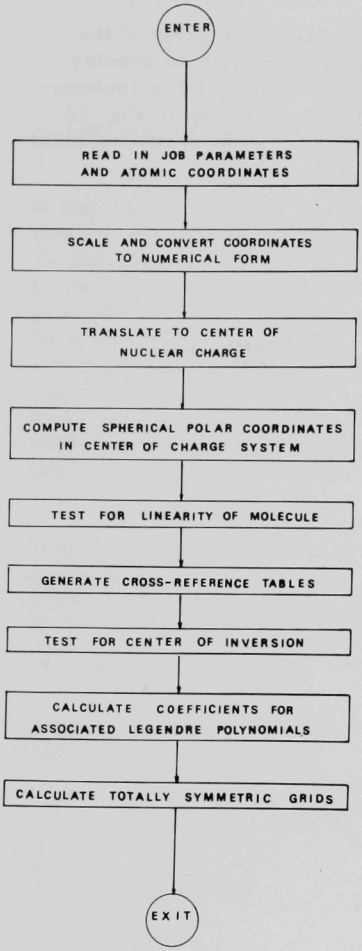

Fig. 8. Flowchart for Molecular-geometry Analysis Section

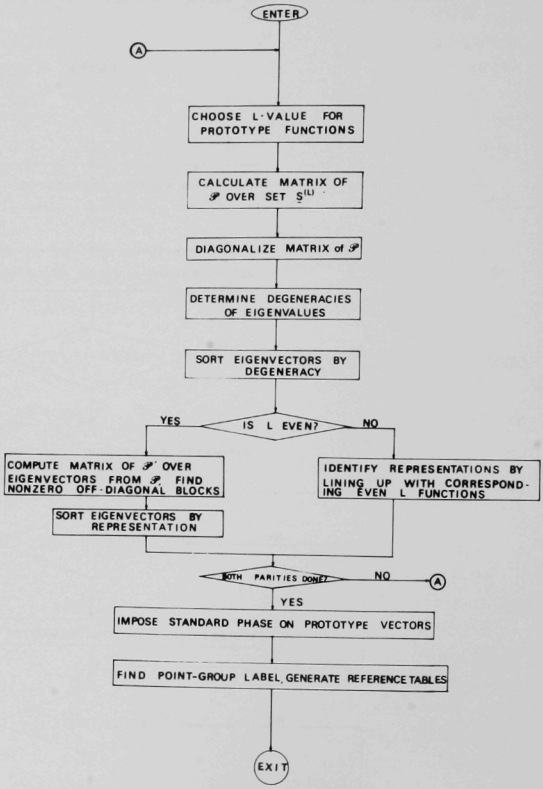

Fig. 9. Flowchart for Determination of Point-group Properties and Prototype Functions

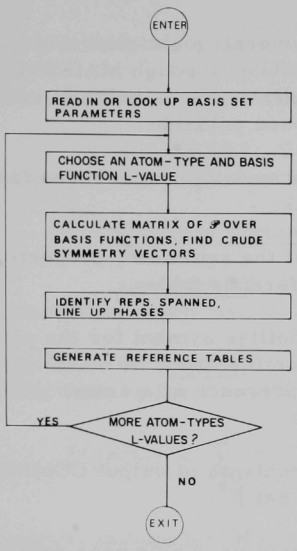

Fig. 10. Flowchart for Symmetry-adaptation of Atomic Basis Set

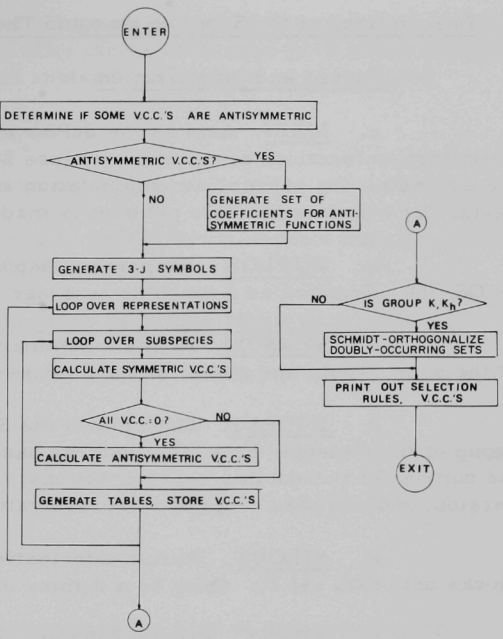

Fig. 11. Flowchart for Calculation of Vector-coupling Coefficients (V.C.C.'s)

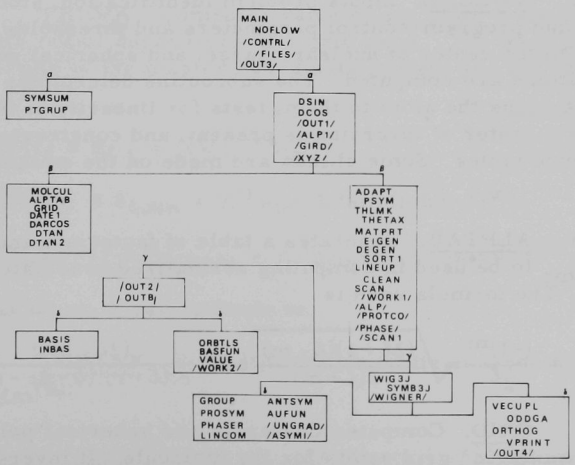

Fig. 12. A Possible Overlay Structure for the GPTHEORY Program

C. Descriptions of the Subprograms and Their Algorithms

1. Driver and Initialization

a. MAIN. Acts as the driver and overall logic controller. Initialization functions are performed (see MAIN0060 through MAIN0140 in the appendix), the appropriate computation segments are called, and error restarting with intermediate printout is made when possible.

b. NOFLOW. Suppresses exponent underflow messages issued by OS/360. Supplied as a machine language deck.

c. SYMSUM. Prints a summary of the symmetry properties of the point group, and stores several cross-reference tables.

d. PTGRUP. Determines the Schönflies symbol for the point group of the molecule by a table look-up based on the order of the group, the number of irreducible representations, the presence of a center of inversion, and complex conjugate representations.

e. FILOUT. Dumps unformatted contents of output COMMON blocks onto data set 20. (May be a dummy data set.)

f. APPLY. Provides dummy subroutine, which user replaces with his own application.

2. Input and Molecular-geometry Analysis

a. MOLCUL. Inputs problem identification, atomic Cartesian coordinates, and program-control parameters and thresholds. The origin is translated to the center of nuclear charge, and spherical polar coordinates of the atoms are computed. The subroutine determines the distinct atom-types, assigns the atom to them, tests for linearity of the molecule, determines if a center of inversion is present, and constructs several cross-reference tables. Some checks are made on the suitability of the input.

b. ALPTAB. Generates a table of factorials and a set of coefficients $C_{\ell mr}$ to be used in computing normalized associated Legendre polynomials. The formula used is

$$C_{\ell mr} = \frac{(-1)^m}{2^\ell} \sqrt{\frac{(2\ell+1)(\ell-m)!}{2(\ell+m)!}} (-1)^r \frac{(2\ell-2r)!}{r!(\ell-r)!(\ell-2r-m)!}.$$

c. GRID. Computes Cartesian and spherical polar coordinates of "totally symmetric" grid points for the molecule. If inversion symmetry is present, the grid is made inversion-symmetric, point for point. The grid

consists of NDIV equally spaced points on each circle centered on the lines joining each atom to the origin. The circle is located 9/10 of the distance from the origin to the atom, and subtends an angle ω at the origin

$$\omega = 2 \tan^{-1}[(\tan 4°)(1 - Z/150)],$$

where Z is the nuclear charge on the atom.

d. DATE1. Consists of the OS/360 assembly language routine that prints out the current calendar date.

3. Routines Common to More Than One Major Segment

a. ADAPT. Supervises symmetry-adaptation of arbitrary closed set of functions.

b. PSYM. Calculates matrix of P over either one or two sets of functions by the formula

$$\langle F_i | P | G_j \rangle = \sum_{\nu \in S} F_i(\underset{\sim}{r}_\nu) G_j(\underset{\sim}{r}_\nu).$$

A control parameter, KALC, determines the kind of computation: If KALC < 0, F_i and G_j are assumed to come from two different sets of functions, otherwise from the same set; if $|\text{KALC}| = 9$, a center of inversion is present and the matrix element is evaluated over only one atom of each inversion-symmetric pair.

c. THLMK. Computes value of set of 2L + 1 real spherical harmonics with radial dependence at a grid point. The functions are centered at the origin of the master coordinate system and are defined as follows for $m \geq 1$:

$$\text{FUNCT}(1) \equiv S_{\ell,1} = (r\sqrt{2})^{-1}\, \Theta_{\ell 0}(\theta);$$

$$\text{FUNCT}(2m) \equiv S_{\ell,2m} = r^{-1}\, \Theta_{\ell m}(\theta) \sin(m\varphi)(-1)^m;$$

$$\text{FUNCT}(2m+1) \equiv S_{\ell,2m+1} = r^{-1}\, \Theta_{\ell m}(\theta) \cos(m\varphi)(-1)^m.$$

For each value of m, a call is made to

d. THETAX. Evaluates the normalized associated Legendre polynomial $\Theta_{\ell m}(\theta)$:

$$\Theta_{\ell m}(\theta) = \sin^m \theta \sum_{r=0}^{\nu} C_{\ell m r} (\cos\theta)^{\ell - m - 2r}, \quad \nu = \text{int}\left(\frac{\ell - m}{2}\right),$$

where the $C_{\ell mr}$ were defined in subroutine ALPTAB. Special formulas are used for the special cases of $\sin\theta = 0$ and $\cos\theta = 0$.

e. MATPRT. Is a matrix output routine for square matrices of arbitrary size. The matrices are printed out six columns at a time, with an appropriate alphanumeric label for each row. If eigenvalues are associated with particular columns of the matrix, they are also output.

f. EIGEN. Diagonalizes a real, symmetric matrix using the Jacobi method. This method is used because no mixing is introduced among vectors belonging to the same degenerate eigenvalue if they are not connected in the original matrix. The eigenvectors are arranged in order of descending eigenvalue.

g. DEGEN. Determines the number of distinct eigenvalues and their degeneracies. Two eigenvalues are considered to be degenerate if the absolute value of their ratio differs from unity by less than CRTDEG (q.v.).

h. SORT1. Sorts eigenvalues and eigenvectors in ascending order of elements in an integer array associated with the matrix. Thus, eigenvectors may be sorted according to representation index, degeneracy, etc.

i. LINEUP. Determines whether two different sets of functions belong to the same irreducible representation, and, if so, rotates one set to align its phase with that of the other set. The phase transformation is carried out by the method of Sec. II.E.

j. CLEAN. Sorts out arbitrary mixing of multiply-occurring representatives of a given symmetry species, to give coefficients that are more easily interpretable. For each symmetry species, the number of sets of vectors belonging to it is determined, and a call is made to:

k. SCAN. Determines the member of a set with the smallest number of nonzero coefficients, and identifies the coefficients within a vector that differ at most by a sign.

l. WIG3J. Computes set of Wigner 3-j symbols for given ℓ values. Current version assumes $\ell_1 = \ell_2 =$ even. The formula used is given on pp. 45-46 of Ref. 8.

m. SYMB3J. When ℓ_3 is odd, assigns the correct phase to the 3-j symbol.

4. Determination of Point-group Properties

a. GROUP. Is the supervisor for this section. On the basis of results of a preliminary call to PROSYM (q.v.) with L = 4, determines

lowest L-values of spherical harmonics spanning all irreducible representations of even (and odd) parity for the particular nuclear configuration. Calls PROSYM to obtain prototype functions, then PHASER to impose standard phase choice. Table II shows the L-values selected for each point group.

TABLE II. L-values Used in the Prototype Functions for the Various Point Groups

L	Groups
0	(C_1)
1	C_{2h}, C_i
2	D_2, C_{2v}, D_{2h}, C_i, C_s, C_2
3	C_{4h}, D_{3d}, D_{2h}, C_{6h}, S_6
4	D_{2d}, C_3, D_3, C_{3v}, D_{3d}, C_{3h}, C_4, D_4, S_4, C_{4v}, C_{4h}, D_{4h}, C_6, S_6, C_{6h}, C_7, C_8, S_8
5	D_{4h}, D_{5d}
6	D_{3h}, C_5, D_5, C_{5v}, C_{5h}, D_6, D_{6h}, D_{6v}, T, T_h, T_d, O, O_h
7	-
8	D_{4d}, D_{8h}
9	D_{6h}, D_{8h}, O_h, T_h
10	D_{5h}
11	-
12	D_{6d}, K, K_h
13	-
14	-
15	K_h
16	D_{7h}, D_{8d}

b. PROSYM. Supervises determination of a symmetry-adapted set of spherical harmonics.

c. ANTSYM. Generates a set of even-parity, odd-ℓ spherical harmonics to be used in establishing the correspondence of the odd-parity (odd-ℓ) symmetry vectors with the even-parity (even-ℓ) ones.

The formulas used are as follows: Let us write $Y_{LM} = C_{LM} + iS_{LM}$. The functions C_{LM} and S_{LM} are then related to the $S_{L,K}$ by a transformation like that defined by Eqs. 29. The antisymmetric functions $\hat{S}_{L',K}$ are then given by

$$\hat{S}_{L',1} = \sum_{M_1=1}^{L} (-1)^{M_1+1}\left[S^1_{LM_1}C^2_{LM_1} - C^1_{LM_1}S^2_{LM_1}\right]\begin{pmatrix} L & L & L' \\ M_1 & -M_1 & 0 \end{pmatrix};$$

$$\hat{S}_{L',2M_3} = -\frac{1}{\sqrt{2}}\sum_{M_1=-L}^{L}\left[S^1_{LM_1}S^2_{LM_2} - C^1_{LM_1}C^2_{LM_2}\right]\begin{pmatrix} L & L & L' \\ M_1 & M_2 & -M_3 \end{pmatrix};$$

$$\hat{S}_{L',2M_3+1} = -\frac{1}{\sqrt{2}}\sum_{M_1=-L}^{L}\left[C^1_{LM_1}S^2_{LM_2} + S^1_{LM_1}C^2_{LM_2}\right]\begin{pmatrix} L & L & L' \\ M_1 & M_2 & -M_3 \end{pmatrix};$$

where $M_3 \geq 0$, $M_2 = M_3 - M_1$, and the superscripts indicate the grid on which the function is evaluated.

d. LINCOM. Calculates matrix of P' (i.e., over the second totally symmetric grid) over the crude symmetry vectors. Nonzero off-diagonal blocks identify sets of vectors belonging to the same symmetry species.

e. PHASER. Imposes a standard phase on all symmetry vectors. The standard phase for a given representation is taken as that evinced by the lowest L-value set of harmonics spanning that representation. If a parity distinction exists, the functions of opposite parity are given the same phase by means of an A_{1u} operator which couples them. Any arbitrary mixing of the final phased vectors is sorted out by subroutine CLEAN.

f. AUFUN. Multiplies an odd-parity spherical harmonic by the A_{1u} symmetry vector to change its parity.

5. Symmetry-adaptation of an Atomic Basis Set

a. BASIS. Constructs the necessary internal bookkeeping tables for the atomic basis set, which has been made available by:

b. INBAS. Generates an atomic basis set for the whole molecule, based either on input values or on table look-ups. Optimized minimal and double-zeta Slater-type basis functions are stored for H-Xe (valence shell only), and H-Kr, respectively. The value of LFLOW(4) controls the choice of stored basis set or the input of different values. (See Sec. IV.A below for description of input.)

c. ORBTLS. Supervises symmetry-adaptation of atomic basis set. Generates symmetrized functions for each distinct ℓ-value in each atom type, and sets up the necessary tables of information about the symmetry functions.

d. BASFUN. Evaluates a set of atomic basis functions at a particular grid point For each atom-centered function, the coordinates of the grid point are transformed to the atom-centered axis system, and a call is made to:

e. VALUE. Evaluates the $2\ell + 1$ atomic basis functions at a grid point. The angular parts of the functions are given in real Cartesian representation for speed of computation, and an overall radial dependence of r^{-1} or r^{-2} is affixed. Functions up through $\ell = 4$ are implemented.

The formulas used are:

$$S_1 = R^{-2},$$

$$P_1 = ZR^{-2},$$

$$P_2 = YR^{-2},$$

$$P_3 = XR^{-2},$$

$$D_1 = \frac{1}{2}(3Z^2 - R^2)\, R^{-4},$$

$$D_2 = \sqrt{3}YZR^{-4},$$

$$D_3 = \sqrt{3}XZR^{-4},$$

$$D_4 = \sqrt{3}XYR^{-4},$$

$$D_5 = \frac{\sqrt{3}}{2}(X^2 - Y^2)\, R^{-4},$$

$$F_1 = Z(5Z^2 - 3R^2)\, R^{-4},$$

$$F_2 = \frac{\sqrt{6}}{2}Y(5Z^2 - R^2)\, R^{-4},$$

$$F_3 = \frac{\sqrt{6}}{2}X(5Z^2 - R^2)\, R^{-4},$$

$$F_4 = \sqrt{60}XYZR^{-4},$$

$$F_5 = \sqrt{15}Z(X^2 - Y^2)\, R^{-4},$$

$$F_6 = \frac{\sqrt{10}}{2}Y[3(R^2 - Z^2) - 4Y^2]\, R^{-4},$$

$$F_7 = \frac{\sqrt{10}}{2}X[4X^2 - 3(R^2 - Z^2)]\, R^{-4},$$

$$G_1 = \left(\frac{35}{2}Z^4 - 15Z^2R^2 + \frac{3}{2}R^4\right) R^{-6},$$

$$G_2 = \frac{\sqrt{10}}{2} YZ(14Z^2 - 6R^2)\, R^{-6},$$

$$G_3 = \frac{\sqrt{10}}{2} XZ(14Z^2 - 6R^2)\, R^{-6},$$

$$G_4 = \sqrt{20}XY(7Z^2 - R^2)\, R^{-6},$$

$$G_5 = \frac{\sqrt{20}}{2}(X^2 - Y^2)(7Z^2 - R^2)\, R^{-6},$$

$$G_6 = \sqrt{70}YZ[3(R^2 - Z^2) - 4Y^2]\, R^{-6},$$

$$G_7 = \sqrt{70}XZ[4X^2 - 3(R^2 - Z^2)]\, R^{-6},$$

$$G_8 = 4\sqrt{35}XY(2X^2 - R^2 + Z^2)\, R^{-6},$$

and

$$G_9 = \sqrt{35}[4X^2(X^2 - R^2 + Z^2) + (R^2 - Z^2)^2]\, R^{-6}.$$

6. Finite Point-group Vector-coupling Coefficients

a. VECUPL. Calculates vector-coupling coefficients for the finite point group, and supervises related operations.

For the "symmetric" vector-coupling coefficients, the formula is

$$(\Gamma''\gamma''|\Gamma\gamma\Gamma'\gamma') = \eta \sum_m \sum_{m'} (-1)^{m+m'} \begin{pmatrix} \ell & \ell & \ell'' \\ m & m' & -m-m' \end{pmatrix}$$

$$\times\, [(AA' - BB')\, A'' + (BA' + AB')\, B''].$$

For the "antisymmetric" coefficients, the last factor is replaced by

$$[(BA' + AB')\, A'' - (AA' - BB')\, B''],$$

where $A' \equiv A_{\ell m'}^{\alpha'\Gamma'\gamma'}$, etc., and the A's and B's are defined in Eqs. 29.

b. ODDGA. Constructs set of transformation coefficients for odd-L-value prototype functions.

c. ORTHOG. Performs Schmidt orthogonalization of doubly-occurring sets of coupling coefficients for the special cases of point groups T, T_h, K, and K_h.

d. VPRINT. Prints out tables of selection rules and coupling coefficients.

D. Definition of Variables in Common Blocks

1. Glossary

The following definitions are used in the program:

Angular quantum number: A combined index denoting the irreducible representation and the subspecies. Assigned sequentially from $A_{1g} = 1$.

Basis index: An index denoting a particular set of n, ℓ, and ζ values in the atomic basis set. There are $2\ell + 1$ times the number of atoms of that element functions belonging to a given basis index.

K-value: An integer, $1 \leq K \leq 2\ell + 1$, indexing the real basis functions belonging to an ℓ-value; K = 2m indexes the function corresponding to sin $(m\varphi)$; K = 2m + 1 indexes the cos $(m\varphi)$ function.

Prototype function: A linear combination of real spherical harmonic functions which is symmetry-adapted and has a canonical phase.

Radial quantum number: An index denoting the particular occurrence of a symmetry species in the symmetry-adapted atomic basis set. That is, the third A_{1g} function would have a radial quantum number = 3, an angular quantum number = 1.

Transformation block: A set of symmetry transformation coefficients for atomic basis functions, characterized by a particular atom-type and ℓ-value.

Transformation index: An index given to the symmetry-adaptation of a particular set of atomic basis functions, i.e., a particular basis index on a particular atom-type. Since the transformation coefficients are determined by the ℓ-value of the functions and their atom-type, several transformation indices may refer to the same block of coefficients.

2. Blocks Containing Output from the Program

a. Block/OUT1/. Information about the nuclear geometry and chemical makeup of the molecule.

POS(I,IA): The Ith Cartesian coordinate for atom No. IA.

RADIUS(IA), THETA(IA), PHI(IA): The spherical polar coordinates for atom IA.

NATOMS: The number of atoms in the molecule.

NATYPE: The number of atom-types.

NUMT(IT): The number of atoms in atom-type IT.

NUCT(IT): The nuclear charge (or atomic number) of atoms in atom type IT.

ITSYMB(IT): The chemical symbol of atoms in type IT.

ITYPA(IN,IT): The loading index of the INth atom of atom-type IT.

ITYPT(IA): The atom-type to which atom IA belongs.

ITYPI(IA): The running index of atom IA within its atom-type.

INVAT(IA): The index of the atom inversion-symmetric with respect to atom IA (if a center of inversion is present).

b. Block/OUTB/. Information about the unsymmetrized atomic basis functions.

BCOEF(I): The self-consistent field coefficient of the Ith Slater-type orbital in the appropriate SCF atomic orbital basis vector.

ZETA(IB,IT): The orbital exponent of the function corresponding to basis index IB in atom-type IT.

NVALUE(IB,IT): The principal quantum number for basis index IB in atom-type IT.

LVALUE(IB,IT): The corresponding value of ℓ.

NBAST(IT): The number of basis indices (sets of r, ℓ, ζ values) on atoms of type IT.

ICRK(IT): The number of core electrons on atoms of type IT.

NASB(IT): The number of atomic symmetry blocks (distinct ℓ-values) for atom-type IT.

IBLKA(L+1,IT): The index of the block of transformation coefficients for the particular L-value in atom-type IT.

NBLK: The number of transformation blocks making up the array ASCOEF.

NBVB(IBLK): The number of basis vectors from functions in IBLK.

NBIB(IBLK): The number of basis indices represented in IBLK.

ITB(IBLK): The atom-type for the transformation block IBLK.

LVB(IBLK): The L-value characterizing the block IBLK.

IBIT(IB,IBLK): The basis index corresponding to running index IB within the block IBLK.

IBCFST(IBLK): The index in BCOEF of the first coefficient belonging to functions in transformation block IBLK.

NOR: The total number of orbitals in the basis set.

IAT(IAO): The index of the atom on which orbital IAO is centered.

IBT(IAO): The basis index of orbital IAO.

IKT(IAO): The k-value ($k = 1, 2, \ldots, 2\ell + 1$: a sequential index numbering the real atomic orbitals of given $\underline{\ell}$) for orbital IAO.

IAOFST(IB,IA): The running index of the first member of basis index IB on atom IA.

c. Block/OUT2/. Information about the symmetry transformation of the atomic orbitals and about the new, symmetry-adapted functions.

ASCOEF(IARRAY): The symmetry-transformation coefficient indexed by IARRAY.

ITRANT(IB,IT): The transformation index corresponding to basis index IB of atom-type IT.

IAOT(IARUN,ITRANS): The index of the IARUNth orbital in the transformation block indexed by ITRANS [= ITRANT(IB,IT)].

ITRAN(ISO): The transformation index corresponding to symmetry-adapted basis function ISO.

ISRUNT(ISO): The running index of symmetry orbital ISO within its transformation index.

IANGT(ISO): The "angular quantum number" of symmetry orbital ISO.

IRADT(ISO): The corresponding "radial quantum number."

IFIRST(ITRANS): The index of the first coefficient in ASCOEF indexed by ITRANS.

ISOT(ISRUN,ITRANS): The index of the ISRUNth symmetry orbital in the transformation block ITRANS.

IFB(IBLK): The index in ASCOEF of the first coefficient in the transformation block IBLK.

ISOQ(IANG,IRAD): The symmetry orbital whose angular and radial quantum numbers are IANG and IRAD, respectively.

NRADA(IREP): The number of times the representation IREP occurs in the atomic basis set.

d. Block/OUT3/. Information about the point group to which the molecule belongs.

NREP: The number of irreducible representations in the point group.

NDEGA(IREP): The degeneracy of the irreducible representation IREP.

IANGA(I,IREP): The angular quantum number assigned to the Ith row of representation IREP.

IREPA(IANG): The representation to which angular quantum number IANG belongs.

IORDER: The order of the group.

LVAL1: The ℓ-value of the prototype functions that span all the even representations of the group.

LVAL2: The ℓ-value spanning all the odd representations. (Set = 0 if no center of inversion.)

IRTYP(IREP): The Mulliken symbol (approximately) for the representation IREP.

IRPAR(IREP): The parity symbol (g or u) for the representation IREP.

IODD: Equals 1 if a center of inversion is present, 0 otherwise.

IRLOWL(IREP): Low ℓ-value spanning the representation IREP. Set = -1 if no low ℓ-value spans the representation.

e. Block/OUT4/. Information about the finite point-group vector-coupling coefficients.

VCCOEF(IVSUB): The normalized, unsymmetrized vector-coupling coefficient indexed by IVSUB.

IVCCB(IDPR,IR3): The index of the coefficient block referring to IDPR = IR2 + IR1 · (IR1 - 1)/2 and IR3, where IR2 ≤ IR1. If IVCCB(IDPR,IR3) = 0, IR3 is not contained in the direct product of IR1 x IR2.

IFVCC(IVCB): The address of the first coefficient in VCCOEF of the coefficient in VCCOEF of the coefficient block IVCB.

3. Blocks Containing Information Internal to the Program

a. Block/CONTRL/. Control parameters and thresholds.

CRTZRO: The zero threshold. A quantity A is set to zero if $|A| \leq$ CRTZRO. Default is 10^{-8}.

CRTDEG: The degeneracy criterion. A and B are considered degenerate if CRTDEG $< A/B <$ $(\text{CRTDEG})^{-1}$. Default is 1.00000001.

ISTOP: Abnormal-termination flag. If program detects an error and retains control, ISTOP is set to 1.

INTER: Intermediate-printout flag. If INTER = 1, many intermediate results are printed out for diagnostic purposes.

LFLOW(I): If LFLOW(I) = 1, the Ith major program segment is executed. See input description (Sec IV.A below) for details.

b. Block/PROTCO/. Information about the prototype functions spanning the irreducible representations of the group.

COPRO(ISUB): The transformation coefficient indexed by ISUB; coefficients are packed columnwise into COPRO.

EVP(IEV): The eigenvalue associated with the IEVth transformation vector; used only in output to show members of a degenerate set.

IREPL(I): The irreducible representation index assigned to the Ith eigenvector (or set of degenerate eigenvectors).

NEVALP(IPAR): The number of distinct eigenvalues in the set of prototype functions of parity IPAR (= 1 for g, 2 for u).

c. Block/GIRD/. Information about the points making up the domain of the totally symmetric operator P.

RGRID(IA,IGRID): The radial distance of points about atom IA and belonging to grid IGRID, from the origin.

CTHGR(IND): The cosine of the θ-coordinate of grid point IND, as measured from the master axis system.

PHIGR(IND): The φ-coordinate of grid point IND.

NGRID: The total number of points in each grid.

NDIV: The number of equally spaced points on each circle comprising the grid.

d. Block/XYZ/. Further information about the grid points.

XG(IND), YG(IND), ZG(IND): The Cartesian coordinates of grid point IND in the master coordinate system.

X(IA), Y(IA), Z(IA): The Cartesian coordinates of atoms IA in the master coordinate system. (These are identical with POS(I,IA), I = 1, 2, 3.)

e. Block/WORK1/. A work area for use within major segments of the program. Content varies, but variables most frequently occurring are:

RES(I,J): A matrix containing the matrix elements of P over one or two sets of functions.

EV(I) or VALU(I): The Ith eigenvalue resulting from the diagonalization of RES.

C(I,J): The eigenvector coefficients that diagonalize RES.

B(I,J): A scratch array, same size as RES or C.

NDEG(K): The degeneracy of the Kth distinct eigenvalue of RES.

NEVAL: The number of distinct eigenvalues of RES.

IREPLB(K): The irreducible representation index assigned to the Kth set of eigenvectors.

LABEL, INDEX: Hollerith and integer arrays used to label rows of matrices printed out.

f. Block/PHASE/. Control variables for phasing of prototype functions.

IFLGG(I) (or IFLAG(I)): Flag set to 1 if representation I has been phased properly.

IPHASE: Control flag determining which functions are phased against which in subroutine LINEUP.

g. Block /ALP/. Parameter communication among routines PSYM, THLMK, AND THETAX.

RCR: Reciprocal of distance of current grid point from origin. SQ2OR: $RCR/\sqrt{2}$.

W (or X): (Cos θ) of current grid point.

Y: (Sin θ) of current grid point.

LC(or L): ℓ-value of current set of spherical harmonics.

h. Block/UNGRAD/. Information about A_{1u} transformation vector (if any).

CAU(I): Ith coefficient in A_{1u} transformation vector.

LODD: ℓ-value spanning the A_{1u} representation.

LODEG: 2 · LODD + 1.

i. Block/WIGNER/. 3-j symbols for set of ℓ-values.

S3J(MSUB1, M3P1): 3-j symbol $\begin{pmatrix} \ell & \ell & \ell' \\ m_1 & m_2 & -m_3 \end{pmatrix}$, where ℓ' is even, MSUB1 = $|m_2 - m_1|/2 + 1$, M3P1 = $m_3 + 1$, and $m_3 \geq 0$, $m_1 \geq \frac{1}{2}$

Z3J(MSUB1, M3P1): 3-j symbol as above, except ℓ' is odd.

j. Block/WORK2/. Communication among routines ORBTLS, BASFUN, and VALUE.

XP or X, YP or Y, ZP or Z } Cartesian coordinates of current grid point in an atom-centered system.

RP or R: Distance of current grid point from current atom.

JSUB or ISUB: Index of last function before set on current atom.

LDEG: $2 \cdot L + 1$, where L is the ℓ-value of the current set of basis functions.

NUMB: The number of atoms in the current atom-type.

IT: The index of the current atom-type.

k. Block/ALP1/. Information that is invariant from molecule to molecule.

THETAC(ISUB): The coefficient $C_{\ell mr}$ for which ISUB equals $8m + 4\ell(\ell + 1) + r + 1$.

FACT(N): The value of $(N - 1)!$.

l. Block/FILES/. Data-set reference numbers.

INPUT: The logical unit number for input; usually = 5.

IOUT: The logical unit number for output; usually = 6.

m. Block/ASYMI/. Parameter communication between subroutines PROSYM and ANTSYM.

LA or L: The even ℓ-value used to form the odd-ℓ set of spherical harmonics. $LA = \max(2, \frac{1}{2}\ell_{even} + 1)$.

MGRID or NGRID: The total number of points in a grid.

L3P1: $L_{odd} + 1$.

LAP1 or LP1: $LA + 1$.

LADEG or LL: $2 \cdot LA + 1$.

n. Block/SCAN1/. Communication between subroutines CLEAN and SCAN.

IFLAG(K): A flag indicating that the Kth component of the column vector has already been examined IFLAG(K) = 1.

NSC(IC): The number of coefficients in the vector having the ICth distinct numerical absolute value.

SAMEC(IC): The value of the ICth distinct numerical magnitude in the vector.

JM: The index of the particular eigenvector being examined.

IC: The number of distinct numerical values counted in the vector.

IV. INSTRUCTIONS FOR USE

A. Description of Input

The following discussion applies to the IBM 360/75 version of the program, using the NAMELIST capability of IBM FORTRAN IV. Since the input is so brief and is handled entirely in subroutines MOLCUL and INBAS, the user who does not have the NAMELIST capability available will find it simple to write his own input format. The input is in three logical components:

Component A1. Problem identification and control parameters. The parameter card (or cards) is in NAMELIST format (see IBM OS/360 Manual C28-6515, "FORTRAN IV Language," for details), and the name of the list is PARAMS. Only those parameters for which the default option is not used need be specified.

Component A2. Cartesian coordinates of atoms and their chemical symbols. In NAMELIST format, with the list ATOM. Default options are provided. Each atom requires at least one card.

Component A3. Atomic basis function parameters. (Optional; used only if LFLOW(4) = 9.) In standard A, I, or F formats.

Table III gives a detailed description of the variable lists and their formats; Figs. 13 and 14 show typical input decks with and without user-supplied basis sets, respectively.

B. Built-in Basis Sets

A feature that was felt to be particularly useful in facilitating the use of the program is the prestoring of several standard, optimized atomic basis sets for each of a wide range of elements. This was done because in a large, multicenter calculation, complete optimization of the basis set will probably not be feasible and the "best atom" sets from the literature are likely to be taken as the starting point. Two sets have been implemented: a minimal, valence-shell basis set for elements hydrogen through xenon,

and a double-zeta set for elements hydrogen through krypton. Tables IV and V show the two types of basis sets we have stored, and give literature references.

Since the user may have his own input scheme for basis function parameters, the input or assignment routine is made completely separate from the generation of cross-reference tables needed for the symmetry-adaptation bookkeeping. Thus the input routine, INBAS, may be replaced by the user's own module, as long as the arrays (NBAST, CRCHRG, NVALUE, LVALUE, ZETA, and BCOEF) are properly filled and transmitted through COMMON/OUTB/.

TABLE III. Description of Input Deck

Card No.	Variable	Description and Format	Default Option
		Component A1. Job I.D. and Control	
1.1	IDENT(20)	Problem identification (19A4,A3).	None
	JOB	New-case flag: An asterisk (*) in column 80. Job terminates if a slash (/) is encountered in column 80.	None
1.2		Parameters in NAMELIST format (namelist name = PARAMS).	
	NATOMS	Number of atoms in the molecule.	None
	CRTZRO	Zero threshold.	1.0×10^{-8}
	CRTDEG	Degeneracy criterion.	1.00000001
	CRTTYP	Atom-type criterion: Two atoms of the same element are put into the same atom-type when $\lvert R_i - R_j \rvert \leq$ CRTTYP.	1.0×10^{-5}
	NTCHEK	The anticipated number of atom-types; used as a check on the input.	1
	INTER	Intermediate printout flag: = 1 if desired, 0 otherwise.	0
	SCALE	Factor by which all Cartesian coordinates are multiplied.	1.0
	NDIV	Number of equally spaced points on each grid circle.	24
	LFLOW(10)	Logical flow control array. A "1" in the appropriate element means that branch is taken.	
		LFLOW(1): Symmetry-adapted atomic basis set.	1
		LFLOW(2): Vibration-rotation symmetry analysis (not yet implemented).	1
		LFLOW(3): Finite point-group vector-coupling coefficients.	1
		LFLOW(4): Choice of atomic basis set.	1
		= 1: Stored, minimal valence-shell Slater basis.	
		= 2: Stored double-zeta Slater basis.	
		= 9: Basis set read in by user.	
		LFLOW(5)-LFLOW(10): Available for further logic control	0
		Component A2. Atomic Coordinates	
2.1 to 2.NATOMS		Each coordinate is expressed internally in the form	
		X = XOP(FX).	
		Parameters in NAMELIST format (namelist name = ATOM).	
	ELMNT	The chemical symbol for the element (e.g., 'FE').	None
	X, Y, Z	The Cartesian coordinates, which may or may not be modified by operators.	0.0
	XOP, YOP, ZOP	Operations, including 'SIN', 'COS', '1/', 'SQRT', ' ' (no operation). If 'SIN' or 'COS' is used, the operand must be in degrees.	' '
	FX, FY, FZ	The operands of XOP, YOP, and ZOP.	1.0
		Component A3. Basis-set Input	None
3.1	NASB(IT)	The number of atomic symmetry blocks for atom-type IT.	
	ICRK(IT)	The number of "core" electrons (2I5).	
3.2	LVB	The ℓ-value of functions in the symmetry block.	
	NBIB	The number of sets of (n,ζ) values in the block.	
	NBVB	The number of basis vectors formed from the primitive functions. ($\leq$ NBIB) (3I5).	
3.3 to 3.NBIB+2	NVALUE	The principal quantum number of the functions.	
	ZETA	The orbital exponent. (One set to a card, (I5, F10.5)).	
3.NBIB+3 to 3.NBIB+2+NBVB	(BCT(I), I=1, NBIB)	The basis-vector coefficients.	
		(Each vector begins a new card, (8F10.8)).	
		Repeat from card 3.2 for each atomic symmetry block in atom-type IT.	
		Repeat from card 3.1 for each atom-type IT.	

```
STAGGERED ETHANE - EXTENDED BASIS SET                                                    *
 &PARAMS NATOMS=8,NTCHEK=2,LFLOW(3)=0,LFLOW(4)=9 &END
 &ATOM ELMNT='C',Z=1. &END
 &ATOM ELMNT='H',Y=2.,Z=2. &END
 &ATOM ELMNT='H',X=1.,XOP='SQRT',FX=3.,Y=-1.,Z=2. &END
 &ATOM ELMNT='H',X=-1.,XOP='SQRT',FX=3.,Y=-1.,Z=2. &END
 &ATOM ELMNT='C',Z=-1. &END
 &ATOM ELMNT='H',Y=-2.,Z=-2. &END
 &ATOM ELMNT='H',X= 1.,XOP='SQRT',FX=3.,Y=1.,Z=-2. &END
 &ATOM ELMNT='H',X=-1.,XOP='SQRT',FX=3.,Y=1.,Z=-2. &END
    2    0
    0    4    2
    1     9.153
    1     5.382
    2     1.428
    3     3.076
   0.08367   0.92300  -0.00049   0.00373
  -0.00604  -0.22226   0.77062   0.28311
    1    3    1
    2    5.152
    2    2.177
    2    1.150
0.02472   0.39516   0.64975
    1    0
    0    1    1
    1    1.000
  1.000
```

Fig. 13. Example of Input Deck Including a Basis Set Supplied by the User

```
                          ICOSHEDRAL B(12) SKELETON                                      *
 &PARAMS NATOMS=12, LFLOW(4)=2, INTER=1  &END
 &ATOM ELMNT='B', Z=1.0, ZOP='SQRT', FZ=1.25       &END
 &ATOM ELMNT='B',     Y=1.0, Z=0.5            &END
 &ATOM ELMNT='B',X= 1.0,XOP='SIN',FX= 72.0, Y=1.0,YOP='COS',FY= 72.0,Z=0.5  &END
 &ATOM ELMNT='B',X= 1.0,XOP='SIN',FX=144.0, Y=1.0,YOP='COS',FY=144.0,Z=0.5  &END
 &ATOM ELMNT='B',X=-1.0,XOP='SIN',FX=144.0, Y=1.0,YOP='COS',FY=144.0,Z=0.5  &END
 &ATOM ELMNT='B',X=-1.0,XOP='SIN',FX= 72.0, Y=1.0,YOP='COS',FY= 72.0,Z=0.5  &END
 &ATOM ELMNT='B', Y=-1.0, Z=-0.5       &END
 &ATOM ELMNT='B',X=-1.0,XOP='SIN',FX=108.0,Y=1.0,YOP='COS',FY=108.0,Z=-.5   &END
 &ATOM ELMNT='B',X=-1.0,XOP='SIN',FX= 36.0,Y=1.0,YOP='COS',FY= 36.0,Z=-.5   &END
 &ATOM ELMNT='B',X= 1.0,XOP='SIN',FX= 36.0,Y=1.0,YOP='COS',FY= 36.0,Z=-.5   &END
 &ATOM ELMNT='B',X= 1.0,XOP='SIN',FX=108.0,Y=1.0,YOP='COS',FY=108.0,Z=-.5   &END
 &ATOM ELMNT='B', Z=-1.0,ZOP='SQRT',FZ=1.25       &END
```

Fig. 14. Example of Input Deck with Basis Set Selected by the Program

TABLE IV. Prestored Minimal Valence-shell Atomic Basis Set

Element	n	ℓ	ζ	Element	n	ℓ	ζ
H	1	S	1.0000	Ge	4	S	2.0109
He	1	S	1.6875		3	D	5.4171
Li	2	S	0.640		4	P	1.6951
Be	2	S	0.956	As	4	S	2.2360
B	2	S	1.288		3	D	5.7928
	2	P	1.211		4	P	1.8623
C	2	S	1.623	Se	4	S	2.4394
	2	P	1.487		3	D	6.1590
N	2	S	1.924		4	P	2.0718
	2	P	1.917	Br	4	S	2.6382
O	2	S	2.246		3	D	6.5197
	2	P	2.227		4	P	2.2570
F	2	S	2.564	Kr	4	S	2.8289
	2	P	2.550		3	D	6.8753
Ne	2	S	2.879		4	P	2.4423
	2	P	2.879	Rb	5	S	0.9969
Na	3	S	0.8358	Sr	5	S	1.2141
Mg	3	S	1.1025	Y	5	S	1.2512
Al	3	S	1.3724		4	D	3.9896
	3	P	1.3552	Zr	5	S	1.2891
Si	3	S	1.6344		4	D	3.2679
	3	P	1.4284	Nb	5	S	1.3392
P	3	S	1.8806		4	D	3.0796
	3	P	1.6288	Mo	5	S	1.3952
S	3	S	2.1223		4	D	3.1110
	3	P	1.8273	Tc	5	S	1.4453
Cl	3	S	2.3561		4	D	3.2205
	3	P	2.0387	Ru	5	S	1.4905
Ar	3	S	2.5856		4	D	3.3470
	3	P	2.2547	Rh	5	S	1.5286
K	4	S	0.8738		4	D	3.4937
Ca	4	S	1.0995	Pd	5	S	1.5675
Sc	4	S	1.1581		4	D	3.6476
	3	D	2.3733	Ag	5	S	1.6057
Ti	4	S	1.2042		4	D	3.8064
	3	D	2.7138	Cd	5	S	1.6384
V	4	S	1.2453		4	D	3.9692
	3	D	2.9943	In	5	S	1.9023
Cr	4	S	1.2833		4	D	4.2354
	3	D	3.2522		5	P	1.6940
Mn	4	S	1.3208	Sn	5	S	2.1257
	3	D	3.5094		4	D	4.4925
Fe	4	S	1.3585		5	P	1.8204
	3	D	3.7266	Sb	5	S	2.3222
Co	4	S	1.3941		4	D	4.7436
	3	D	3.9518		5	P	1.9989
Ni	4	S	1.4277	Te	5	S	2.5076
	3	D	4.1765		4	D	4.9900
Cu	4	S	1.4606		5	P	2.1617
	3	D	4.4002	I	5	S	2.6807
Zn	4	S	1.4913		4	D	5.2335
	3	D	4.6261		5	P	2.3223
Ga	4	S	1.7667	Xe	5	S	2.8436
	3	D	5.0311		4	D	5.4733
	4	P	1.5554		5	P	2.4849

References:

H-Ne: P. Bagus and T. L. Gilbert, unpublished work.

Na-Xe: E. Clementi et al., J. Chem. Phys. 38, 2686 (1963); J. Chem. Phys. 47, 1300 (1967).

TABLE V. Examples of Double-zeta Slater Basis Sets

n	ℓ	ζ	SCF Coefficients C_1	C_2	C_3	C_4
Kr						
1	0	28.9410	0.30191	-0.47092	0.18287	-0.06059
1	0	37.8590	0.71223	0.02751	-0.00684	0.00387
2	0	15.5500	0.02495	1.38570	-0.96946	0.31961
2	0	18.8680	-0.03566	-0.27267	0.36941	-0.12187
3	0	6.5065	0.00393	-0.03026	0.73304	-0.37484
3	0	8.3700	-0.00544	0.06030	0.41654	-0.10581
4	0	2.4358	0.00025	-0.00133	-0.00322	0.70702
4	0	4.0461	-0.00075	0.00422	0.02389	0.45131
2	1	15.4600	0.93533	-0.42145	0.11673	
2	1	29.2240	0.05118	-0.01096	0.00227	
3	1	8.1315	0.07231	0.44406	-0.14806	
3	1	6.0579	-0.03686	0.64246	-0.18834	
4	1	1.8095	-0.00221	-0.00943	0.57945	
4	1	3.2184	0.00572	0.00309	0.54963	
3	2	5.1816	0.70696			
3	2	10.3260	0.37915			
F						
1	0	8.3660	1.06848	-0.24322		
2	0	10.8850	-0.07301	-0.02281		
2	0	2.6270	0.00379	1.03628		
2	1	4.1800	0.35507			
2	1	1.8480	0.72760			

References:

He, Na-Ar: Clementi et al., J. Chem. Phys. 40, 1944 (1964).
Li-Ne: Bagus and Gilbert, unpublished work.
K-Kr: Clementi et al., J. Chem. Phys. 47, 1865 (1967).

C. Sample Input and Program Output

Table VI summarizes the input and output for the program. Figures 13-15 show input decks and sample output.

TABLE VI. Summary of Input and Output for the Computer Program as Currently Implemented

Input
- Chemical symbols and Cartesian coordinates of atoms
- Parameters describing basis set (either read in or selected from the tables stored in program)
- Control parameters (or use of default options)

Output
- Characterization of all symmetry species in point group
- Identification of point group
- Set of symmetry-adapted prototype functions
- Transformation coefficients for converting atomic basis set to symmetry-adapted set
- Labels and degeneracies for symmetry species spanned by basis set
- Internal bookkeeping information
- Selection rules for direct products of representations
- Finite point-group vector-coupling coefficients

D. Table Limits and Error Messages

The following are the limits of various variables according to the current implementation of the program:

No. of atoms ≤ 30

Total No. of basis functions ≤ 120

No. of atom-types ≤ 10

No. of transformation blocks ≤ 50

Maximum matrix size $\leq 50 \times 50$

No. of irreducible representations ≤ 14

No. of ℓ-values per atom-type ≤ 5

No. of basis vector coefficients ≤ 500

No. of symmetry transformation coefficients ≤ 3000

ℓ-value of basis function ≤ 4

No. of points per grid ≤ 720

ℓ-value of prototype functions ≤ 16

Due principally to the large matrices provided for, the program requires roughly 388K bytes of core storage without overlays. If space is a consideration, of course, the matrix sizes can be reduced.

Several error messages, whose content should be self-explanatory, are included in the program for debugging purposes and should not arise under normal operation, unless a mistake has been made in the input. See the program listings for the specific messages.

```
**** GPTHEORY - AUTOMATED MOLECULAR POINT GROUP THEORY ****
THOMAS D. BOUMAN AND GORDON L. GOODMAN
CHEMISTRY DIVISION, ARGONNE NATIONAL LABORATORY
ARGONNE, ILLINOIS 60439

PROBLEM TITLE --          KRYPTON HEXAFLUORIDE, DOUBLE-ZETA BASIS SET                    9/14/71

INPUT DESCRIPTION OF MOLECULE
-----------------------------
    NUMBER OF ATOMS =  7
    ASSUMED NO. OF EQUIVALENT SETS (ATOM TYPES) =  2

    CARTESIAN COORDINATES (IN FORM  X = A*FUNC(B)  )

     ATOM   CHEMICAL          X                        Y                        Z
    NUMBER   SYMBOL      A     FUNC    B          A     FUNC    B          A     FUNC    B

       1       KR      0.0           1.000      0.0           1.000      0.0           1.000
       2       F       1.0000        1.000      0.0           1.000      0.0           1.000
       3       F       0.0           1.000      1.0000        1.000      0.0           1.000
       4       F       0.0           1.000      0.0           1.000      1.0000        1.000
       5       F      -1.0000        1.000      0.0           1.000      0.0           1.000
       6       F       0.0           1.000     -1.0000        1.000      0.0           1.000
       7       F       0.0           1.000      0.0           1.000     -1.0000        1.000

THRESHOLDS AND CONTROL PARAMETERS
---------------------------------
    ZERO THRESHOLD = 1.00D-08
    DEGENERACY TEST (RATIO OF EIGENVALUES) =  1.00000001
    ATOM EQUIVALENCE THRESHOLD = 1.00D-05
    NO. OF GRID POINTS PER ATOM =  24
    SCALE FACTOR MULTIPLYING COORDINATES =   10.00000
    PROGRAM OPTIONS SELECTED -
         POINT GROUP PROPERTIES
         SYMMETRY-ADAPTED ATOMIC BASIS SET
           DOUBLE-ZETA SLATER BASIS
         VIBRATION-ROTATION SYMMETRY
         VECTOR-COUPLING COEFFICIENTS
```

Fig. 15. Sample Output from the Program for an Octahedral KrF_6 Molecule, quantized about the Fourfold Axis

LECULAR GEOMETRY ANALYSIS

ATOM NUMBER	CHEMICAL SYMBOL	ATOMIC NUMBER (Z)	CARTESIAN COORDINATES IN CENTER OF CHARGE SYSTEM X	Y	Z
1	KR	36	0.0	0.0	0.0
2	F	9	1.00000000	0.0	0.0
3	F	9	0.0	1.00000000	0.0
4	F	9	0.0	0.0	1.00000000
5	F	9	-1.00000000	0.0	0.0
6	F	9	0.0	-1.00000000	0.0
7	F	9	0.0	0.0	-1.00000000

ATOM NUMBER	CHEMICAL SYMBOL	ATOMIC NUMBER (Z)	SPHERICAL COORDINATES IN CENTER OF CHARGE SYSTEM R	THETA	PHI
1	KR	36	0.0	0.0	0.0
2	F	9	1.00000000	1.57079633	0.0
3	F	9	1.00000000	1.57079633	1.57079633
4	F	9	1.00000000	0.0	0.0
5	F	9	1.00000000	1.57079633	3.14159265
6	F	9	1.00000000	1.57079633	4.71238898
7	F	9	1.00000000	3.14159265	0.0

THE PROGRAM DISTINGUISHES 2 ATOM TYPES, AS FOLLOWS...

ATOM-TYPE INDEX	CHEMICAL SYMBOL	NO. OF ATOMS	NUCLEAR CHARGE	UPPER : LOADING INDICES OF THE ATOMS IN THIS ATOM-TYPE LOWER : RUNNING INDICES WITHIN THE ATOM-TYPE
1	KR	1	36	1 1
2	F	6	9	2 3 4 5 6 7 1 2 3 4 5 6

CENTER OF INVERSION - YES

Fig. 15 (Contd.)

SUMMARY OF POINT GROUP PROPERTIES

NO. OF IRREDUCIBLE REPRESENTATIONS = 10
COMPLEX-CONJUGATE PAIRS OF I.R.S. - NO
ORDER OF THE GROUP = 48
THE POINT GROUP FOR THIS NUCLEAR CONFIGURATION IS O(H)
L VALUES OF REAL SPHERICAL HARMONICS USED TO SPAN ALL THE I.R.S IN THIS GROUP - 6 9

REPRESENTATION INDEX	DEGENERACY	SPECIES LABEL	PHASE STANDARD	ANGULAR QUANTUM NUMBERS		
1	1	A,B(G)	L = 0	1		
2	1	A,B(G)	(L = 3)	2		
3	2	E(G)	L = 2	3	4	
4	3	T(G)	L = 2	5	6	7
5	3	T(G)	(L = 1)	8	9	10
6	1	A,B(U)	(L = 0)	11		
7	1	A,B(U)	L = 3	12		
8	2	E(U)	(L = 2)	13	14	
9	3	T(U)	(L = 2)	15	16	17
10	3	T(U)	L = 1	18	19	20

ATOMIC BASIS SET AND SYMMETRY ADAPTATION

TRANSFORMATION BLOCK 1

ATOM-TYPE 1, S FUNCTIONS FOR ELEMENT KR

TOTAL OF 1 BASIS FUNCTIONS FOR EACH BASIS INDEX

BLOCK OF SYMMETRY TRANSFORMATION COEFFICIENTS STARTS AT LOCATION 1 OF ASCOEF

BASIS INDEX	TRANSF. INDEX	QUANTUM NUMBERS N	ZETA	BASIS VECTORS C(1)	C(2)	C(3)	C(4)
1	1	1	28.9410	0.30191	-0.47092	0.18287	-0.06059
2	2	1	37.8590	0.71223	0.02751	-0.00684	0.00387
3	3	2	15.5500	0.02495	1.38570	-0.96946	0.31961
4	4	2	18.8680	-0.03566	-0.27267	0.36941	-0.12187
5	5	3	6.5065	0.00393	-0.03026	0.73304	-0.37484
6	6	3	8.3700	-0.00544	0.06030	0.41654	-0.10581
7	7	4	2.4358	0.00025	-0.00133	-0.00322	0.70702
8	8	4	4.0461	-0.00075	0.00422	0.02389	0.45131

Fig. 15 (Contd.)

TRANSFORMATION BLOCK 2

ATOM-TYPE 1, P FUNCTIONS FOR ELEMENT KR

TOTAL OF 3 BASIS FUNCTIONS FOR EACH BASIS INDEX

BLOCK OF SYMMETRY TRANSFORMATION COEFFICIENTS STARTS AT LOCATION 2 OF ASCOEF

BASIS INDEX	TRANSF. INDEX	QUANTUM NUMBERS N	ZETA	BASIS VECTORS C(1)	C(2)	C(3)
9	9	2	15.4600	0.93533	-0.42145	0.11673
10	10	2	29.2240	0.05118	-0.01096	0.00227
11	11	3	8.1315	0.07231	0.44406	-0.14806
12	12	3	6.0579	-0.03686	0.64246	-0.18834
13	13	4	1.8095	-0.00221	-0.00943	0.57945
14	14	4	3.2184	0.00572	0.00309	0.54963

TRANSFORMATION BLOCK 3

ATOM-TYPE 1, D FUNCTIONS FOR ELEMENT KR

TOTAL OF 5 BASIS FUNCTIONS FOR EACH BASIS INDEX

BLOCK OF SYMMETRY TRANSFORMATION COEFFICIENTS STARTS AT LOCATION 11 OF ASCOEF

BASIS INDEX	TRANSF. INDEX	QUANTUM NUMBERS N	ZETA	BASIS VECTORS C(1)
15	15	3	5.1816	0.70696
16	16	3	10.3260	0.37915

TRANSFORMATION BLOCK 4

ATOM-TYPE 2, S FUNCTIONS FOR ELEMENT F

TOTAL OF 6 BASIS FUNCTIONS FOR EACH BASIS INDEX

BLOCK OF SYMMETRY TRANSFORMATION COEFFICIENTS STARTS AT LOCATION 36 OF ASCOEF

BASIS INDEX	TRANSF. INDEX	QUANTUM NUMBERS N	ZETA	BASIS VECTORS C(1)	C(2)
1	17	1	8.3660	1.06848	-0.24322
2	18	2	10.8850	-0.07301	-0.02281
3	19	2	2.6270	0.00379	1.03628

Fig. 15 (Contd.)

```
TRANSFORMATION BLOCK  5

ATCM-TYPE  2,  P FUNCTIONS FOR ELEMENT F

TOTAL OF  18 BASIS FUNCTIONS FOR EACH BASIS INDEX

BLOCK OF SYMMETRY TRANSFCRMATICN COEFFICIENTS STARTS AT LOCATION   72 OF ASCOEF

BASIS  TRANSF.  QUANTUM NUMBERS   BASIS VECTORS
INDEX   INDEX      N    ZETA         C( 1)

  4      20        2   4.18C0        0.35507
  5      21        2   1.8480        0.72760


 FINAL SYMMETRY VECTORS FOR ATOM-TYPE  1, L = 0

COLUMNS  1 TO  1

EIGENVALUES

              2.1759440318D 02

EIGENVECTORS

     S ( 1)    1.00CC000000D 00


ANGULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTOR

   1


 FINAL SYMMETRY VECTORS FOR ATOM-TYPE  1, L = 1

  LUMNS  1 TO  3

  GENVALUES

              5.9004325696D 01    5.9004325696D 01    5.9004325696D 01

  GENVECTORS

     Z ( 1)    1.00CC000000D 00    0.0                 0.0

     Y ( 1)    0.0                 1.0000000000D 00    0.0

     X ( 1)    0.0                 0.0                 1.0000000000D 00


  GULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTOR

  18   19   20
```

Fig. 15 (Contd.)

```
 FINAL SYMMETRY VECTORS FOR ATOM-TYPE  1, L = 2
COLUMNS  1 TO  5
EIGENVALUES
                1.0739834917D 02    1.0739834917D 02    9.3256827896D-01    9.3256827896D-01    9.3256827896D-01
EIGENVECTORS
 3Z2-1 ( 1)     1.0000000000D 00    0.0                 0.0                 0.0                 0.0
    YZ ( 1)     0.0                 0.0                 1.0000000000D 00    0.0                 0.0
    XZ ( 1)     0.0                 0.0                 0.0                 1.0000000000D 00    0.0
    XY ( 1)     0.0                 0.0                 0.0                 0.0                 1.0000000000D 00
 X2-Y2 ( 1)     0.0                 1.0000000000D 00    0.0                 0.0                 0.0

ANGULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTOR
   3    4    5    6    7

 FINAL SYMMETRY VECTORS FOR ATOM-TYPE  2, L = 0
COLUMNS  1 TO  6
EIGENVALUES
                1.4068530633D 05    1.2876429568D 05    1.2876429568D 05    1.3071148504D 05    1.3071148504D 05    1.3071148504D 05
EIGENVECTORS
     S ( 1)     4.0824829046D-01   -2.8867513459D-01    5.0000000000D-01    0.0                 0.0                 7.0710678119D-01
     S ( 2)     4.0824829046D-01   -2.8867513459D-01   -5.0000000000D-01    0.0                 7.0710678119D-01    0.0
     S ( 3)     4.0824829046D-01    5.7735026919D-01    0.0                 7.0710678119D-01    0.0                 0.0
     S ( 4)     4.0824829046D-01   -2.8867513459D-01    5.0000000000D-01    0.0                 0.0                -7.0710678119D-01
     S ( 5)     4.0824829046D-01   -2.8867513459D-01   -5.0000000000D-01    0.0                -7.0710678119D-01    0.0
     S ( 6)     4.0824829046D-01    5.7735026919D-01    0.0                -7.0710678119D-01    0.0                 0.0

ANGULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTOR
   1    3    4   18   19   20
```

Fig. 15 (Contd.)

FINAL SYMMETRY VECTORS FOR ATOM-TYPE 2, L = 1

COLUMNS 1 TO 6

EIGENVALUES

	2.4674313665D 03	1.1196818379D 03	1.1196818379D 03	2.3558150381D 02	2.3558150381D 02	2.3558150381D 02

EIGENVECTORS

Z (1)	0.0	0.0	0.0	0.0	5.0000000000D-01	0.0
Y (1)	0.0	0.0	0.0	0.0	0.0	5.0000000000D-01
X (1)	-4.0824829046D-01	2.8867513459D-01	-5.0000000000D-01	0.0	0.0	0.0
Z (2)	0.0	0.0	0.0	5.0000000000D-01	0.0	0.0
Y (2)	-4.0824829046D-01	2.8867513459D-01	5.0000000000D-01	0.0	0.0	0.0
X (2)	0.0	0.0	0.0	0.0	0.0	5.0000000000D-01
Z (3)	-4.0824829046D-01	-5.7735026919D-01	0.0	0.0	0.0	0.0
Y (3)	0.0	0.0	0.0	5.0000000000D-01	0.0	0.0
X (3)	0.0	0.0	0.0	0.0	5.0000000000D-01	0.0
Z (4)	0.0	0.0	0.0	0.0	-5.0000000000D-01	0.0
Y (4)	0.0	0.0	0.0	0.0	0.0	-5.0000000000D-01
X (4)	4.0824829046D-01	-2.8867513459D-01	5.0000000000D-01	0.0	0.0	0.0
Z (5)	0.0	0.0	0.0	-5.0000000000D-01	0.0	0.0
Y (5)	4.0824829046D-01	-2.8867513459D-01	-5.0000000000D-01	0.0	0.0	0.0
X (5)	0.0	0.0	0.0	0.0	0.0	-5.0000000000D-01
Z (6)	4.0824829046D-01	5.7735026919D-01	0.0	0.0	0.0	0.0
Y (6)	0.0	0.0	0.0	-5.0000000000D-01	0.0	0.0
X (6)	0.0	0.0	0.0	0.0	-5.0000000000D-01	0.0

Fig. 15 (Contd.)

INAL SYMMETRY VECTORS FOR ATOM-TYPE 2, L = 1

LUMNS 7 TO 12

GENVALUES

	2.2196766423D 02	2.2196766423D 02	2.2196766423D 02	2.2532419478D 02	2.2532419478D 02	2.2532419478D 02

GENVECTORS

Z (1)	0.0	-5.0000000000D-01	0.0	0.0	0.0	-5.0000000000D-01
Y (1)	5.0000000000D-01	0.0	0.0	0.0	5.0000000000D-01	0.0
X (1)	0.0	0.0	0.0	0.0	0.0	0.0
Z (2)	0.0	0.0	5.0000000000D-01	0.0	0.0	5.0000000000D-01
Y (2)	0.0	0.0	0.0	0.0	0.0	0.0
X (2)	-5.0000000000D-01	0.0	0.0	-5.0000000000D-01	0.0	0.0
Z (3)	0.0	0.0	0.0	0.0	0.0	0.0
Y (3)	0.0	0.0	-5.0000000000D-01	0.0	-5.0000000000D-01	0.0
X (3)	0.0	5.0000000000D-01	0.0	5.0000000000D-01	0.0	0.0
Z (4)	0.0	5.0000000000D-01	0.0	0.0	0.0	-5.0000000000D-01
Y (4)	-5.0000000000D-01	0.0	0.0	0.0	5.0000000000D-01	0.0
X (4)	0.0	0.0	0.0	0.0	0.0	0.0
Z (5)	0.0	0.0	-5.0000000000D-01	0.0	0.0	5.0000000000D-01
Y (5)	0.0	0.0	0.0	0.0	0.0	0.0
X (5)	5.0000000000D-01	0.0	0.0	-5.0000000000D-01	0.0	0.0
Z (6)	0.0	0.0	0.0	0.0	0.0	0.0
Y (6)	0.0	0.0	5.0000000000D-01	0.0	-5.0000000000D-01	0.0
X (6)	0.0	-5.0000000000D-01	0.0	5.0000000000D-01	0.0	0.0

Fig. 15 (Contd.)

FINAL SYMMETRY VECTORS FOR ATOM-TYPE 2, L = 1

COLUMNS 13 TO 18

EIGENVALUES

	1.1962162381D 03	1.1962162381D 03	1.1962162381D 03	2.2700108659D 02	2.2700108659D 02	2.2700108659D 02
EIGENVECTORS						
Z (1)	5.0000000000D-01	0.0	0.0	0.0	0.0	0.0
Y (1)	0.0	5.0000000000D-01	0.0	0.0	0.0	0.0
X (1)	0.0	0.0	0.0	0.0	0.0	7.0710678119D-01
Z (2)	5.0000000000D-01	0.0	0.0	0.0	0.0	0.0
Y (2)	0.0	0.0	0.0	0.0	7.0710678119D-01	0.0
X (2)	0.0	0.0	5.0000000000D-01	0.0	0.0	0.0
Z (3)	0.0	0.0	0.0	7.0710678119D-01	0.0	0.0
Y (3)	0.0	5.0000000000D-01	0.0	0.0	0.0	0.0
X (3)	0.0	0.0	5.0000000000D-01	0.0	0.0	0.0
Z (4)	5.0000000000D-01	0.0	0.0	0.0	0.0	0.0
Y (4)	0.0	5.0000000000D-01	0.0	0.0	0.0	0.0
X (4)	0.0	0.0	0.0	0.0	0.0	7.0710678119D-01
Z (5)	5.0000000000D-01	0.0	0.0	0.0	0.0	0.0
Y (5)	0.0	0.0	0.0	0.0	7.0710678119D-01	0.0
X (5)	0.0	0.0	5.0000000000D-01	0.0	0.0	0.0
Z (6)	0.0	0.0	0.0	7.0710678119D-01	0.0	0.0
Y (6)	0.0	5.0000000000D-01	0.0	0.0	0.0	0.0
X (6)	0.0	0.0	5.0000000000D-01	0.0	0.0	0.0

GULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTOR

1 3 4 5 6 7 8 9 10 15 16 17 18 19 20 18 19 20

. OF TIMES EACH REPRESENTATION OCCURS

13 0 7 4 2 0 0 0 2 13

Fig. 15 (Contd.)

LECTION RULES FOR DIRECT PRODUCTS OF IRREDUCIBLE REPRESENTATIONS

REP.1 X REP.2	REP.3 --> A,B(G) 1	A,B(G) 2	E(G) 3	T(G) 4	T(G) 5
1(A,B(G)) X 1(A,B(G))	1	0	0	0	0
1(A,B(G)) X 1(A,B(G))	0	1	0	0	0
1(A,B(G)) X 2(A,B(G))	1	0	0	0	0
3(E(G)) X 1(A,B(G))	0	0	1	0	0
3(E(G)) X 2(A,B(G))	0	0	-1	0	0
3(E(G)) X 3(E(G))	1	-1	1	0	0
4(T(G)) X 1(A,B(G))	0	0	0	1	0
4(T(G)) X 2(A,B(G))	0	0	0	0	1
4(T(G)) X 3(E(G))	0	0	0	1	1
4(T(G)) X 4(T(G))	1	0	1	1	-1
5(T(G)) X 1(A,B(G))	0	0	0	0	1
5(T(G)) X 2(A,B(G))	0	0	0	1	0
5(T(G)) X 3(E(G))	0	0	0	1	1
5(T(G)) X 4(T(G))	0	1	1	-1	1
5(T(G)) X 5(T(G))	1	0	1	1	-1

VECTOR COUPLING COEFFICIENTS

REPRESENTATIONS 1 X 1 --> 1

IANG1	IANG2	IANG3 --> 1
1	1	1.000000000

Fig. 15 (Contd.)

REPRESENTATIONS 2 X 2 --> 1

IANG1	IANG2	IANG3 --> 1
2	2	1.000000000

REPRESENTATIONS 3 X 3 --> 1

IANG1	IANG2	IANG3 --> 1
3	3	-0.707106781
3	4	0.0
4	3	0.0
4	4	-0.707106781

REPRESENTATIONS 4 X 4 --> 1

IANG1	IANG2	IANG3 --> 1
5	5	-0.577350269
5	6	0.0
5	7	0.0
6	5	0.0
6	6	-0.577350269
6	7	0.0
7	5	0.0
7	6	0.0
7	7	-0.577350269

PRESENTATIONS 5 X 5 --> 1

IANG1	IANG2	IANG3 --> 1
8	8	0.577350269
8	9	0.0
8	10	0.0
9	8	0.0
9	9	0.577350269
9	10	0.0
10	8	0.0
10	9	0.0
10	10	0.577350269

Fig. 15 (Contd.)

REPRESENTATIONS 2 X 1 --> 2

IANG1	IANG2	IANG3 --> 2
2	1	1.000000000

REPRESENTATIONS 3 X 3 --> 2

IANG1	IANG2	IANG3 --> 2
3	3	0.0
3	4	0.707106781
4	3	-0.707106781
4	4	0.0

REPRESENTATIONS 5 X 4 --> 2

IANG1	IANG2	IANG3 --> 2
8	5	0.0
8	6	0.0
8	7	0.577350269
9	5	0.0
9	6	0.577350269
9	7	0.0
10	5	0.577350269
10	6	0.0
10	7	0.0

REPRESENTATIONS 3 X 1 --> 3

IANG1	IANG2	IANG3 --> 3	4
3	1	-1.000000000	0.0
4	1	0.0	-1.000000000

Fig. 15 (Contd.)

REPRESENTATIONS 3 X 2 --> 3

IANG1	IANG2	IANG3 --> 3	4
3	2	0.0	1.000000000
4	2	-1.000000000	0.0

REPRESENTATIONS 3 X 3 --> 3

IANG1	IANG2	IANG3 --> 3	4
3	3	-0.707106781	0.0
3	4	0.0	0.707106781
4	3	0.0	0.707106781
4	4	0.707106781	0.0

REPRESENTATIONS 4 X 4 --> 3

IANG1	IANG2	IANG3 --> 3	4
5	5	-0.408248290	0.707106781
5	6	0.0	0.0
5	7	0.0	0.0
6	5	0.0	0.0
6	6	-0.408248290	-0.707106781
6	7	0.0	0.0
7	5	0.0	0.0
7	6	0.0	0.0
7	7	0.816496581	0.0

EPRESENTATIONS 5 X 4 --> 3

IANG1	IANG2	IANG3 --> 3	4
8	5	0.0	0.0
8	6	0.0	0.0
8	7	0.0	0.816496581
9	5	0.0	0.0
9	6	0.707106781	-0.408248290
9	7	0.0	0.0
10	5	-0.707106781	-0.408248290
10	6	0.0	0.0
10	7	0.0	0.0

Fig. 15 (Contd.)

RESENTATIONS 5 X 5 --> 3

IANG3 -->

IANG1	IANG2	3	4
8	8	0.816496581	0.0
8	9	0.0	0.0
8	10	0.0	0.0
9	8	0.0	0.0
9	9	-0.408248290	-0.707106781
9	10	0.0	0.0
10	8	0.0	0.0
10	9	0.0	0.0
10	10	-0.408248290	0.707106781

REPRESENTATIONS 4 X 1 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
5	1	-1.000000000	0.0	0.0
6	1	0.0	-1.000000000	0.0
7	1	0.0	0.0	-1.000000000

REPRESENTATIONS 4 X 3 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
5	3	-0.500000000	0.0	0.0
5	4	0.866025404	0.0	0.0
6	3	0.0	-0.500000000	0.0
6	4	0.0	-0.866025404	0.0
7	3	0.0	0.0	1.000000000
7	4	0.0	0.0	0.0

REPRESENTATIONS 4 X 4 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
5	5	0.0	0.0	0.0
5	6	0.0	0.0	-0.707106781
5	7	0.0	-0.707106781	0.0
6	5	0.0	0.0	-0.707106781
6	6	0.0	0.0	0.0
6	7	-0.707106781	0.0	0.0
7	5	0.0	-0.707106781	0.0
7	6	-0.707106781	0.0	0.0
7	7	0.0	0.0	0.0

Fig. 15 (Contd.)

REPRESENTATIONS 5 X 2 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
8	2	0.0	0.0	1.000000000
9	2	0.0	1.000000000	0.0
10	2	1.000000000	0.0	0.0

REPRESENTATIONS 5 X 3 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
8	3	0.0	0.0	0.0
8	4	0.0	0.0	1.000000000
9	3	0.0	0.866025404	0.0
9	4	0.0	-0.500000000	0.0
10	3	-0.866025404	0.0	0.0
10	4	-0.500000000	0.0	0.0

REPRESENTATIONS 5 X 4 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
8	5	0.0	0.707106781	0.0
8	6	-0.707106781	0.0	0.0
8	7	0.0	0.0	0.0
9	5	0.0	0.0	-0.707106781
9	6	0.0	0.0	0.0
9	7	0.707106781	0.0	0.0
10	5	0.0	0.0	0.0
10	6	0.0	0.0	0.707106781
10	7	0.0	-0.707106781	0.0

EPRESENTATIONS 5 X 5 --> 4

IANG3 -->

IANG1	IANG2	5	6	7
8	8	0.0	0.0	0.0
8	9	-0.707106781	0.0	0.0
8	10	0.0	-0.707106781	0.0
9	8	-0.707106781	0.0	0.0
9	9	0.0	0.0	0.0
9	10	0.0	0.0	-0.707106781
10	8	0.0	-0.707106781	0.0
10	9	0.0	0.0	-0.707106781
10	10	0.0	0.0	0.0

Fig. 15 (Contd.)

EPRESENTATIONS 4 X 2 --> 5

IANG3 -->

IANG1	IANG2	8	9	10
5	2	0.0	0.0	1.000000000
6	2	0.0	1.000000000	0.0
7	2	1.000000000	0.0	0.0

EPRESENTATIONS 4 X 3 --> 5

IANG3 -->

IANG1	IANG2	8	9	10
5	3	0.0	0.0	-0.866025404
5	4	0.0	0.0	-0.500000000
6	3	0.0	0.866025404	0.0
6	4	0.0	-0.500000000	0.0
7	3	0.0	0.0	0.0
7	4	1.000000000	0.0	0.0

REPRESENTATIONS 4 X 4 --> 5

IANG3 -->

IANG1	IANG2	8	9	10
5	5	0.0	0.0	0.0
5	6	-0.707106781	0.0	0.0
5	7	0.0	0.707106781	0.0
6	5	0.707106781	0.0	0.0
6	6	0.0	0.0	0.0
6	7	0.0	0.0	-0.707106781
7	5	0.0	-0.707106781	0.0
7	6	0.0	0.0	0.707106781
7	7	0.0	0.0	0.0

REPRESENTATIONS 5 X 1 --> 5

IANG3 -->

IANG1	IANG2	8	9	10
8	1	1.000000000	0.0	0.0
9	1	0.0	1.000000000	0.0
10	1	0.0	0.0	1.000000000

Fig. 15 (Contd.)

REPRESENTATIONS 5 X 3 --> 5

		IANG3 -->		
IANG1	IANG2	8	9	10
8	3	1.000000000	0.0	0.0
8	4	0.0	0.0	0.0
9	3	0.0	-0.500000000	0.0
9	4	0.0	-0.866025404	0.0
10	3	0.0	0.0	-0.500000000
10	4	0.0	0.0	0.866025404

REPRESENTATIONS 5 X 4 --> 5

		IANG3 -->		
IANG1	IANG2	8	9	10
8	5	0.0	-0.707106781	0.0
8	6	0.0	0.0	-0.707106781
8	7	0.0	0.0	0.0
9	5	-0.707106781	0.0	0.0
9	6	0.0	0.0	0.0
9	7	0.0	0.0	-0.707106781
10	5	0.0	0.0	0.0
10	6	-0.707106781	0.0	0.0
10	7	0.0	-0.707106781	0.0

REPRESENTATIONS 5 X 5 --> 5

		IANG3 -->		
IANG1	IANG2	8	9	10
8	8	0.0	0.0	0.0
8	9	0.0	0.0	-0.707106781
8	10	0.0	0.707106781	0.0
9	8	0.0	0.0	0.707106781
9	9	0.0	0.0	0.0
9	10	-0.707106781	0.0	0.0
10	8	0.0	-0.707106781	0.0
10	9	0.707106781	0.0	0.0
10	10	0.0	0.0	0.0

Fig. 15 (Contd.)

APPENDIX

Source-code Listings

INDEX FOR PROGRAM: GPTHEORY PAGE I
DATE = 4/07/71

ROUTINE	PAGE	ROUTINE	PAGE	ROUTINE	PAGE	ROUTINE	PAGE
ADAPT	16						
ALPTAB	8						
ANTSYM	32						
APPLY	76						
AUFUN	40						
BASFUN	62						
BASIS	48						
CLEAN	41						
DEGEN	25						
EIGEN	23						
FILOUT	77						
GRID	9						
GROUP	11						
INBAS	51						
LINCOM	27						
LINEUP	33						
MAIN	1						
MATPRT	22						
MOLCUL	2						
ODDGA	70						
ORBTLS	58						
ORTHOG	72						
PHASER	36						
PROSYM	13						
PSYM	18						
PTGRUP	46						
SCAN	43						
SORT1	26						
SYMB3J	31						
SYMSUM	44						
THETAX	21						
THLMK	20						
VALUE	63						
VECUPL	65						
VIBROT	75						
VPRINT	73						
WIG3J	29						

COMMON CROSS-REFERENCE TABLE FOR: GPTHEORY PAGE II

/ALP/		WIG3J	29	VPRINT	73
PSYM	18				
THETAX	21	/GIRD/		/PHASE/	
THLMK	20	GRID	9	LINEUP	33
		MOLCUL	2	PHASER	36
/ALP1/		PROSYM	13	PROSYM	13
ALPTAB	8	PSYM	18		
THETAX	21	THLMK	20	/PROTCO/	
WIG3J	29			LINEUP	33
		/OUT8/		ODDGA	70
/ASYMI/		BASIS	48	PHASER	36
ANTSYM	32	FILOUT	77	PROSYM	13
PROSYM	13	INBAS	51	VECUPL	65
		ORBTLS	58		
/CONTRL/				/SCAN1/	
ADAPT	16	/OUT1/		CLEAN	41
BASIS	48	BASFUN	62	SCAN	43
GRID	9	BASIS	48		
GROUP	11	FILOUT	77	/UNGRAD/	
INBAS	51	GRID	9	AUFUN	40
LINCOM	27	INBAS	51	PHASER	36
LINEUP	33	MOLCUL	2		
MAIN	1	ORBTLS	58	/WIGNER/	
MOLCUL	2	PSYM	18	SYMB3J	31
ODDGA	70			VECUPL	65
ORBTLS	58	/OUT2/		WIG3J	29
ORTHOG	72	BASIS	48		
PHASER	36	FILOUT	77	/WORK1/	
PROSYM	13	ORBTLS	58	ADAPT	16
PSYM	18			LINEUP	33
PTGRUP	46	/OUT3/		ODDGA	70
SYMSUM	44	ADAPT	16	ORBTLS	58
VECUPL	65	CLEAN	41	PHASER	36
		FILOUT	77	PROSYM	13
/FILES/		GROUP	11	VECUPL	65
ADAPT	16	LINEUP	33		
BASIS	48	MOLCUL	2	/WORK2/	
INBAS	51	ODDGA	70	BASFUN	62
LINEUP	33	ORBTLS	58	ORBTLS	58
MAIN	1	PHASER	36	VALUE	63
MATPRT	22	PROSYM	13		
MOLCUL	2	PTGRUP	46	/XYZ/	
ODDGA	70	SYMSUM	44	BASFUN	62
ORBTLS	58	VECUPL	65	GRID	9
PHASER	36	VPRINT	73		
PROSYM	13				
PTGRUP	46	/OUT4/			
SYMSUM	44	FILOUT	77		
VECUPL	65	ORTHOG	72		
VPRINT	73	VECUPL	65		

```
C**** PROGRAM GPTHEORY
C**** AUTOMATION OF MOLECULAR POINT GROUP THEORY
C**** T.D. BOUMAN AND G.L. GOODMAN
C**** CHEMISTRY DIVISION, ARGONNE NATIONAL LABORATORY
C**** 9700 S. CASS AVE., ARGONNE, ILLINOIS 60439
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)
      COMMON/FILES/INPUT,IOUT
      INPUT=5
      IOUT=6
      REWIND 20
      CALL NOFLOW
      ISTOP=0
C**** INPUT AND MOLECULAR GEOMETRY ANALYSIS
   10 CALL MOLCUL (IORDER)
      IF (ISTOP.EQ.1) GO TO 10
      IF (IORDER.EQ.1) GO TO 5
C**** DETERMINATION OF POINT GROUP PROPERTIES
    1 CALL GROUP(ISUM,JSUM)
      CALL SYMSUM(ISUM,JSUM)
      IF (ISTOP.EQ.0) GO TO 4
      IF (INTER.EQ.1) GO TO 10
      INTER=1
      ISTOP=0
      WRITE (IOUT,20)
   20 FORMAT (//' RESTART WITH INTERMEDIATE OUTPUT'/)
      GO TO 1
C**** SYMMETRY-ADAPTATION OF ATOMIC BASIS SET
    4 IF(LFLOW(1).EQ.0) GO TO 7
    5 CALL BASIS
      IF (ISTOP.EQ.1) GO TO 10
      IF (IORDER.EQ.1) GO TO 3
      CALL ORBTLS
      IF (ISTOP.EQ.0) GO TO 7
      IF (INTER.EQ.1) GO TO 10
      INTER=1
      ISTOP=0
      WRITE (IOUT,20)
      GO TO 1
    7 IF(LFLOW(2).EQ.0) GO TO 8
C**** VIBRATION-ROTATION SYMMETRY ANALYSIS
      CALL VIBROT
      IF (ISTOP.EQ.0) GO TO 8
      IF (INTER.EQ.1) GO TO 10
      INTER=1
      ISTOP=0
      WRITE (IOUT,20)
      GO TO 1
C**** FINITE POINT GROUP VECTOR-COUPLING COEFFICIENTS
    8 IF (LFLOW(3).EQ.0) GO TO 3
      CALL VECUPL
    3 CONTINUE
C**** INTERFACE TO APPLICATIONS PROGRAMS
      CALL APPLY
      CALL FILOUT
      GO TO 10
      END
```



```
      SUBROUTINE MOLCUL (IORDER)
C**** READS AND ANALYZES DATA PERTAINING TO NUCLEAR GEOMETRY
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*4 SYMBOL
      INTEGER*4 ELMNT,XOP,YOP,ZOP
      DIMENSION XX(30),YY(30),ZZ(30)
      DIMENSION IDENT (20),P(3),FC(3),IOP(3),RCCT(10),NUC(30),IHOL(6),
     1SYMBOL(103),IASYMB(30)
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE,
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30),
     2 INVAT(30)
      COMMON/OUT3/NREP,NOESA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)
      COMMON/GIRD/RGRID(30,3),CTHGR(1447),PHIGR(1447),NGRID,NDIV
      COMMON/FILES/INPUT,IOUT
      DATA IB/'    '/,HYES/3HYES/,HNO/3HNO /
      DATA IHOL/'COS','SIN','1/','SQRT','*','/'/
      DATA SYMBOL/'H ','HE','LI','BE','B ','C ','N ','O ','F ','NE','NA
     1,'MG','AL','SI','P ','S ','CL','AR','K ','CA','SC','TI','V ','CR','MOLC0200
     2 'MN','FE','CO','NI','CU','ZN','GA','GE','AS','SE','BR','KR','RB','MOLC0210
     3 'SR','Y ','ZR','NB','MO','TC','RU','RH','PD','AG','CD','IN','SN','MOLC0220
     4 'SB','TE','I ','XE','CS','BA','LA','CE','PR','ND','PM','SM','EU','MOLC0230
     5 'GD','TB','DY','HO','ER','TM','YB','LU','HF','TA','W ','RE','OS','MOLC0240
     6 'IR','PT','AU','HG','TL','PB','BI','PO','AT','RN','FR','RA','AC','MOLC0250
     7 'TH','PA','U ','NP','PU','AM','CM','BK','CF','ES','FM','MD','NO','MOLC0260
     8 'LW'/
      NAMELIST/PARAMS/NATOMS,CRTZRO,CRTDEG,CRTTYP,NTCHEK,INTER,SCALE,
     1 NDIV,LFLOW
      NAMELIST/ATOM/ELMNT,X,Y,Z,XOP,YOP,ZOP,FX,FY,FZ
      CALEND=DATE1(DUMMY)
      PI=3.141592653589793
      DR=PI/180.D0
      IF (ISTOP.EQ.1) GO TO 203
C**** INPUT OF MOLECULAR GEOMETRY AND JOB PARAMETERS
      READ (INPUT,300) IDENT,JOB
  300 FORMAT (19A4,A3,A1)
      IF (JOB.EQ.IHOL(6)) STOP
      WRITE (IOUT,302)
  302 FORMAT (1H1,35X,'****  GPTHEORY - AUTOMATED MOLECULAR POINT GROUP
     1THEORY  ****'/47X,'THOMAS D. BOUMAN AND GORDON L. GOODMAN'/
     2 42X,'CHEMISTRY DIVISION, ARGONNE NATIONAL LABORATORY'/54X,
     3 'ARGONNE, ILLINOIS 60439'//)
 1000 WRITE (IOUT,301) IDENT,CALEND
  301 FORMAT(' PROBLEM TITLE --  ',20A4,A8//)
      LVAL1=0
      LVAL2=0
      IORDER=0
      ISTOP=0
      INTER=0
      NTCHEK=1
      NDIV=24
      DO 99 I=5,10
   99 LFLOW(I)=0
      DO 98 I=1,4
   98 LFLOW(I)=1
      SCALE=1.D0
      CRTZRO=1.D-8
```

```
      CRTDEG=1.00000001                                                 MOLC0590
      CRTTYP=1.0D-5                                                     MOLC0600
      READ (INPUT,PARAMS)                                               MOLC0610
      IF (NATOMS.GT.30) GO TO 150                                       MOLC0620
      IF (NATOMS.GT.2) GO TO 23                                         MOLC0630
      IF (NATOMS.EQ.0) GO TO 203                                        MOLC0640
      WRITE (IOUT,20)                                                   MOLC0650
   20 FORMAT (/' MOLECULE IS DIATOMIC.')                                MOLC0660
      WRITE (IOUT,101)                                                  MOLC0670
  101 FORMAT (/' ONLY THE REPRESENTATIONS SPANNED BY THE INPUT L-VALUES MOLC0680
     1WILL BE CHARACTERIZED.'/' NO VECTOR-COUPLING COEFFICIENTS WILL BE MOLC0690
     2COMPUTED.')                                                       MOLC0700
   23 WRITE(IOUT,59) NATOMS,NTCHEK                                      MOLC0710
   59 FORMAT (' INPUT DESCRIPTION OF MOLECULE'/1X,28(1H-)/5X,'NUMBER OF MOLC0720
     1ATOMS = ',I2/5X,'ASSUMED NO. OF EQUIVALENT SETS (ATOM TYPES) = ', MOLC0730
     2 I2/)                                                             MOLC0740
      WRITE(IOUT,70)                                                    MOLC0750
   70 FORMAT (5X,'CARTESIAN COORDINATES (IN FORM  X = A*FUNC(B)  )'//5X,MOLC0760
     1 ' ATOM   CHEMICAL',11X,'X',23X,'Y',23X,'Z'/5X,'NUMBER   SYMBOL', MOLC0770
     2 2X,3(4X,'A',4X,'FUNC',4X,'B',6X)/)                               MOLC0780
      DO 220 I=1,NATOMS                                                 MOLC0790
      X=0.                                                              MOLC0800
      Y=0.                                                              MOLC0810
      Z=0.                                                              MOLC0820
      XOP=IB                                                            MOLC0830
      YOP=IB                                                            MOLC0840
      ZOP=IB                                                            MOLC0850
      FX=1.D0                                                           MOLC0860
      FY=1.D0                                                           MOLC0870
      FZ=1.D0                                                           MOLC0880
      READ (INPUT,ATOM)                                                 MOLC0890
      WRITE (IOUT,71) I,ELMNT,X,XOP,FX,Y,YOP,FY,Z,ZOP,FZ                MOLC0900
   71 FORMAT(7X,I2,7X,A2,3X,3(F8.4,1X,A4,F8.3,3X))                      MOLC0910
      IASYMB(I)=ELMNT                                                   MOLC0920
      POS(1,I)=X                                                        MOLC0930
      POS(2,I)=Y                                                        MOLC0940
      POS(3,I)=Z                                                        MOLC0950
      IOP(1)=XOP                                                        MOLC0960
      IOP(2)=YOP                                                        MOLC0970
      IOP(3)=ZOP                                                        MOLC0980
      FC(1)=FX                                                          MOLC0990
      FC(2)=FY                                                          MOLC1000
      FC(3)=FZ                                                          MOLC1010
      DO 222 N=1,103                                                    MOLC1020
      NCH=N                                                             MOLC1030
      IF (IASYMB(I).EQ.SYMBOL(N)) GO TO 223                             MOLC1040
  222 CONTINUE                                                          MOLC1050
      WRITE (IOUT,224) IASYMB(I),I                                      MOLC1060
  224 FORMAT (/' THE CHEMICAL SYMBOL',A2,' INPUT ON ATOM-PARAMETER CARD MOLC1070
     1NO. ',I2,' IS INCORRECTLY PUNCHED.')                              MOLC1080
      GO TO 203                                                         MOLC1090
  223 NUC(I)=NCH                                                        MOLC1100
      DO 220 J=1,3                                                      MOLC1110
      IF (FC(J).EQ.0) FC(J)=1.D0                                        MOLC1120
      IF (IOP(J).EQ.IHOL(1)) FC(J)=DCOS(DR*FC(J))                       MOLC1130
      IF (IOP(J).EQ.IHOL(2)) FC(J)=DSIN(DR*FC(J))                       MOLC1140
      IF (IOP(J).EQ.IHOL(3)) FC(J)=1.D0/FC(J)                           MOLC1150
      IF (IOP(J).EQ.IHOL(4)) FC(J)=DSQRT(FC(J))                         MOLC1160
```



```
  220 POS(J,I)=POS(J,I)*FC(J)*SCALE                                     MOLC1170
      IF (NATOMS.LE.2) GO TO 203                                        MOLC1180
      WRITE (IOUT,72) CRTZRO,CRTDEG,CRTTYP,NDIV,SCALE                   MOLC1190
   72 FORMAT(//' THRESHOLDS AND CONTROL PARAMETERS'/1X,34(1H-)/5X,'ZERO MOLC1200
     1THRESHOLD = ',1PE8.2/5X,'DEGENERACY TEST (RATIO OF EIGENVALUES) = MOLC1210
     2',0PF11.8/5X,'ATOM EQUIVALENCE THRESHOLD = ',1PE8.2/5X,'NO. OF GRIMOLC1220
     3D POINTS PER ATOM = ',I3/5X,'SCALE FACTOR MULTIPLYING COORDINATES MOLC1230
     4= ',0PF10.5)                                                      MOLC1235
      WRITE(IOUT,73)                                                    MOLC1240
   73 FORMAT (5X,'PROGRAM OPTIONS SELECTED -'/10X,'POINT GROUP PROPERTIEMOLC1250
     1S')                                                               MOLC1260
      IF (INTER.EQ.1) WRITE(IOUT,74)                                    MOLC1270
   74 FORMAT(10X,'INTERMEDIATE PRINTOUT')                               MOLC1280
      IF (LFLOW(1).EQ.1) WRITE(IOUT,75)                                 MOLC1290
   75 FORMAT (10X,'SYMMETRY-ADAPTED ATOMIC BASIS SET')                  MOLC1300
      IF (LFLOW(4).EQ.1) WRITE(IOUT,76)                                 MOLC1310
   76 FORMAT(12X,'MINIMAL SLATER BASIS')                                MOLC1320
      IF (LFLOW(4).EQ.2) WRITE(IOUT,77)                                 MOLC1330
   77 FORMAT(12X,'DOUBLE-ZETA SLATER BASIS')                            MOLC1340
      IF (LFLOW(4).EQ.9) WRITE(IOUT,78)                                 MOLC1350
   78 FORMAT(12X,'BASIS SET READ IN')                                   MOLC1360
      IF (LFLOW(2).EQ.1) WRITE(IOUT,79)                                 MOLC1370
   79 FORMAT(10X,'VIBRATION-ROTATION SYMMETRY')                         MOLC1380
      IF (LFLOW(3).EQ.1) WRITE(IOUT,80)                                 MOLC1390
   80 FORMAT(10X,'VECTOR-COUPLING COEFFICIENTS')                        MOLC1400
C**** TRANSLATION TO CENTER OF NUCLEAR CHARGE COORDINATE SYSTEM         MOLC1410
 1005 FN=0.                                                             MOLC1420
      DO 109 J=1,3                                                      MOLC1430
  109 P(J)=0.                                                           MOLC1440
      DO 3 I=1,NATOMS                                                   MOLC1450
      FNI=NUC(I)                                                        MOLC1460
      DO 110 J=1,3                                                      MOLC1470
  110 P(J)=P(J)+POS(J,I)*FNI                                            MOLC1480
    3 FN=FN+FNI                                                         MOLC1490
      DO 111 J=1,3                                                      MOLC1500
  111 P(J)=P(J)/FN                                                      MOLC1510
      DO 400 I=1,NATOMS                                                 MOLC1520
      R2=0.                                                             MOLC1530
      DO 401 J=1,3                                                      MOLC1540
      POS(J,I)=POS(J,I)-P(J)                                            MOLC1550
  401 R2=R2+POS(J,I)**2                                                 MOLC1560
      RADIUS(I)=DSQRT(R2)                                               MOLC1570
      XX(I)=POS(1,I)                                                    MOLC1580
      YY(I)=POS(2,I)                                                    MOLC1590
      ZZ(I)=POS(3,I)                                                    MOLC1600
  400 CONTINUE                                                          MOLC1610
C**** SPHERICAL POLAR COORDINATES IN C.C. SYSTEM                        MOLC1620
      DO 4 I=1,NATOMS                                                   MOLC1630
      THETA(I)=0.                                                       MOLC1640
      PHI(I)=0.                                                         MOLC1650
      W=POS(3,I)                                                        MOLC1660
      IF (RADIUS(I).LE.CRTZRO) GO TO 4                                  MOLC1670
      IF (DABS(DABS(W)-RADIUS(I)).GT.1.D-7) GO TO 305                   MOLC1680
      IF (POS(3,I).GE.0.) GO TO 4                                       MOLC1690
      THETA(I)=PI                                                       MOLC1700
      GO TO 4                                                           MOLC1710
  305 THETA(I)=DARCOS(POS(3,I)/RADIUS(I))                               MOLC1720
      IF (DABS(POS(2,I)).LE.CRTZRO.AND.DABS(POS(1,I)).LE.CRTZRO) GO TO 4MOLC1730
```

```
      PHI(I)=DATAN2(POS(2,I),POS(1,I))                                  MOLC1740
      IF (PHI(I).LT.0.) PHI(I)=PHI(I)+2.D0*PI                           MOLC1750
    4 CONTINUE                                                          MOLC1760
C**** MOLECULAR GEOMETRY ANALYSIS                                       MOLC1770
      WRITE(IOUT,81)                                                    MOLC1780
   81 FORMAT(1H1,'MOLECULAR GEOMETRY ANALYSIS'/1X,27(1H-)//)            MOLC1790
      WRITE (IOUT,205)                                                  MOLC1800
  205 FORMAT (/25X,'ATOM',3X,'CHEMICAL',4X,'ATOMIC',6X,'CARTESIAN COORDI MOLC1810
     1NATES IN CENTER OF CHARGE SYSTEM')                                MOLC1820
      WRITE (IOUT,60)                                                   MOLC1830
   60 FORMAT(24X,'NUMBER',3X,'SYMBOL',3X,'NUMBER (Z)',10X,'X',16X,'Y',  MOLC1840
     116X,'Z'/)                                                         MOLC1850
  204 FORMAT (26X,I2,7X,A2,8X,I2,4X,3F17.8/)                            MOLC1860
      WRITE (IOUT,204) (I,IASYMB(I),NUC(I),(POS(J,I),J=1,3),I=1,NATOMS) MOLC1870
      WRITE (IOUT,50)                                                   MOLC1880
   50 FORMAT (/25X,'ATOM',3X,'CHEMICAL',4X,'ATOMIC',6X,'SPHERICAL COORDI MOLC1890
     1NATES IN CENTER OF CHARGE SYSTEM'/24X,'NUMBER',3X,'SYMBOL',3X,'NUMMOLC1900
     2BER (Z)',10X,'R',14X,'THETA',13X,'PHI'/)                          MOLC1910
      WRITE (IOUT,204) (I,IASYMB(I),NUC(I),RADIUS(I),THETA(I),PHI(I),   MOLC1920
     1 I=1,NATOMS)                                                      MOLC1930
C**** TEST FOR LINEAR MOLECULE                                          MOLC1940
      IF (NATOMS.EQ.2) GO TO 310                                        MOLC1950
      IS=2                                                              MOLC1960
      IF (RADIUS(1).LE.CRTZRO) IS=3                                     MOLC1970
      LIN=1                                                             MOLC1980
      IS1=IS-1                                                          MOLC1990
      DO 104 I=IS,NATOMS                                                MOLC2000
      PHIPI=DMOD(PHI(I)+PI,2.D0*PI)                                     MOLC2010
      IF (DSIN(THETA(I)).LE.1.D-7) PHIPI=0.                             MOLC2020
      IF ((DABS(THETA(IS1)-THETA(I)).LE.CRTZRO.AND.DABS(PHI(IS1)-PHI(I)))MOLC2030
     1 .LE.CRTZRO)   .OR.   (DABS(THETA(IS1)-PI+THETA(I)).LE.CRTZRO      MOLC2040
     2 .AND.DABS(DABS(PHI(IS1)-PHIPI)).LE.CRTZRO)) GO TO 104            MOLC2050
      LIN=0                                                             MOLC2060
  104 CONTINUE                                                          MOLC2070
      IF (LIN.EQ.0) GO TO 21                                            MOLC2080
      WRITE (IOUT,105)                                                  MOLC2090
  105 FORMAT (/' MOLECULE IS LINEAR.')                                  MOLC2100
      WRITE (IOUT,101)                                                  MOLC2110
   10 READ (INPUT,315) LVAL1,LVAL2                                      MOLC2120
   15 FORMAT (2I5)                                                      MOLC2130
      LFLOW(3) = 0                                                      MOLC2140
      IF (MAX0(LVAL1,LVAL2).GT.16) GO TO 150                            MOLC2150
   ** GENERATE AND LABEL ATOM-TYPES, OTHER TABLES                       MOLC2160
   21 NUMT(1)=1                                                         MOLC2170
      NATYPE=1                                                          MOLC2180
      RCCT(1)=RADIUS(1)                                                 MOLC2190
      NUCT(1)=NUC(1)                                                    MOLC2200
      ITSYMB(1)=IASYMB(1)                                               MOLC2210
      ITYPT(1)=1                                                        MOLC2220
      ITYPI(1)=1                                                        MOLC2230
      ITYPA(1,1)=1                                                      MOLC2240
      DO 24 IA=2,NATOMS                                                 MOLC2250
      DO 25 IT=1,NATYPE                                                 MOLC2260
      IF (DABS(RADIUS(IA)-RCCT(IT)).GT.CRTTYP.OR.NUC(IA).NE.NUCT(IT))   MOLC2270
     1 GO TO 25                                                         MOLC2280
   11 NUMB=NUMT(IT)+1                                                   MOLC2290
      NUMT(IT)=NUMB                                                     MOLC2300
      ITYPT(IA)=IT                                                      MOLC2310
```



```
      ITYPI(IA)=NUMB                                                    MOLC2320
      ITYPA(NUMB,IT)=IA                                                 MOLC2330
      GO TO 24                                                          MOLC2340
   25 CONTINUE                                                          MOLC2350
      NATYPE=NATYPE+1                                                   MOLC2360
      NUCT(NATYPE)=NUC(IA)                                              MOLC2370
      ITSYMB(NATYPE)=IASYMB(IA)                                         MOLC2380
      RCCT(NATYPE)=RADIUS(IA)                                           MOLC2390
      NUMT(NATYPE)=1                                                    MOLC2400
      ITYPT(IA)=NATYPE                                                  MOLC2410
      ITYPI(IA)=1                                                       MOLC2420
      ITYPA(1,NATYPE)=IA                                                MOLC2430
   24 CONTINUE                                                          MOLC2440
      IF (NATYPE.EQ.NATOMS) IORDER=1                                    MOLC2450
      JORDER=IORDER                                                     MOLC2460
      IF (NATYPE.GT.10) GO TO 150                                       MOLC2470
      IF (NATYPE.NE.NTCHEK) GO TO 200                                   MOLC2480
C**** TEST FOR CENTER OF INVERSION                                      MOLC2490
      NAT=NATOMS                                                        MOLC2500
      NINV=0                                                            MOLC2510
      DO 61 IT=1,NATYPE                                                 MOLC2520
      IF (RCCT(IT).GT.CRTZRO) GO TO 65                                  MOLC2530
      NAT=NATOMS-1                                                      MOLC2540
      GO TO 61                                                          MOLC2550
   65 NUMB=NUMT(IT)                                                     MOLC2560
      DO 62 IN=1,NUMB                                                   MOLC2570
      IA=ITYPA(IN,IT)                                                   MOLC2580
      THPI=PI-THETA(IA)                                                 MOLC2590
      PHIPI=DMOD(PHI(IA)+PI,2.D0*PI)                                    MOLC2600
      IF (DSIN(THETA(IA)).LE.1.D-7) PHIPI=0.                            MOLC2610
      DO 63 JN=1,NUMB                                                   MOLC2620
      JA=ITYPA(JN,IT)                                                   MOLC2630
      IF (DABS(THETA(JA)-THPI).GT.CRTZRO.OR.DABS(PHI(JA)-PHIPI).GT.     MOLC2640
     1 CRTZRO) GO TO 63                                                 MOLC2650
      NINV=NINV+1                                                       MOLC2660
      INVAT(IA)=JA                                                      MOLC2670
      GO TO 62                                                          MOLC2680
   63 CONTINUE                                                          MOLC2690
      IODD=0                                                            MOLC2700
      GO TO 160                                                         MOLC2710
   62 CONTINUE                                                          MOLC2720
   61 CONTINUE                                                          MOLC2730
      IF (NINV.EQ.NAT) IODD=1                                           MOLC2740
  160 IF (NAT*NDIV.GT.720) GO TO 150                                    MOLC2750
      GO TO 201                                                         MOLC2760
  200 WRITE (IOUT,202)                                                  MOLC2770
  202 FORMAT (1H1,'THE CALCULATED NUMBER OF ATOM-TYPES DOES NOT AGREE WIMOLC2780
     1TH YOUR ASSUMPTION.  PLEASE RECHECK YOUR INPUT.')                 MOLC2790
      ISTOP=1                                                           MOLC2800
  201 WRITE(IOUT,51) NATYPE                                             MOLC2810
   51 FORMAT(///' THE PROGRAM DISTINGUISHES ',I2,' ATOM TYPES, AS FOLLOWMOLC2820
     1S...'/)                                                           MOLC2830
      WRITE (IOUT,53)                                                   MOLC2840
   53 FORMAT (/18X,'ATOM-TYPE   CHEMICAL   NO. OF   NUCLEAR   UPPER : LOMOLC2850
     1ADING INDICES OF THE ATOMS IN THIS ATOM-TYPE'/20X,'INDEX      SYMBMOLC2860
     2OL     ATOMS     CHARGE    LOWER : RUNNING INDICES WITHIN THE ATOM-TMOLC2870
     3YPE'/18X,9('-'),3X,8('-'),3X,6('-'),3X,7('-'),3X,54('-')/)         MOLC2880
      DO 54 IT=1,NATYPE                                                 MOLC2890
```

```
      NUMB=NUMT(IT)                                                     MOLC2900
      WRITE (IOUT,55) IT,ITSYMB(IT),NUMB,NUCT(IT),(ITYPA(I,IT),I=1,NUMB)MOLC2910
   55 FORMAT (/21X,I2,10X,A2,8X,I2,7X,I2,6X,12I5,(1X/60X,12I5))         MOLC2920
   54 WRITE (IOUT,56) (I,I=1,NUMB)                                      MOLC2930
   56 FORMAT (60X,12I5)                                                 MOLC2940
      IF (IORDER.EQ.1) WRITE (IOUT,100)                                 MOLC2950
  100 FORMAT (32X,'SINCE THIS MOLECULE HAS NO NON-TRIVIAL SYMMETRY ELEMEMOLC2960
     1NTS,'/40X,'FURTHER CALCULATIONS OF SYMMETRY-RELATED'/46X,'PROPERTIMOLC2970
     2ES WILL BE BYPASSED.')                                            MOLC2980
      HINV=HNO                                                          MOLC2990
      IF(IODD.EQ.1) HINV=HYES                                           MOLC3000
      WRITE(IOUT,82) HINV                                               MOLC3010
   82 FORMAT (//' CENTER OF INVERSION - ',A3/)                          MOLC3020
      IF (ISTOP.EQ.0) GO TO 999                                         MOLC3030
  203 READ (INPUT,300) IDENT,JOB                                        MOLC3040
      IF (JOB.EQ.IHOL(6)) STOP                                          MOLC3050
      IF (JOB.NE.IHOL(5)) GO TO 203                                     MOLC3060
      WRITE (IOUT,302)                                                  MOLC3070
      GO TO 1000                                                        MOLC3080
  150 WRITE (IOUT,151)                                                  MOLC3090
  151 FORMAT (//' ONE OR MORE OF THE INPUT PARAMETERS WILL CAUSE ARRAY LMOLC3100
     1IMITS TO BE EXCEEDED. CALCULATION ABANDONED.')                    MOLC3110
      ISTOP=1                                                           MOLC3120
      GO TO 201                                                         MOLC3130
  999 CALL ALPTAB                                                       MOLC3132
      CALL GRID(1,IODD)                                                 MOLC3134
      CALL GRID(2,IODD)                                                 MOLC3136
      RETURN                                                            MOLC3138
      END                                                               MOLC3140
```



```
      SUBROUTINE ALPTAB                                                 ALPT0010
C**** SETS UP TABLE OF COEFFICIENTS FOR CALCULATION OF ASSOCIATED       ALPT0020
C**** LEGENDRE POLYNOMIALS, L .LE. 16, AND TABLE OF FACTORIALS          ALPT0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         ALPT0040
      COMMON/ALP1/THETAC(1224),FACT(50)                                 ALPT0050
      FACT(1)=1.D0                                                      ALPT0060
      DO 3 I=2,50                                                       ALPT0070
      FI=I-1                                                            ALPT0080
    3 FACT(I)=FACT(I-1)*FI                                              ALPT0090
      LMAX=16                                                           ALPT0100
      THETAC(1)=DSQRT(0.5D0)                                            ALPT0110
      DO 4 LI=1,LMAX                                                    ALPT0120
      TLP1=2*LI+1                                                       ALPT0130
      TL=2**LI                                                          ALPT0140
      A=THETAC(1)*DSQRT(TLP1)/TL                                        ALPT0150
      LP1=LI+1                                                          ALPT0160
      ISUB=4*LI*LP1                                                     ALPT0170
      DO 4 MP1=1,LP1                                                    ALPT0180
      MI=MP1-1                                                          ALPT0190
      JSUB=ISUB+8*MI                                                    ALPT0200
      ANORM=A*DSQRT(FACT(LP1-MI)/FACT(LP1+MI))                          ALPT0210
      LIM=(LI-MI)/2+1                                                   ALPT0220
      PHASE=-1.D0                                                       ALPT0230
      IF(2*(MI/2).NE.MI) PHASE=-PHASE                                   ALPT0235
      DO 2 IP=1,LIM                                                     ALPT0240
      NU=IP-1                                                           ALPT0250
      PHASE=-PHASE                                                      ALPT0260
    2 THETAC(JSUB+IP)=ANORM*PHASE*FACT(2*(LI-NU)+1)/(FACT(NU+1)*FACT(LP1ALPT0270
     1 -NU)*FACT(LP1-MI-2*NU))                                          ALPT0280
    4 CONTINUE                                                          ALPT0290
   37 THETAI=0.                                                         ALPT0300
      RETURN                                                            ALPT0310
      END                                                               ALPT0320
```

```
      SUBROUTINE GRID (IGRID,IODD)                                      GRID0010
C**** CALCULATES GRID OF POINTS REFLECTING SYMMETRY OF MOLECULE, WHERE  GRID0020
C**** FUNCTION VALUES WILL BE SAMPLED                                   GRID0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         GRID0040
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE, GRID0050
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30),   GRID0060
     2 INVAT(30)                                                        GRID0070
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 GRID0080
      COMMON/GIRD/RGRID(30,3),CTHGR(1447),PHIGR(1447),NGRID,NDIV        GRID0090
      COMMON/XYZ/XG(1440),YG(1440),ZG(1440),X(30),Y(30),Z(30)           GRID0100
      DIMENSION IAFLAG(30)                                              GRID0110
C>>>> CHANGE NEXT CARD IF DIMENSION OF, E.G., 'RADIUS' IS CHANGED       GRID0120
      MAXNAT=30                                                         GRID0130
      PI =3.141592653589793                                             GRID0140
      PIPI=2.D0*PI                                                      GRID0150
      A1=DTAN (PI/45.D0)                                                GRID0160
      A2=150.D0                                                         GRID0170
      IBASE=0                                                           GRID0180
      NGRID=NDIV*NATOMS                                                 GRID0190
      DO 44 IA=1,NATOMS                                                 GRID0200
      IF (RADIUS(IA).LE.CRTZRO) NGRID=NGRID-NDIV                        GRID0210
   44 CONTINUE                                                          GRID0220
      IF (IGRID.EQ.2) IBASE=NGRID                                       GRID0230
   41 IND=IBASE                                                         GRID0240
      FDIV=NDIV                                                         GRID0250
C**** GRID CONSISTS OF A CIRCLE OF 'NDIV' POINTS ABOUT THE RADIUS VECTORGRID0260
C**** JOINING EACH ATOM TO THE ORIGIN                                   GRID0270
      DO 50 IA=1,NATOMS                                                 GRID0280
   50 IAFLAG(IA)=0                                                      GRID0290
      IACC=0                                                            GRID0300
      DO 1 IA=1,NATOMS                                                  GRID0310
      RIT=RADIUS(IA)                                                    GRID0320
      IF (RIT.GT.CRTZRO) GO TO 55                                       GRID0330
      IACC=1                                                            GRID0340
      GO TO 1                                                           GRID0350
   55 IF (IODD.EQ.0) GO TO 51                                           GRID0360
      IF (IAFLAG(IA).EQ.0) GO TO 45                                     GRID0370
      IND=IND+NDIV                                                      GRID0380
      GO TO 1                                                           GRID0390
   45 IP1=IA+1                                                          GRID0400
      IF (IP1.GT.NATOMS) GO TO 1                                        GRID0410
      JA=INVAT(IA)                                                      GRID0420
   54 JNDIV=(JA-1-IACC)*NDIV+IBASE                                      GRID0430
      IAFLAG(JA)=1                                                      GRID0440
   ** ALGORITHM FOR ANGULAR OPENING SUBTENDED BY GRID CIRCLES           GRID0450
   51 IT=ITYPT(IA)                                                      GRID0460
   20 ZIT=NUCT(IT)                                                      GRID0470
      OPSIDE=A1*(1.D0-ZIT/A2)                                           GRID0480
      IF (IGRID.EQ.2) OPSIDE=1.5D0*OPSIDE                               GRID0490
      HYPOT=DSQRT(1.D0+OPSIDE*OPSIDE)                                   GRID0500
      COSOM=1.D0/HYPOT                                                  GRID0510
      SINOM=OPSIDE/HYPOT                                                GRID0520
      RGRID(IA,IGRID)=0.9D0*RIT/COSOM                                   GRID0530
      RCR=RGRID(IA,IGRID)                                               GRID0540
      IF (IODD.EQ.1) RGRID(JA,IGRID)=RCR                                GRID0550
   ** ASSIGNMENT OF GRID COORDINATES                                   GRID0560
      COSTN=DCOS(THETA(IA))                                             GRID0570
      SINTN=DSIN(THETA(IA))                                             GRID0580
```



```
   34 ALPH=0.                                                           GRID0590
      DO 3 NPT=1,NDIV                                                   GRID0600
      IND=IND+1                                                         GRID0610
      COSAL=DCOS(ALPH)                                                  GRID0620
      CTH=COSTN*COSOM+SINTN*SINOM*COSAL                                 GRID0630
      CTHGR(IND)=CTH                                                    GRID0640
      RADIX=1.D0-CTH*CTH                                                GRID0650
      IF (RADIX.LT.CRTZRO) RADIX=0.                                     GRID0660
      V=DSQRT(RADIX)                                                    GRID0670
      RST=V*RCR                                                         GRID0680
      IF (V*SINTN.LE.CRTZRO) GO TO 4                                    GRID0690
   21 W=(COSOM-COSTN*CTH)/(SINTN*V)                                     GRID0700
      IF (DABS(DABS(W)-1.D0).GT.CRTZRO) GO TO 30                        GRID0710
      IF (W)32,30,33                                                    GRID0720
   32 W=-1.D0                                                           GRID0730
      GO TO 30                                                          GRID0740
   33 W=1.D0                                                            GRID0750
   30 PNMPH=DARCOS(W)                                                   GRID0760
      IF (ALPH-PI)5,6,7                                                 GRID0770
    5 PHIG=PHI(IA)+PNMPH                                                GRID0780
      GO TO 8                                                           GRID0790
    6 PHIG=PHI(IA)                                                      GRID0800
      GO TO 8                                                           GRID0810
    7 PHIG=PHI(IA)-PNMPH                                                GRID0820
      GO TO 8                                                           GRID0830
    4 PHIG=ALPH                                                         GRID0840
    8 PHIGR(IND)=PHIG                                                   GRID0850
      IF (IODD.EQ.0) GO TO 53                                           GRID0860
C**** COORDINATES OF INVERSION-SYMMETRIC POINTS                         GRID0870
      JND=JNDIV+NPT                                                     GRID0880
      CTHGR(JND)=-CTH                                                   GRID0890
      PHIGR(JND)=PHIG+PI                                                GRID0900
      IF (PHIGR(JND).GT.PIPI) PHIGR(JND)=PHIG-PI                        GRID0910
   53 CONTINUE                                                          GRID0920
      XG(IND)=RST*DCOS(PHIG)                                            GRID0930
      YG(IND)=RST*DSIN(PHIG)                                            GRID0940
      ZG(IND)=RCR*CTH                                                   GRID0950
      IF (IODD.EQ.0) GO TO 3                                            GRID0960
      ZG(JND)=-ZG(IND)                                                  GRID0970
      XG(JND)=-XG(IND)                                                  GRID0980
      YG(JND)=-YG(IND)                                                  GRID0990
    3 ALPH=ALPH+2.D0*PI/FDIV                                            GRID1000
    1 CONTINUE                                                          GRID1010
      IF (IGRID.NE.1) RETURN                                            GRID1020
      DO 10 I=1,NATOMS                                                  GRID1030
      X(I)=POS(1,I)                                                     GRID1040
      Y(I)=POS(2,I)                                                     GRID1050
   10 Z(I)=POS(3,I)                                                     GRID1060
      RETURN                                                            GRID1070
      END                                                               GRID1080
```

```
      SUBROUTINE GROUP(ISUM,JSUM)                                       GROU0010
*** DETERMINES APPROPRIATE L-VALUES OF PROTOTYPE FUNCTIONS NEEDED FOR  GROU0020
*** THE MOLECULAR SYMMETRY GROUP AND SUPERVISES CALCULATION OF EVEN AN GROU0030
*** ODD REPRESENTATIONS.                                               GROU0040
      IMPLICIT REAL*8 (A-H,O-Z)                                         GROU0050
      DIMENSION IHOL(18),LDEG(14),IDE(2)                                GROU0060
      DIMENSION ISUMA(10),JSUMA(7),KSUMA(4),L1ARAY(10),L2ARAY(10),L2A(4)GROU0070
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 GROU0080
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    GROU0090
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IFLOWL(14)                        GROU0100
      DATA  IHOL/4H (L ,4H (L=,4H= 1),4H= 2),4H= 3),4H= 4),4H= 5),4H= 6)GROU0110
     1,4H= 7),4H= 8),4H= 9),4H10) ,4H11) ,4H12) ,4H13) ,4H14) ,4H15) ,  GROU0120
     2 4H16) /                                                          GROU0130
      DATA ISUMA/9,13,15,15,17,17,17,23,25,41/,JSUMA/4,8,11,15,19,23,25/GROU0140
     1,KSUMA/9,13,15,17/,L1ARAY/2,4,6,8,10,12,16,6,4,12/,L2ARAY/3,5,9,9,GROU0150
     2  0,0,0,9,0,15/,L2A/1,3,3,5/                                      GROU0160
C**** L=4 IS DONE FIRST.  THE RESULTING BEHAVIOR HELPS TO DETERMINE WHICGROU0190
C**** L-VALUES ARE NEEDED                                               GROU0200
      IF (LVAL1.EQ.0) GO TO 100                                         GROU0210
      L=LVAL1                                                           GROU0220
      ASSIGN 110 TO NN                                                  GROU0230
      GO TO 40                                                          GROU0240
  110 IF (IODD.EQ.0) GO TO 50                                           GROU0250
      L=LVAL2                                                           GROU0260
      GO TO 30                                                          GROU0270
  100 ISUM=0                                                            GROU0280
      IDE(1)=IHOL(1)                                                    GROU0290
      IDE(2)=IHOL(6)                                                    GROU0300
      LVAL1=4                                                           GROU0310
      CALL PROSYM (4,IDE,NR,ISUM,LDEG,1)                                GROU0320
      IF (ISTOP.EQ.1) RETURN                                            GROU0330
C**** BRANCH TO APPROPRIATE L-VALUE CALCULATIONS                        GROU0340
C**** TEST VALUE OF ISUM = SUM OF SQUARES OF DEGENERACIES OF EIGENVALUESGROU0350
C**** FROM L=4 MATRIX                                                   GROU0360
      IFLAG=4                                                           GROU0370
      IF (ISUM.LE.17) GO TO 200                                         GROU0380
C**** IF ISUM.GT.17, LOOK UP L VALUES IN TABLE, FIND PROTOTYPE FUNCTIONSGROU0390
      DO 210 I=8,10                                                     GROU0400
      IF (ISUM.NE.ISUMA(I)) GO TO 210                                   GROU0410
      L1=L1ARAY(I)                                                      GROU0420
      IF (IODD.NE.0) L2=L2ARAY(I)                                       GROU0430
      GO TO 115                                                         GROU0440
  210 CONTINUE                                                          GROU0450
  115 L=L1                                                              GROU0460
      IF (L.EQ.4) IFLAG=2                                               GROU0470
      ASSIGN 120 TO NN                                                  GROU0480
      GO TO 40                                                          GROU0490
  120 IF(IODD.EQ.0) GO TO 50                                            GROU0500
      L=L2                                                              GROU0510
      GO TO 30                                                          GROU0520
C**** FOR ISUM.LE.17,TRY ANOTHER L VALUE, FIND JSUM=SUM OF SQUARES OF   GROU0530
C**** DISTINCT REPS. FOUND                                              GROU0540
  200 L=(ISUM-5)/2                                                      GROU0550
      IF (ISUM/4.EQ.3)L=4                                               GROU0560
      IF (ISUM.EQ.15)IFLAG=2                                            GROU0570
      ASSIGN 140 TO NN                                                  GROU0580
      IF (ISUM.EQ.13) GO TO 60                                          GROU0590
      GO TO 40                                                          GROU0600
```



```
C**** COMPARE ISUM,JSUM WITH TABLES, LOOK UP L VALUES                   GROU0610
  140 DO 150 I=1,7                                                      GROU0620
      IF (ISUM.EQ.ISUMA(I).AND.JSUM.EQ.JSUMA(I)) GO TO 155              GROU0630
      GO TO 150                                                         GROU0640
  155 L1=L1ARAY(I)                                                      GROU0650
      IF (IODD.NE.0) L2=L2ARAY(I)                                       GROU0660
      GO TO 160                                                         GROU0670
  150 CONTINUE                                                          GROU0680
      DO 170 I=1,4                                                      GROU0690
      IF (ISUM.NE.KSUMA(I)) GO TO 170                                   GROU0700
      L1=L                                                              GROU0710
      L2=L2A(I)                                                         GROU0720
      GO TO 160                                                         GROU0730
  170 CONTINUE                                                          GROU0740
  160 IF(L1.EQ.L) GO TO 161                                             GROU0750
      L=L1                                                              GROU0760
      ASSIGN 161 TO NN                                                  GROU0770
      IFLAG=4                                                           GROU0780
      GO TO 40                                                          GROU0790
  161 IF(IODD.EQ.0) GO TO 50                                            GROU0800
      L=L2                                                              GROU0810
      GO TO 30                                                          GROU0820
C**** GENERATION OF SYMMETRY-ADAPTED PROTOTYPE VECTORS                  GROU0830
   50 NRO=0                                                             GROU0840
      LVAL2=0                                                           GROU0850
      GO TO 5                                                           GROU0860
   40 IDE(1)=IHOL(1)                                                    GROU0870
      IF (L.GT.9) IDE(1)=IHOL(2)                                        GROU0880
      IDE(2)=IHOL(L+2)                                                  GROU0890
      LVAL1=L                                                           GROU0900
      CALL PROSYM (L,IDE,NR,ISUM,LDEG,IFLAG)                            GROU0910
      IF (ISTOP.EQ.1) RETURN                                            GROU0920
   60 JSUM=0                                                            GROU0930
      DO 41 I=1,NR                                                      GROU0940
   41 JSUM=JSUM+LDEG(I)**2                                              GROU0950
      GO TO NN, (120,140,161,110)                                       GROU0960
   30 IDE(1)=IHOL(1)                                                    GROU0970
      IF (L.GT.9) IDE(1)=IHOL(2)                                        GROU0980
      IDE(2)=IHOL(L+2)                                                  GROU0990
      LVAL2=L                                                           GROU1000
      CALL PROSYM (L,IDE,NRO,ISUM,LDEG,4)                               GROU1010
      IF (ISTOP.EQ.1) RETURN                                            GROU1020
C**** SUMMARY OF SYMMETRY PROPERTIES                                    GROU1030
    5 NREP=NR+NRO                                                       GROU1040
      CALL PHASER                                                       GROU1050
      RETURN                                                            GROU1070
      END                                                               GROU1080
```

```
      SUBROUTINE PROSYM (L,IDE,NR,ISUM,LDEGT,IFLAG)                     PROS0010
C**** SUPERVISES DETERMINATION OF IRREDUCIBLE REPRESENTATIONS SPANNED   PROS0020
C**** BY A PARTICULAR SET OF SPHERICAL HARMONICS                        PROS0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         PROS0040
      REAL*8 LABEL,LABEL2                                               PROS0050
      COMMON/GIRD/RGRID(30,3),CTHGR(1447),PHIGR(1447),NGRID,NDIV        PROS0060
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 PROS0070
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    PROS0080
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                        PROS0090
      COMMON/PROTCO/COPRO(1802),EVP(56),IREPL(25),NEVALP(2)             PROS0100
      COMMON/PHASE/IFLGG(20),IPHASE                                     PROS0110
      COMMON/ASYMI/LA,MGRID,L3P1,LAP1,LADEG                             PROS0120
      COMMON/FILES/INPUT,IOUT                                           PROS0130
      COMMON/WORK1/DUMMY(8000)                                          PROS0140
      DIMENSION LABEL(50),INDEX(50),IDENT1(20),LDEGT(14),IDE(2),        PROS0150
     1 NDEGT(33),IREPLT(33),RES(50,50),EV(50),C(50,50),B(50,50),NDEG(50)PROS0150
      DIMENSION IDENT2(20),LABEL2(50),INDEX2(50)                        PROS0170
      EQUIVALENCE (RES,DUMMY),(EV,DUMMY(2501)),(C,DUMMY(2551)),         PROS0180
     1 (B,DUMMY(5051)),(NDEG,DUMMY(7975)),(NEVAL,DUMMY(8000)),          PROS0190
     2 (NDEGT,DUMMY(7551)),(IREPLT,DUMMY(7950)),(LABEL,DUMMY(7875)),    PROS0200
     3(INDEX,DUMMY(7925)),(LABEL2, DUMMY(7576)),(INDEX2,DUMMY(7626))    PROS0210
      DATA IDENT1/4H MAT,4HRIX ,4HOF P,4H OVE,4HR SP,4HHERI,4HCAL ,     PROS0220
     14HHARM,4HONIC,4HS    ,10*4H    /                                  PROS0230
      DATA IDENT2/4H MAT,4HRIX ,4HOF P,4H' OV,4HER E,4HIGEN,4HVECT,4HORSPROS0240
     1 ,4HOF F,4HIRST,4H MAT,4HRIX ,8*4H    /                           PROS0250
      DATA HOL3/7H   COS /,HOL4/7H   SIN /,HOL5/7H  E.V. /              PROS0260
      EXTERNAL THLMK,ANTSYM                                             PROS0270
      NMAX=50                                                           PROS0280
      IPHASE=3                                                          PROS0290
      IF (IFLAG.EQ.2) GO TO 1                                           PROS0300
      IDENT1(11) = IDE(1)                                               PROS0310
      IDENT1(12)=IDE(2)                                                 PROS0320
      INDEX(1)=0                                                        PROS0330
      LABEL(1)=HOL3                                                     PROS0340
      DO 60 M=1,L                                                       PROS0350
      M2=2*M                                                            PROS0360
      INDEX(M2)=M                                                       PROS0370
      INDEX(M2+1)=M                                                     PROS0380
      LABEL(M2)=HOL4                                                    PROS0390
   60 LABEL(M2+1)=HOL3                                                  PROS0400
      LL=2*L+1                                                          PROS0410
      KALC=1                                                            PROS0420
      IF (IODD.NE.0) KALC=9                                             PROS0430
**** GENERATE CRUDE SYMMETRY VECTORS                                    PROS0440
      CALL ADAPT (L,LL,THLMK,KALC,1,IDENT1,1)                           PROS0450
      IF (L.NE.4) GO TO 1                                               PROS0460
      ISUM=0                                                            PROS0470
      DO 2 I=1,NEVAL                                                    PROS0480
    2 ISUM=ISUM+NDEG(I)**2                                              PROS0490
      IF (ISUM.NE.13) RETURN                                            PROS0500
    1 IF (INTER.EQ.0) GO TO 3                                           PROS0510
      WRITE (IOUT,200)                                                  PROS0520
  200 FORMAT (//' DEGENERACIES OF EIGENVALUES FOR PROTOTYPE FUNCTIONS'/)PROS0530
      WRITE (IOUT,100) (NDEG(I),I=1,NEVAL)                              PROS0540
  100 FORMAT (16I5)                                                     PROS0550
    3 IF (ISTOP.EQ.1) RETURN                                            PROS0560
      IF (2*(L/2).NE.L) GO TO 20                                        PROS0570
**** CALCULATE SECOND MATRIX OF SYMMETRY OPERATOR                       PROS0580
```



```
      CALL PSYM (L,LL,THLMK,L,LL,THLMK,2,KALC,RES,B,NMAX)               PROS0590
      L1=L                                                              PROS0600
      LL1=LL                                                            PROS0610
C**** TRANSFORM MATRIX WITH VECTORS - RESULTING MATRIX WILL BE BLOCK     PROS0520
C**** DIAGONAL EXCEPT FOR BLOCKS CONNECTING VECTORS BELONGING TO THE SAMPROS0630
C**** REPRESENTATION                                                    PROS0640
      CALL LINCOM (RES,EV,NDEG,NMAX,LL,C,B,NEVAL,IREPLT,NR,LDEGT)       PROS0650
      IDENT2(13)=IDE(1)                                                 PROS0660
      IDENT2(14)=IDE(2)                                                 PROS0670
      DO 30 I=1,LL                                                      PROS0680
      LABEL2(I)=HOL5                                                    PROS0690
   30 INDEX2(I)=I                                                       PROS0700
      IF(INTER.EQ.1)CALL MATPRT (LL,0,IDENT2,RES,EV,NMAX,LABEL2,INDEX2) PROS0710
      DO 6 I=1,LL                                                       PROS0720
    6 EV(I)=RES(I,I)                                                    PROS0730
C**** CHECK TO MAKE SURE DEGENERACIES HAVEN'T BEEN DEGRADED             PROS0740
      CALL DEGEN (LL,EV,NDEGT,NEVALT,CRTDEG)                            PROS0750
      IF (NEVAL.NE.NEVALT) GO TO 7                                      PROS0760
      CALL SORT1 (C,B,EV,NMAX,LL,NDEG,-NEVAL,IREPLT)                    PROS0770
      NRT=NR                                                            PROS0780
      NREP=NR                                                           PROS0790
      IND=0                                                             PROS0800
      IBASE=0                                                           PROS0810
      IS1=0                                                             PROS0820
      ISUB=0                                                            PROS0830
      NEVALP(1)=NEVAL                                                   PROS0840
      GO TO 13                                                          PROS0850
C**** IF L IS ODD, LINE UP PHASES OF ODD REPRESENTATIONS WITH EVEN ONES PROS0860
   20 LA=MAX0(2,L/2+1)                                                  PROS0870
      CALL WIG3J (LA,LA,L,ISTOP)                                        PROS0880
      MGRID=NGRID                                                       PROS0890
      LAP1=LA+1                                                         PROS0900
      LADEG=2*LA+1                                                      PROS0910
      L3P1=L+1                                                          PROS0920
      CALL ADAPT (L,LL,ANTSYM,9,1,IDENT1,2)                             PROS0930
      DO 41 I=1,NEVAL                                                   PROS0940
   41 IREPLT(I)=IREPLT(I)+NRT                                           PROS0950
      IND=NRT                                                           PROS0960
      NR=NRT                                                            PROS0970
      NREP=2*NR                                                         PROS0980
      NEVALP(2)=NEVAL                                                   PROS0990
      IBASE=NEVALP(1)                                                   PROS1000
      IS1=LL1                                                           PROS1010
      ISUB=IS1*IS1                                                      PROS1020
C**** STORE VECTORS FOR LATER USE                                       PROS1030
   13 DO 15 I=1,NEVAL                                                   PROS1040
   15 IREPL(I+IBASE)=IREPLT(I)                                          PROS1050
      DO 40 I=1,NR                                                      PROS1060
   40 NDEGA(I+IND)=LDEGT(I)                                             PROS1070
      DO 11 J=1,LL                                                      PROS1080
      EVP(J+IS1)=EV(J)                                                  PROS1090
      DO 11 I=1,LL                                                      PROS1100
      ISUB=ISUB+1                                                       PROS1110
   11 COPRO(ISUB)=C(I,J)                                                PROS1120
      RETURN                                                            PROS1130
    7 ISTOP=1                                                           PROS1140
      WRITE (IOUT,50)                                                   PROS1150
   50 FORMAT (1H1,'DEGENERACIES OF MATRIX 2 DO NOT AGREE WITH THOSE IN MPROS1160
```

```
IX 1.  CALCULATION ABANDONED.')                                  PROS1170
URN                                                              PROS1180
                                                                 PROS1190
```



```
      SUBROUTINE ADAPT (L,N,F2,KALC,MPAR,IDENT,IFPRO)                    ADAP0010
C**** SYMMETRY-ADAPTS SET OF FUNCTIONS DEFINED BY FUNCTION F2            ADAP0020
      IMPLICIT REAL*8 (A-H,O-Z)                                          ADAP0030
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                  ADAP0040
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,     ADAP0050
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                         ADAP0060
      COMMON/FILES/INPUT,IOUT                                            ADAP0070
      COMMON/WORK1/DUMMY(8000)                                           ADAP0080
      DIMENSION LABEL(50),INDEX(50),IREPLB(50),IDENT(10),RES(50,50),     ADAP0090
     1 EV(50),C(50,50),B(50,50),NDEG(50)                                 ADAP0100
      EQUIVALENCE (RES,DUMMY),(EV,DUMMY(2501)),(C,DUMMY(2551)),          ADAP0110
     1 (B,DUMMY(5051)),(LABEL,DUMMY(7875)),(INDEX,DUMMY(7925)),          ADAP0120
     2 (IREPLB,DUMMY(7950)),(NDEG,DUMMY(7975)),(NEVAL,DUMMY(8000))       ADAP0130
      EXTERNAL F2,THLMK                                                  ADAP0140
      NMAX=50                                                            ADAP0150
      IF (IFPRO.EQ.2) GO TO 10                                           ADAP0160
C**** CALCULATE CONNECTION MATRIX OVER BASIS FUNCTIONS                   ADAP0170
      CALL PSYM (L,N,F2,L,N,F2,1,KALC,RES,B,NMAX)                        ADAP0180
      IF (INTER.EQ.1) CALL MATPRT (N,0,IDENT,RES,EV,NMAX,LABEL,INDEX)    ADAP0190
C**** DIAGONALIZE TO GET CRUDE SYMMETRY VECTORS                          ADAP0200
      CALL EIGEN (RES,B,N,EV,N,SRNORM,NMAX)                              ADAP0210
      DO 1 I=1,N                                                         ADAP0220
      DO 1 J=1,N                                                         ADAP0230
      IF (DABS(RES(I,J)).LE.CRTZRO) RES(I,J)=0.                          ADAP0240
    1 CONTINUE                                                           ADAP0250
C**** FIND DEGENERACIES                                                  ADAP0260
      CALL DEGEN (N,EV,NDEG,NEVAL,CRTDEG)                                ADAP0270
C**** SORT VECTORS BY DEGENERACY                                         ADAP0280
      CALL SORT1 (RES,C,EV,NMAX,N,NDEG,NEVAL,NDEG)                       ADAP0290
      IF (INTER.EQ.1) CALL MATPRT (N,1,IDENT,C,EV,NMAX,LABEL,INDEX)      ADAP0300
      IF (IFPRO.EQ.1) RETURN                                             ADAP0310
   10 DO 2 I=1,NEVAL                                                     ADAP0320
    2 IREPLB(I)=0                                                        ADAP0330
C**** LINE UP PHASES WITH PROTOTYPE FUNCTIONS                            ADAP0340
      DO 3 IPAR=1,2                                                      ADAP0350
      IF (MPAR.EQ.0) GO TO 4                                             ADAP0360
      IF (MPAR.EQ.1.AND.IPAR.EQ.2) GO TO 3                               ADAP0370
      IF (MPAR.EQ.-1.AND.IPAR.EQ.1) GO TO 3                              ADAP0380
    4 LV=LVAL1                                                           ADAP0390
      IF (IPAR.EQ.2) LV=LVAL2                                            ADAP0400
      IF (LV.EQ.0) GO TO 3                                               ADAP0410
      LVD=2*LV+1                                                         ADAP0420
      CALL PSYM (LV,LVD,THLMK,L,N,F2,2,-KALC,RES,B,NMAX)                 ADAP0430
      CALL LINEUP (N,IPAR)                                               ADAP0440
    3 CONTINUE                                                           ADAP0450
      DO 5 I=1,NEVAL                                                     ADAP0460
      IF (IREPLB(I).EQ.0) GO TO 6                                        ADAP0470
    5 CONTINUE                                                           ADAP0480
C**** SORT VECTORS BY REPRESENTATION                                     ADAP0490
      CALL SORT1 (C,B,EV,NMAX,N,NDEG,-NEVAL,IREPLB)                      ADAP0500
      IF (IFPRO.EQ.2) RETURN                                             ADAP0510
C**** UNDO ARBITRARY MIXING OF SETS BELONGING TO SAME REP.               ADAP0520
      CALL CLEAN (C,N,NDEG,NEVAL,IREPLB,NMAX,CRTZRO,0)                   ADAP0530
      RETURN                                                             ADAP0540
    6 WRITE (IOUT,7)  (IREPLB(I),I=1,NEVAL)                              ADAP0550
    7 FORMAT (//' ONE OR MORE VECTORS NOT SYMMETRY-ADAPTED.  CALCULATIONADAP0560
     1 ABANDONED.'//2015)                                                ADAP0570
      ISTOP=1                                                            ADAP0580
```

```
RETURN                                                    ADAP0590
END                                                       ADAP0600
```



```
      SUBROUTINE PSYM (L1,N1,F1,L2,N2,F2,IGRID,KALC,RES,R,NMAX)          PSYM0010
C**** CALCULATES MATRIX OF SYMMETRY OPERATOR OVER FUNCTIONS SPECIFIED    PSYM0020
C**** BY F1, F2                                                          PSYM0030
      IMPLICIT REAL*8 (A-H,O-Z)                                          PSYM0040
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE,  PSYM0050
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30),    PSYM0060
     2 INVAT(30)                                                         PSYM0070
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                  PSYM0080
      COMMON/GIRD/RGRID(30,3),CTHGR(1447),PHIGR(1447),NGRID,NDIV         PSYM0090
      COMMON/ALP/RCR,SQ2OR,W,Y,LC                                        PSYM0100
      DIMENSION FUNCT1(100),FUNCT2(100)                                  PSYM0110
      DIMENSION IAFLAG(30)                                               PSYM0120
      DIMENSION RES(NMAX,N1),R(NMAX)                                     PSYM0130
      EXTERNAL F1,F2                                                     PSYM0140
      ROOTHF=DSQRT(0.500)                                                PSYM0150
      IBASE=0                                                            PSYM0160
      IF (IGRID.EQ.2) IBASE=NGRID                                        PSYM0170
   41 ISUB=N1*N2                                                         PSYM0180
      DO 31 I=1,ISUB                                                     PSYM0190
   31 R(I)=0.                                                            PSYM0200
      IND=IBASE                                                          PSYM0210
      DO 4 I=1,NATOMS                                                    PSYM0220
    4 IAFLAG(I)=0                                                        PSYM0230
C**** LOOP OVER GRID POINTS                                              PSYM0240
      DO 1 IA=1,NATOMS                                                   PSYM0250
      IF (RADIUS(IA).LE.CRTZRO) GO TO 1                                  PSYM0260
      IF (IABS(KALC).NE.9) GO TO 3                                       PSYM0270
C**** IF /KALC/=9, TAKE ONLY ONE ATOM FROM EACH INVERSION-SYMMETRIC      PSYM0280
C**** PAIR.  CONTRIBUTION TO MATRIX ELEMENT FROM OTHER ATOM IS IDENTICAL PSYM0290
      IF (IAFLAG(IA).EQ.0) GO TO 5                                       PSYM0300
      IND=IND+NDIV                                                       PSYM0310
      GO TO 1                                                            PSYM0320
    5 JA=INVAT(IA)                                                       PSYM0330
      IAFLAG(JA)=1                                                       PSYM0340
    3 RCR=1.D0/RGRID(IA,IGRID)                                           PSYM0350
      SQ2OR=ROOTHF*RCR                                                   PSYM0360
      DO 2 NPT=1,NDIV                                                    PSYM0370
      IND=IND+1                                                          PSYM0380
      CALL  F1 (L1,N1,FUNCT1,IND)                                        PSYM0390
      IF(KALC.LT.0) GO TO 39                                             PSYM0400
      DO 25 I=1,N1                                                       PSYM0410
   25 FUNCT2(I)=FUNCT1(I)                                                PSYM0420
      GO TO 24                                                           PSYM0430
   39 CALL   F2 (L2,N2,FUNCT2,IND)                                       PSYM0440
   24 ISUB=0                                                             PSYM0450
      DO 10 I=1,N1                                                       PSYM0460
      LIM=I                                                              PSYM0470
      IF (KALC.LT.0) LIM=N2                                              PSYM0480
      DO 10 J=1,LIM                                                      PSYM0490
      ISUB=ISUB+1                                                        PSYM0500
   10 R(ISUB)=R(ISUB)+FUNCT1(I)*FUNCT2(J)                                PSYM0510
    2 CONTINUE                                                           PSYM0520
    1 CONTINUE                                                           PSYM0530
C**** CLEAN UP MATRIX                                                    PSYM0540
      ISUB=0                                                             PSYM0550
      DO 20 I=1,N1                                                       PSYM0560
      LIM=I                                                              PSYM0570
      IF (KALC.LT.0) LIM=N2                                              PSYM0580
```

```
      DO 20 J=1,LIM                                                     PSYM0590
      ISUB=ISUB+1                                                       PSYM0600
      IF (DABS(R(ISUB)).LE.CRTZRO) R(ISUB)=0.                           PSYM0610
      RES(I,J)=R(ISUB)                                                  PSYM0620
      IF (KALC.GT.0) RES(J,I)=R(ISUB)                                   PSYM0630
   20 CONTINUE                                                          PSYM0640
      RETURN                                                            PSYM0550
      END                                                               PSYM0660
```



```
      SUBROUTINE THLMK (L,LL,FUNCT,IND)                                 THLM0010
C**** CALCULATES FUNCTION VALUES OF REAL SPHERICAL HARMONICS CENTERED   THLM0020
C**** AT ORIGIN, EVALUATED AT GRID POINT 'IND'                          THLM0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         THLM0040
      COMMON/GIRD/RGRID(30,3),CTHGR(1447),PHIGR(1447),NGRID,NDIV        THLM0050
      COMMON/ALP/RCR,SQ2OR,W,Y,LC                                       THLM0060
      DIMENSION FUNCT (40)                                              THLM0070
      DIMENSION COSMP(30),SINMP(30)                                     THLM0080
      LC=L                                                              THLM0090
      W=CTHGR(IND)                                                      THLM0100
      RADIX=1.D0-W*W                                                    THLM0110
      IF (RADIX.LT.1.D-8) RADIX=0.                                      THLM0120
      Y=DSQRT(RADIX)                                                    THLM0130
      PHIX=PHIGR(IND)                                                   THLM0140
      FUNCT(1)=THETAX(0)*SQ2OR                                          THLM0150
      IF (L.EQ.0) RETURN                                                THLM0160
      SINMP(1)=DSIN(PHIX)                                               THLM0170
      COSMP(1)=DCOS(PHIX)                                               THLM0180
      DO 1 M=1,L                                                        THLM0190
      IF (M.EQ.1) GO TO 10                                              THLM0200
      SINMP(M)=SINMP(M-1)*COSMP(1)+COSMP(M-1)*SINMP(1)                  THLM0210
      COSMP(M)=COSMP(M-1)*COSMP(1)-SINMP(M-1)*SINMP(1)                  THLM0220
   10 M2=M*M                                                            THLM0230
      FACTOR=THETAX(M)*RCR                                              THLM0240
      IF(2*(M/2).NE.M) FACTOR=-FACTOR                                   THLM0245
      FUNCT(M2+1)=FACTOR*COSMP(M)                                       THLM0250
    1 FUNCT(M2)=FACTOR*SINMP(M)                                         THLM0260
      RETURN                                                            THLM0270
      END                                                               THLM0280
```

```
      FUNCTION THETAX (M)
C**** CALCULATION OF VALUE OF THETA(L,M) (X)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ALP1/THETAC(1224),FACT(50)
      COMMON/ALP/RCR,SQ2OR,X,Y,L
      ISUB=8*M+4*L*(L+1)
      MAXIP=(L-M)/2+1
      LM2=L-M+2
      THETAX=0.
      IF (DABS(X).LE.1.D-7) GO TO 60
      IF (DABS(Y).LE.1.D-7) GO TO 50
      DO 34 IP=1,MAXIP
   34 THETAX=THETAX+THETAC(ISUB+IP)*X**(LM2-2*IP)
      THETAX=THETAX*Y**M
      RETURN
   60 IF (2*((L-M)/2).EQ.L-M) THETAX=THETAC(ISUB+MAXIP)
      RETURN
   50 IF (M.EQ.0) THETAX=THETAC(ISUB+1)*X**L
      RETURN
      END
```



```
THET0010  SUBROUTINE MATPRT (N,IEV,IDENT,A,B,NMAX,LABEL,INDEX)          MATP0010
THET0020  MATRIX OUTPUT ROUTINE                                         MATP0020
THET0030  IMPLICIT REAL*8 (A-H,O-Z)                                     MATP0030
THET0040  REAL*8 LABEL                                                  MATP0040
THET0050  COMMON/FILES/INPUT,IOUT                                       MATP0050
THET0060  DIMENSION IDENT(20),A(NMAX,N),B(N)                            MATP0060
THET0070  DIMENSION LABEL(NMAX),INDEX(NMAX)                             MATP0070
THET0080  NFIN=0                                                        MATP0080
THET0090  NINIT=NFIN+1                                                  MATP0090
THET0100  NFIN=NFIN+6                                                   MATP0100
THET0110  IF (NFIN.GT.N) NFIN=N                                         MATP0110
THET0120  WRITE (IOUT,10) IDENT,NINIT,NFIN                              MATP0120
THET0130  FORMAT (1H1,20A4//' COLUMNS ',I2,' TO ',I2)                   MATP0130
THET0140  IF (IEV.EQ.0) GO TO 8                                         MATP0140
THET0150  WRITE (IOUT,5)                                                MATP0150
THET0160  FORMAT (/' EIGENVALUES')                                      MATP0160
THET0170  WRITE (IOUT,6) (B(I),I=NINIT,NFIN)                            MATP0170
THET0180  FORMAT (/12X,1P6E20.10)                                       MATP0180
THET0190  WRITE (IOUT,7)                                                MATP0190
THET0200  FORMAT (/' EIGENVECTORS')                                     MATP0200
          DO 11 I=1,N                                                   MATP0210
          WRITE (IOUT,2) LABEL(I),INDEX(I),(A(I,J),J=NINIT,NFIN)        MATP0220
          FORMAT (/1X,A7,1H(,I2,1H),1P6E20.10)                          MATP0230
          IF (NFIN.LT.N) GO TO 1                                        MATP0240
          RETURN                                                        MATP0250
          END                                                           MATP0260
```

```
      SUBROUTINE EIGEN (A,S,N,EV,NN,ONORM2,NMAX)                        EIGE0010
C**** JACOBI METHOD - DIAGONALIZES REAL, SYMMETRIC MATRIX 'A', STORES   EIGE0020
C**** EIGENVALUES IN 'EV', EIGENVECTORS IN 'S'.  'EV' AND 'S' ARE SORTEDEIGE0030
C**** IN ORDER OF DECREASING EIGENVALUE.  EIGENVECTORS ARE COLUMNS OF 'SEIGE0040
      IMPLICIT REAL*8 (A-H,O-Z)                                         EIGE0050
      DIMENSION A(NMAX,N),S(NMAX,N),EV(NMAX)                            EIGE0060
      ROOTHF=DSQRT(0.5D0)                                               EIGE0070
      RHO=1.0D-13                                                       EIGE0080
      DO 30 I=1,N                                                       EIGE0090
      S(I,I)=1.D0                                                       EIGE0100
      IF (I.EQ.N) GO TO 31                                              EIGE0110
      K=I+1                                                             EIGE0120
      DO 30 J=K,N                                                       EIGE0130
      S(I,J)=0.                                                         EIGE0140
   30 S(J,I)=0.                                                         EIGE0150
   31 TINT1=0.                                                          EIGE0160
      DO 60 I=2,N                                                       EIGE0170
      K=I-1                                                             EIGE0180
      DO 60 J=1,K                                                       EIGE0190
   60 TINT1=TINT1+A(I,J)**2                                             EIGE0200
      THR=DSQRT(2.D0*TINT1)                                             EIGE0210
      B=N                                                               EIGE0220
      ONORM2=THR*RHO/B                                                  EIGE0230
      IND=0                                                             EIGE0240
   90 THR=THR/B                                                         EIGE0250
      IF (N.EQ.1) GO TO 310                                             EIGE0260
C**** LOOP THROUGH OFF-DIAGONAL ELEMENTS                                EIGE0270
   95 DO 240 K=2,N                                                      EIGE0280
      KK=K-1                                                            EIGE0290
      DO 240 J=1,KK                                                     EIGE0300
      IF (DABS(A(J,K)).LE.THR) GO TO 240                                EIGE0310
      IND=1                                                             EIGE0320
      V1=A(J,J)                                                         EIGE0330
      V2=A(J,K)                                                         EIGE0340
      V3=A(K,K)                                                         EIGE0350
      UM=0.5D0*(V1-V3)                                                  EIGE0360
      IF (UM.NE.0.) GO TO 155                                           EIGE0370
      OMEGA=-1.D0                                                       EIGE0380
      GO TO 160                                                         EIGE0390
  155 OMEGA=-V2/DSQRT(V2*V2+UM*UM)*DSIGN(1.D0,UM)                       EIGE0400
  160 RADIX=1.D0-OMEGA*OMEGA                                            EIGE0410
      IF (RADIX.LT.1.D-10) RADIX=0.                                     EIGE0420
      SINT=ROOTHF*OMEGA/DSQRT(1.D0+DSQRT(RADIX))                        EIGE0430
      RADIX=1.D0-SINT*SINT                                              EIGE0440
      IF (RADIX.LT.1.D-10) RADIX=0.                                     EIGE0450
      COST=DSQRT(RADIX)                                                 EIGE0460
      L=J-1                                                             EIGE0470
      DO 185 I=1,L                                                      EIGE0480
      TINT1=A(I,K)                                                      EIGE0490
      A(I,K)=A(I,J)*SINT+TINT1*COST                                     EIGE0500
  185 A(I,J)=A(I,J)*COST-TINT1*SINT                                     EIGE0510
      L=J+1                                                             EIGE0520
      M=K-1                                                             EIGE0530
      IF (L.GT.M) GO TO 192                                             EIGE0540
      DO 191 I=L,M                                                      EIGE0550
      TINT1=A(I,K)                                                      EIGE0560
      A(I,K)=A(J,I)*SINT+TINT1*COST                                     EIGE0570
  191 A(J,I)=A(J,I)*COST-TINT1*SINT                                     EIGE0580
```



```
  192 M=K+1                                                             EIGE0590
      IF (M.GT.N) GO TO 199                                             EIGE0600
      DO 198 I=M,N                                                      EIGE0610
      TINT1=A(K,I)                                                      EIGE0620
      A(K,I)=A(J,I)*SINT+TINT1*COST                                     EIGE0630
  198 A(J,I)=A(J,I)*COST-TINT1*SINT                                     EIGE0640
  199 DO 215 I=1,N                                                      EIGE0650
      TINT1=S(I,K)                                                      EIGE0660
      S(I,K)=S(I,J)*SINT+S(I,K)*COST                                    EIGE0670
  215 S(I,J)=S(I,J)*COST-TINT1*SINT                                     EIGE0680
      COST2=COST*COST                                                   EIGE0690
      SINT2=SINT*SINT                                                   EIGE0700
      V4=2.D0*V2*SINT*COST                                              EIGE0710
      A(J,J)=V1*COST2+V3*SINT2-V4                                       EIGE0720
      A(K,K)=V1*SINT2+V3*COST2+V4                                       EIGE0730
      A(J,K)=0.                                                         EIGE0740
  240 CONTINUE                                                          EIGE0750
C**** THRESHOLD TEST                                                    EIGE0760
      IF (IND.NE.1) GO TO 260                                           EIGE0770
      IND=0                                                             EIGE0780
      GO TO 95                                                          EIGE0790
  260 IF (THR.GT.ONORM2) GO TO 90                                       EIGE0800
C**** SORTING OF EIGENVALUES AND EIGENVECTORS                           EIGE0810
      DO 285 K=1,N                                                      EIGE0820
      DO 285 J=K,N                                                      EIGE0830
      IF (A(K,K).GE.A(J,J)) GO TO 285                                   EIGE0840
      TEMP=A(K,K)                                                       EIGE0850
      A(K,K)=A(J,J)                                                     EIGE0860
      A(J,J)=TEMP                                                       EIGE0870
      DO 280 L=1,N                                                      EIGE0880
      TEMP=S(L,K)                                                       EIGE0890
      S(L,K)=S(L,J)                                                     EIGE0900
  280 S(L,J)=TEMP                                                       EIGE0910
  285 CONTINUE                                                          EIGE0920
  310 DO 300 I=1,N                                                      EIGE0930
      EV(I)=A(I,I)                                                      EIGE0940
      DO 300 J=1,N                                                      EIGE0950
  300 A(I,J)=S(I,J)                                                     EIGE0960
      RETURN                                                            EIGE0970
      END                                                               EIGE0980
```

```
      SUBROUTINE DEGEN (NMAT,EV,NDEG,NEVAL,CRTDEG)                      DEGE0010
C**** DETERMINES NUMBER OF DISTINCT EIGENVALUES AND THEIR DEGENERACIES  DEGE0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         DEGE0030
      DIMENSION EV(NMAT),NDEG(NMAT)                                     DEGE0040
      IEV=1                                                             DEGE0050
      NEVAL=0                                                           DEGE0060
   10 LDEG=1                                                            DEGE0070
      IP1=IEV+1                                                         DEGE0080
      NEVAL=NEVAL+1                                                     DEGE0090
      IF (IP1.GT.NMAT) GO TO 30                                         DEGE0100
      DO 3 J=IP1,NMAT                                                   DEGE0110
      IF (EV(IEV)/EV(J).GT.CRTDEG.OR.EV(IEV)/EV(J).LT.1.D0/CRTDEG)GOTO 4DEGE0120
    3 LDEG=LDEG+1                                                       DEGE0130
    4 NDEG(NEVAL)=LDEG                                                  DEGE0140
      IEV=IEV+LDEG                                                      DEGE0150
      GO TO 10                                                          DEGE0160
   30 NDEG(NEVAL)=LDEG                                                  DEGE0170
      NSUM=0                                                            DEGE0180
      DO 5 I=1,NEVAL                                                    DEGE0190
    5 NSUM=NSUM+NDEG(I)                                                 DEGE0200
      IF (NSUM.GT.NMAT) NEVAL=NEVAL-1                                   DEGE0210
      RETURN                                                            DEGE0220
      END                                                               DEGE0230
```



```
      SUBROUTINE SORT1 (RES,C,EV,NMAX,NBASIS,NDEG,NEVLL,IARRAY)         SORT0010
C**** SORTS VECTORS IN ASCENDING ORDER OF ELEMENTS OF 'IARRAY'.         SORT0020
C**** IF NEVLL.GT.0, SORTED VECTORS ARE STORED IN 'C', OTHERWISE IN     SORT0030
C**** 'RES'. ARRAYS 'EV', 'NDEG', AND 'IARRAY' ARE REORDERED.           SORT0040
      IMPLICIT REAL*8 (A-H,O-Z)                                         SORT0050
      DIMENSION RES(NMAX,NBASIS),C(NMAX,NBASIS),EV(NBASIS),NDEG(NBASIS) SORT0060
      DIMENSION IARRAY(NBASIS),NDG(100)                                 SORT0070
      DIMENSION NDT(100),EVT(100)                                       SORT0080
      NEVAL=IABS(NEVLL)                                                 SORT0090
      IBASE=0                                                           SORT0100
      IEV=0                                                             SORT0110
      IAMAX=0                                                           SORT0120
      DO 2 I=1,NEVAL                                                    SORT0130
      IF (IARRAY(I).GT.IAMAX) IAMAX=IARRAY(I)                           SORT0140
    2 CONTINUE                                                          SORT0150
      DO 50 I=1,IAMAX                                                   SORT0160
      JBASE=0                                                           SORT0170
      DO 51 JEV=1,NEVAL                                                 SORT0180
      ND=NDEG(JEV)                                                      SORT0190
      NDB=IARRAY(JEV)                                                   SORT0200
      IF (NDB.NE.I) GO TO 51                                            SORT0210
      IEV=IEV+1                                                         SORT0220
      NDT(IEV)=NDB                                                      SORT0230
      NDG(IEV)=ND                                                       SORT0240
      DO 52 M=1,ND                                                      SORT0250
      IBM=IBASE+M                                                       SORT0260
      JBM=JBASE+M                                                       SORT0270
      EVT(IBM)=EV(JBM)                                                  SORT0280
      DO 52 J=1,NBASIS                                                  SORT0290
   52 C(J,IBM)=RES(J,JBM)                                               SORT0300
      IBASE=IBASE+ND                                                    SORT0310
   51 JBASE=JBASE+ND                                                    SORT0320
   50 CONTINUE                                                          SORT0330
   55 J=0                                                               SORT0340
      DO 53 I=1,NEVAL                                                   SORT0350
      IARRAY(I)=NDT(I)                                                  SORT0360
      NDEG(I)=NDG(I)                                                    SORT0370
      NDI=NDEG(I)                                                       SORT0380
      DO 54 M=1,NDI                                                     SORT0390
      J=J+1                                                             SORT0400
   54 EV(J)=EVT(J)                                                      SORT0410
   53 CONTINUE                                                          SORT0420
      IF (NEVLL.GT.0) RETURN                                            SORT0430
      DO 1 I=1,NBASIS                                                   SORT0440
      DO 1 J=1,NBASIS                                                   SORT0450
    1 RES(I,J)=C(I,J)                                                   SORT0460
      RETURN                                                            SORT0470
      END                                                               SORT0480
```

```
      SUBROUTINE LINCOM (RES,EV,NDEG,NDIM,LL,C,R,NEVAL,IREPL,NREP,LDEG) LINC0010
C**** TRANSFORMS MATRIX OF PROTOTYPE FUNCTIONS WITH CRUDE SYMMETRY VECTOLINC0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         LINC0030
      DIMENSION IREPL(33),LDEG(14),ISKIP(33)                            LINC0040
      DIMENSION RES(NDIM,LL),EV(LL),C(NDIM),R(NDIM),NDEG(NDIM)          LINC0050
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 LINC0060
      ISUB=0                                                            LINC0070
      DO 7 J=1,LL                                                       LINC0080
      ISKIP(J)=0                                                        LINC0090
      IREPL(J)=0                                                        LINC0100
      DO 7 I=1,LL                                                       LINC0110
      ISUB=ISUB+1                                                       LINC0120
    7 R(ISUB)=RES(I,J)                                                  LINC0130
      NREP=0                                                            LINC0140
   50 DO 5 I=1,LL                                                       LINC0150
      DO 5 J=1,I                                                        LINC0160
    5 RES(I,J)=0.                                                       LINC0170
      I1=1                                                              LINC0180
      DO 1 I=1,NEVAL                                                    LINC0190
      NDI=NDEG(I)                                                       LINC0200
      INDIC=0                                                           LINC0210
      NDI1=I1+NDI-1                                                     LINC0220
      IF (ISKIP(I).NE.0) GO TO 51                                       LINC0230
      NREP=NREP+1                                                       LINC0240
      IREPL(I)=NREP                                                     LINC0250
      LDEG(NREP)=NDI                                                    LINC0260
C**** CALCULATE BLOCKS ALONG DIAGONAL                                   LINC0270
   51 DO 2 M=I1,NDI1                                                    LINC0280
      MIND=(M-1)*NDIM                                                   LINC0290
      DO 2 N=I1,M                                                       LINC0300
      A=0.                                                              LINC0310
      ISUB=0                                                            LINC0320
      NIND=(N-1)*NDIM                                                   LINC0330
      DO 21 L=1,LL                                                      LINC0340
      LPNIND=L+NIND                                                     LINC0350
      DO 21 K=1,LL                                                      LINC0360
      ISUB=ISUB+1                                                       LINC0370
      IF (R(ISUB).EQ.0.) GO TO 21                                       LINC0380
      A=A+C(K+MIND)*R(ISUB)*C(LPNIND)                                   LINC0390
   21 CONTINUE                                                          LINC0400
      IF (DABS(A).LE.CRTZRO) A=0.                                       LINC0410
    2 RES(M,N)=A                                                        LINC0420
      J1=NDI1+1                                                         LINC0430
C**** CALCULATE OFF-DIAGONAL BLOCKS WITH SAME DEGENERACY                LINC0440
      IF (I.EQ.NEVAL) GO TO 1                                           LINC0450
      IP1=I+1                                                           LINC0460
      DO 3 J=IP1,NEVAL                                                  LINC0470
      NDJ=NDEG(J)                                                       LINC0480
      IF (NDI.NE.NDJ) GO TO 3                                           LINC0490
      NDJ1=J1+NDJ-1                                                     LINC0500
      DO 4 M=J1,NDJ1                                                    LINC0510
      MIND=(M-1)*NDIM                                                   LINC0520
      DO 4 N=I1,NDI1                                                    LINC0530
      A=0.                                                              LINC0540
      ISUB=0                                                            LINC0550
      NIND=(N-1)*NDIM                                                   LINC0560
      DO 41 L=1,LL                                                      LINC0570
      LPNIND=L+NIND                                                     LINC0580
```



```
      DO 41 K=1,LL                                                      LINC0590
      ISUB=ISUB+1                                                       LINC0600
      IF (R(ISUB).EQ.0.) GO TO 41                                       LINC0610
      A=A+C(K+MIND)*R(ISUB)*C(LPNIND)                                   LINC0620
   41 CONTINUE                                                          LINC0630
      IF (DABS(A).LE.CRTZRO) A=0.                                       LINC0640
    4 RES(M,N)=A                                                        LINC0650
      IF (ISKIP(I).EQ.1) GO TO 3                                        LINC0660
      DO 40 M=J1,NDJ1                                                   LINC0670
      IF (DABS(RES(M,I1)).GT.CRTZRO) GO TO 42                           LINC0680
   40 CONTINUE                                                          LINC0690
      GO TO 3                                                           LINC0700
   42 ISKIP(J)=1                                                        LINC0710
      IREPL(J)=NREP                                                     LINC0720
    3 J1=J1+NDJ                                                         LINC0730
      I1=I1+NDI                                                         LINC0740
    1 CONTINUE                                                          LINC0750
      DO 6 I=1,LL                                                       LINC0760
      DO 6 J=1,I                                                        LINC0770
    6 RES(J,I)=RES(I,J)                                                 LINC0780
      RETURN                                                            LINC0790
      END                                                               LINC0800
```

```
      SUBROUTINE WIG3J (L1,L2,L3,ISTOP)                                 WIG30010
C**** CALCULATE ALL NON-REDUNDANT 3-J SYMBOLS FOR GIVEN L-VALUES        WIG30020
      IMPLICIT REAL*8 (A-H,O-Z)                                         WIG30030
      COMMON/WIGNER/S3J(17,17),Z3J(17,17)                               WIG30040
      COMMON/FILES/INPUT,IOUT                                           WIG30050
      COMMON/ALP1/THETAC(1224),FACT(50)                                 WIG30060
      IF (L3.GT.L1+L2.OR.L3.LT.IABS(L1-L2)) GO TO 100                   WIG30070
C**** CURRENT VERSION ASSUMES L1=L2=EVEN                                WIG30080
      IF (L1.NE.L2) GO TO 100                                           WIG30090
      MARK=0                                                            WIG30100
      L3HF=L3/2                                                         WIG30110
      IF (2*L3HF.NE.L3) MARK=1                                          WIG30120
      LP1=L3+1                                                          WIG30130
      L1P1=L1+1                                                         WIG30140
C**** LOOP OVER RESTRICTED RANGE OF M-VALUES                            WIG30150
      DO 2 M3P1=1,LP1                                                   WIG30160
      M3=M3P1-1                                                         WIG30170
      MO2=M3/2                                                          WIG30180
      M1MIN=MO2+1                                                       WIG30190
      IF (2*MO2.NE.M3) M1MIN=MO2+2                                      WIG30200
      DO 2 M1P1=M1MIN,L1P1                                              WIG30210
      M1=M1P1-1                                                         WIG30220
      M2=M3-M1                                                          WIG30230
      MSUB1=IABS(M2-M1)/2+1                                             WIG30240
C**** COMPUTATION OF 3-J SYMBOL FOR GENERAL ARGUMENTS                   WIG30250
      N1=L1+L2-L3                                                       WIG30260
      N11=L1-L2+L3                                                      WIG30270
      N12=L2-L1+L3                                                      WIG30280
      N2=L1+M1                                                          WIG30290
      N3=L1-M1                                                          WIG30300
      N4=L2+M2                                                          WIG30310
      N5=L2-M2                                                          WIG30320
      N6=L3+M3                                                          WIG30330
      N7=L3-M3                                                          WIG30340
      N8=L1+L2+L3+1                                                     WIG30350
      FACTOR=(FACT(N1+1)*FACT(N11+1)*FACT(N12+1)/FACT(N8+1))*FACT(N2+1) WIG30360
     1 *FACT(N3+1)*FACT(N4+1)*FACT(N5+1)*FACT(N6+1)*FACT(N7+1)          WIG30370
      FACTOR=DSQRT(FACTOR)                                              WIG30380
      LLM=L1-L2-M3                                                      WIG30390
      IF (2*(LLM/2).NE.LLM) FACTOR=-FACTOR                              WIG30400
      N6=L3-L2+M1                                                       WIG30410
      N7=L3-L1-M2                                                       WIG30420
      KMIN=MAX0(0,-N6,-N7)                                              WIG30430
      KMAX=MIN0(N1,N3,N4)                                               WIG30440
      N1=N1+1                                                           WIG30450
      N3=N3+1                                                           WIG30460
      N4=N4+1                                                           WIG30470
      N6=N6+1                                                           WIG30480
      N7=N7+1                                                           WIG30490
      SUM=0.                                                            WIG30500
      IF (KMIN.GT.0) GO TO 4                                            WIG30510
      KMIN=1                                                            WIG30520
      SUM=1.D0/(FACT(N1)*FACT(N3)*FACT(N4)*FACT(N6)*FACT(N7))           WIG30530
      IF (KMAX.EQ.0) GO TO 6                                            WIG30540
    4 PHASE=1.D0                                                        WIG30550
      IF (2*(KMIN/2).NE.KMIN) PHASE=-1.D0                               WIG30560
      DO 5 K=KMIN,KMAX                                                  WIG30570
      KP1=K+1                                                           WIG30580
```



```
      SUM=SUM+  PHASE   /(FACT(KP1)*FACT(N1-K)*FACT(N3-K)*FACT(N4-K)    WIG30590
     1 *FACT(N6+K)*FACT(N7+K))                                          WIG30600
    5 PHASE=-PHASE                                                      WIG30610
    6 PROD=FACTOR*SUM                                                   WIG30620
      IF (DABS(PROD).LE.1.D0-8) PROD=0.                                 WIG30630
      IF (MARK.EQ.0) GO TO 3                                            WIG30640
      Z3J(MSUB1,M3P1)=PROD                                              WIG30650
      GO TO 2                                                           WIG30660
    3 S3J(MSUB1,M3P1)=PROD                                              WIG30670
    2 CONTINUE                                                          WIG30680
      RETURN                                                            WIG30690
C**** ERROR MESSAGE                                                     WIG30700
  100 WRITE (IOUT,101)                                                  WIG30710
  101 FORMAT (//' ILLEGAL ARGUMENT IN WIG3J')                           WIG30720
      ISTOP=1                                                           WIG30730
      RETURN                                                            WIG30740
      END                                                               WIG30750
```

```
      FUNCTION SYMB3J (M1,M2,M3)                                        SYMB0010
C**** ASSIGNS PROPER PHASE TO 3-J SYMBOL WHERE L3 IS ODD.               SYMB0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         SYMB0030
      COMMON/WIGNER/S3J(17,17),Z3J(17,17)                               SYMB0040
      M3P1=IABS(M3)+1                                                   SYMB0050
      MSUB=IABS(M2-M1)/2+1                                              SYMB0060
      SYMB3J=Z3J(MSUB,M3P1)                                             SYMB0070
      IF (M2.GT.M1) SYMB3J=-SYMB3J                                      SYMB0080
      RETURN                                                            SYMB0090
      END                                                               SYMB0100
```



```
      SUBROUTINE ANTSYM (L3,LL3,FUNCT,IND)                              ANTS0010
C**** CALCULATES VALUE OF ANTISYMMETRIC PRODUCT OF PROTOTYPE FUNCTIONS  ANTS0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         ANTS0030
      COMMON/ASYMI/L,NGRID,L3P1,LP1,LL                                  ANTS0040
      DIMENSION FUNCT(50),SSIN1(33),SSIN2(33),SCOS1(33),SCOS2(33),      ANTS0050
     1 FUN1(50),FUN2(50)                                                ANTS0060
      DATA ROOTHF/0.7071067811865475/,RTHFD/7.0710678118654752/         ANTS0070
      CALL THLMK (L,LL,FUN1,IND)                                        ANTS0080
      IND2=IND-NGRID                                                    ANTS0090
      CALL THLMK (L,LL,FUN2,IND2)                                       ANTS0100
      SSIN1(LP1)=0.                                                     ANTS0110
      SSIN2(LP1)=0.                                                     ANTS0120
      SCOS1(LP1)=FUN1(1)                                                ANTS0130
      SCOS2(LP1)=FUN2(1)                                                ANTS0140
      PHASE=1.D0                                                        ANTS0150
      DO 1 M=1,L                                                        ANTS0160
      PHASE=-PHASE                                                      ANTS0170
      M2=2*M                                                            ANTS0180
      M2P1=M2+1                                                         ANTS0190
      MM=LP1-M                                                          ANTS0200
      MP=LP1+M                                                          ANTS0210
      SSIN1(MM)=-FUN1(M2)*ROOTHF                                        ANTS0220
      SSIN2(MM)=-FUN2(M2)*ROOTHF                                        ANTS0230
      SSIN1(MP)=-PHASE*SSIN1(MM)                                        ANTS0240
      SSIN2(MP)=-PHASE*SSIN2(MM)                                        ANTS0250
      SCOS1(MM)=FUN1(M2P1)*ROOTHF                                       ANTS0260
      SCOS2(MM)=FUN2(M2P1)*ROOTHF                                       ANTS0270
      SCOS1(MP)=PHASE*SCOS1(MM)                                         ANTS0280
    1 SCOS2(MP)=PHASE*SCOS2(MM)                                         ANTS0290
      DO 2 M3P1=1,L3P1                                                  ANTS0300
      M3=M3P1-1                                                         ANTS0310
      IF (M3.GT.0) GO TO 4                                              ANTS0320
      SUM=0.                                                            ANTS0330
      PHASE=-1.D0                                                       ANTS0340
      DO 5 M1=1,L                                                       ANTS0350
      K=M1+LP1                                                          ANTS0360
      PHASE=-PHASE                                                      ANTS0370
      SYMB=SYMB3J(M1,-M1,0)*PHASE                                       ANTS0380
    5 SUM=SUM+(SSIN1(K)*SCOS2(K)-SCOS1(K)*SSIN2(K))*SYMB                ANTS0390
      FUNCT(1)=SUM*10.D0                                                ANTS0400
      GO TO 2                                                           ANTS0410
    4 M32=2*M3                                                          ANTS0420
      SUMA=0.                                                           ANTS0430
      SUMC=0.                                                           ANTS0440
      DO 3 K=1,LL                                                       ANTS0450
      M1=K-LP1                                                          ANTS0460
      M2=M3-M1                                                          ANTS0470
      K2=M2+LP1                                                         ANTS0480
      IF (K2.GT.LL) GO TO 3                                             ANTS0490
      SYMB=SYMB3J(M1,M2,M3)                                             ANTS0500
      SUMA=SUMA+(SCOS1(K)*SSIN2(K2)+SSIN1(K)*SCOS2(K2))*SYMB            ANTS0510
      SUMC=SUMC+(SSIN1(K)*SSIN2(K2)-SCOS1(K)*SCOS2(K2))*SYMB            ANTS0520
    3 CONTINUE                                                          ANTS0530
      FUNCT(M32)=-SUMC*RTHFD                                            ANTS0540
      FUNCT(M32+1)=-SUMA*RTHFD                                          ANTS0550
    2 CONTINUE                                                          ANTS0560
      RETURN                                                            ANTS0570
      END                                                               ANTS0580
```

```
      SUBROUTINE LINEUP (NBASIS,IPAR)                                   LINE0010
C**** LINE UP SYMMETRY VECTORS WITH PROTOTYPE FUNCTIONS                 LINE0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         LINE0030
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 LINE0040
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    LINE0050
     1 LVAL2,IKTYP(14),IRPAR(14),IODD,IRLOWL(14).                       LINE0060
      COMMON/PROTCO/COPRO(1802),EVP(56),IREPL(25),NEVALP(2)             LINE0070
      COMMON/PHASE/IFLAG(20),IPHASE                                     LINE0080
      COMMON/FILES/INPUT,IOUT                                           LINE0090
      COMMON/WORK1/DUMMY(8000)                                          LINE0100
      DIMENSION P(2500),EV(50),C(2500),RECON(5,5),CC(5,5),TEMP(5),      LINE0110
     1 NDEGB(50),IREPLB(50)                                             LINE0120
      DIMENSION IPHOL(8),IPTYPE(2),JFLAG(20)                            LINE0130
      EQUIVALENCE (P,DUMMY),(EV,DUMMY(2501)),(C,DUMMY(2551)),           LINE0140
     1 (RECON,DUMMY(5051)),(CC,DUMMY(5076)),(TEMP,DUMMY(5101)),         LINE0150
     2 (IREPLB,DUMMY(7950)),(NDEGB,DUMMY(7975)),(NEVALB,DUMMY(8000))    LINE0160
      DATA  IPHOL/4HBASI,4HS FN,4HPROT,4H.FN.,4HPROT,4H.FN.,4H NOT,4HHINLINE0170
     1G/                                                                LINE0180
      NDIM=50                                                           LINE0190
      IP1=IPHASE+1                                                      LINE0200
      IPTYPE(1)=IPHOL(2*IPHASE+1)                                       LINE0210
      IPTYPE(2)=IPHOL(2*IPHASE+2)                                       LINE0220
      NEVAL=NEVALP(IPAR)                                                LINE0230
      LL=2*LVAL1+1                                                      LINE0240
      IBASE=0                                                           LINE0250
      ISUB=0                                                            LINE0260
      IF(IPAR.EQ.1) GO TO 4                                             LINE0270
      IBASE=NEVALP(1)                                                   LINE0280
      ISUB=LL**2                                                        LINE0290
      LL=2*LVAL2+1                                                      LINE0300
    4 JI=-1                                                             LINE0310
      DO 61 IEV=1,NEVAL                                                 LINE0320
   61 JFLAG(IEV)=0                                                      LINE0330
      DO 1 I=1,NEVALB                                                   LINE0340
      NDI=NDEGB(I)                                                      LINE0350
      II=-1                                                             LINE0360
      LABELT=0                                                          LINE0370
      DO 3 IEV=1,NEVAL                                                  LINE0380
      LABEL=IREPL(IEV+IBASE)                                            LINE0390
      NDEG=NDEGA(LABEL)                                                 LINE0400
      IF (NDEG.NE.NDI.OR.LABELT.EQ.LABEL) GO TO 3                       LINE0410
      GO TO (51,200,300,511, IP1                                        LINE0420
   00 IF (IREPLB(I).NE.0.AND.IFLAG(LABEL).NE.0) GO TO 1                 LINE0430
      GO TO 51                                                          LINE0440
  100 IF (JFLAG(IEV).EQ.1) GO TO 3                                      LINE0450
      IPTYPE(1)=IPHOL(3)                                                LINE0460
      IPTYPE(2)=IPHOL(4)                                                LINE0470
      IF (IFLAG(LABEL).EQ.0) GO TO 51                                   LINE0480
      IPTYPE(1)=IPHOL(1)                                                LINE0490
      IPTYPE(2)=IPHOL(2)                                                LINE0500
** CALCULATE BLOCK CONNECTING SYMMETRY-ADAPTED SET OF ATOMIC FUNCTIONLINE0510
** WITH SYMMETRIZED PROTOTYPES OF SAME DEGENERACY                      LINE0520
   51 DO 5 M=1,NDI                                                      LINE0530
      ISUB1=(II+M)*LL+ISUB                                              LINE0540
      DO 7 N=1,NDI                                                      LINE0550
      JSUB1=(JI+N)*NDIM                                                 LINE0560
      A=0.                                                              LINE0570
      DO 60 K=1,LL                                                      LINE0580
```



```
      ISUBK=ISUB1+K                                                     LINE0590
      IF (COPRO(ISUBK).EQ.0.) GO TO 60                                  LINE0600
      KSUB=K-NDIM                                                       LINE0610
      DO 6 L=1,NBASIS                                                   LINE0620
      KSUB=KSUB+NDIM                                                    LINE0630
    6 A=A+COPRO(ISUBK)*P(KSUB)*C(JSUB1+L)                               LINE0640
   60 CONTINUE                                                          LINE0650
      IF (DABS(A).LE.CRTZRO) A=0.                                       LINE0660
    7 RECON(M,N)=A                                                      LINE0670
      IF (M.GT.1) GO TO 9                                               LINE0680
C**** TEST FOR NON-ZERO ELEMENT                                         LINE0690
      DO 8 N=1,NDI                                                      LINE0700
      IF (DABS(RECON(1,N)).GT.1.0D-7) GO TO 19                          LINE0710
    8 CONTINUE                                                          LINE0720
      LABELT=LABEL                                                      LINE0730
      GO TO 3                                                           LINE0740
C**** IF INTERACTION IS NON-ZERO, LINE UP PHASES                        LINE0750
   19 IF (IPHASE.NE.1.OR .IFLAG(LABEL).EQ.0) GO TO 40                   LINE0760
      IPTYPE(1)=IPHOL(7)                                                LINE0770
      IPTYPE(2)=IPHOL(8)                                                LINE0780
   40 IF (INTER.EQ.1) WRITE (IOUT,30) I,IPTYPE                          LINE0790
   30 FORMAT (//' PHASE CONNECTION MATRIX FOR EIGENVALUE NO. ',I2,5X,2A4LINE0800
     1,' REPHASED.'/)                                                   LINE0810
      IREPLB(I)=LABEL                                                   LINE0820
    9 ALAM=0.                                                           LINE0830
      DO 10 N=1,NDI                                                     LINE0840
   10 ALAM=ALAM+RECON(M,N)**2                                           LINE0850
      TEMP(M)=DSQRT(ALAM)                                               LINE0860
      IF (INTER.EQ.0) GO TO 50                                          LINE0870
      IF (IPHASE.NE.1) GOTO 45                                          LINE0880
      IF (IFLAG(LABEL).NE.0) LABELT=LABEL                               LINE0890
      IPTYPE(1)=IPHOL(3)                                                LINE0900
      IPTYPE(2)=IPHOL(4)                                                LINE0910
   45 IMM=I1+M+1                                                        LINE0920
      WRITE (IOUT,35) IMM, (RECON(M,N),N=1,NDI)                         LINE0930
   35 FORMAT (' STO. FN. ',I2,1P5E20.10)                                LINE0940
   50 IF (IPHASE.EQ.3) GO TO 1                                          LINE0950
    5 CONTINUE                                                          LINE0960
      DO 11 M=1,NDI                                                     LINE0970
      DO 11 N=1,NDI                                                     LINE0980
   11 CC(N,M)=RECON(M,N)/TEMP(M)                                        LINE0990
      IF  (IPHASE-1) 20,26,28                                           LINE1000
C**** ROTATE PROTOTYPE FUNCTIONS PARALLEL TO FIRST OCCURRENCE OF THAT   LINE1010
C**** REP. (LOWEST L VALUE)                                             LINE1020
   28 IF (IFLAG(LABEL).NE.0) GO TO 20                                   LINE1030
      GO TO 27                                                          LINE1040
   26 IF (IFLAG(LABEL).NE.0) GO TO 3                                    LINE1050
   27 ISUB2=ISUB+II*LL                                                  LINE1060
      DO 21 K=1,LL                                                      LINE1070
      ISUB2=ISUB2+1                                                     LINE1080
      DO 22 N=1,NDI                                                     LINE1090
      A=0.                                                              LINE1100
      ISUB1=ISUB2                                                       LINE1110
      DO 23 M=1,NDI                                                     LINE1120
      ISUB1=ISUB1+LL                                                    LINE1130
   23 A=A+COPRO(ISUB1)*CC(N,M)                                          LINE1140
      IF (DABS(A).LE.CRTZRO) A=0.                                       LINE1150
   22 TEMP(N)=A                                                         LINE1160
```

```
      ISUB1=ISUB2                                                       LINE1170
      DO 24 N=1,NDI                                                     LINE1180
      ISUB1=ISUB1+LL                                                    LINE1190
   24 COPRO(ISUB1)=TEMP(N)                                              LINE1200
   21 CONTINUE                                                          LINE1210
      JFLAG(IEV)=1                                                      LINE1220
      GO TO 3                                                           LINE1230
C**** ROTATE BASIS VECTORS PARALLEL TO PROTOTYPE VECTORS                LINE1240
   20 JM2=J1*NDIM                                                       LINE1250
      DO 12 K=1,NBASIS                                                  LINE1260
      JM2=JM2+1                                                         LINE1270
      DO 13 N=1,NDI                                                     LINE1280
      A=0.                                                              LINE1290
      JM1=JM2                                                           LINE1300
      DO 14 M=1,NDI                                                     LINE1310
      JM1=JM1+NDIM                                                      LINE1320
   14 A=A+C(JM1)*CC(M,N)                                                LINE1330
      IF (DABS(A).LE.CRTZRO) A=0.                                       LINE1340
   13 TEMP(N)=A                                                         LINE1350
      JN1=JM2                                                           LINE1360
      DO 15 N=1,NDI                                                     LINE1370
      JN1=JN1+NDIM                                                      LINE1380
   15 C(JN1)=TEMP(N)                                                    LINE1390
   12 CONTINUE                                                          LINE1400
      GO TO 1                                                           LINE1410
    3 I1=I1+NDEG                                                        LINE1420
    1 J1=J1+NDI                                                         LINE1430
      RETURN                                                            LINE1440
      END                                                               LINE1450
```



```
      SUBROUTINE PHASER                                                 PHAS0010
C**** ROTATES PROTOTYPE FUNCTIONS INTO STANDARD PHASE (AXIS) CHOICE.    PHAS0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         PHAS0030
      REAL*8 LABEL                                                      PHAS0040
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    PHAS0050
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                        PHAS0060
      COMMON/PROTCO/COPRO(1802),EVP(56),IREPL(25),NEVALP(2)             PHAS0070
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 PHAS0080
      COMMON/PHASE/IFLAG(20),IPHASE                                     PHAS0090
      COMMON/UNGRAD/CAU(33),LODD,LODEG                                  PHAS0100
      COMMON/FILES/INPUT,IOUT                                           PHAS0110
      COMMON/WORK1/DUMMY(8000)                                          PHAS0120
      DIMENSION LABEL(50),INDEX(50),IDENT(20),IREPLB(50),NDEGB(50)      PHAS0130
      DIMENSION C(50,50),RES(50,50),B(2500),EV(50),ITITLE(20)           PHAS0140
      EQUIVALENCE (C,DUMMY(2551)),(LABEL,DUMMY(7875)),(INDEX,           PHAS0150
     1 DUMMY(7925)),(IREPLB,DUMMY(7950)),(NDEGB,DUMMY(7975)),           PHAS0160
     2 (NEVALB,DUMMY(8000)),(RES,DUMMY),(EV,DUMMY(2501)),(B,DUMMY(5051))PHAS0170
      DIMENSION IHPAR(2)                                                PHAS0180
      DATA  IDENT/4H STA,4HNDAR,4HD PH,4HASE ,4HCHOI,4HCE V,4HECTO,4HRS PHAS0190
     1 ,12*4H    /, ITITLE/4H FIN,4HAL P,4HROTO,4HTYPE,4H FUN,4HCTIO,   PHAS0200
     2 4HNS, ,4HEVEN,4H PAR,4HITY ,10*4H    /, IHPAR/4HEVEN,4H ODD/      PHAS0210
      EXTERNAL THLMK,AUFUN                                              PHAS0220
      NMAX=50                                                           PHAS0230
      IPHASE=1                                                          PHAS0240
      NR=NREP                                                           PHAS0250
      LL1=2*LVAL1+1                                                     PHAS0260
      DO 6 I=1,NREP                                                     PHAS0270
      IRLOWL(I)=-1                                                      PHAS0280
    6 IFLAG(I)=0                                                        PHAS0290
C**** IDENTIFY A(1G) PROTOTYPE FUNCTION BY LINING UP WITH L=0           PHAS0300
      KALC=1                                                            PHAS0310
      IF (IODD.NE.0) KALC=9                                             PHAS0320
      CALL ADAPT (0,1,THLMK,KALC,1,IDENT,0)                             PHAS0330
      IRAG=IREPLB(1)                                                    PHAS0340
      IF (INTER.EQ.0) GO TO 40                                          PHAS0350
      WRITE (IOUT,30)                                                   PHAS0360
   30 FORMAT (//' REPRESENTATION INDICES'/)                             PHAS0370
      WRITE (IOUT,35) IRAG                                              PHAS0380
   35 FORMAT (16I5)                                                     PHAS0390
   40 IFLAG(IRAG)=1                                                     PHAS0400
      IRLOWL(IRAG)=0                                                    PHAS0410
      IF (IODD.EQ.0) GO TO 10                                           PHAS0420
      NR=NREP/2                                                         PHAS0430
      IFLAG(IRAG+NR)=1                                                  PHAS0440
      NEV1=NEVALP(1)                                                    PHAS0450
      NEVAL=NEVALP(2)                                                   PHAS0460
      LL2=2*LVAL2+1                                                     PHAS0470
      LODD=LVAL2                                                        PHAS0480
      LODEG=LL2                                                         PHAS0490
C**** FIND THE A(1U) FUNCTION                                           PHAS0500
      NNON=0                                                            PHAS0510
      DO 1 I=1,NEVAL                                                    PHAS0520
      IRI=IREPL(I+NEV1)                                                 PHAS0530
      IF (NDEGA(IRI).EQ.1) NNON=NNON+1                                  PHAS0540
    1 CONTINUE                                                          PHAS0550
      IVEC=1                                                            PHAS0560
      IF (NNON.EQ.1) GO TO 2                                            PHAS0570
      IRAU=IRAG+NR                                                      PHAS0580
```

```
      DO 3 I=1,NEVAL
      IF (IREPL(I+NEV1).NE.IRAU) GO TO 3
      IVEC=I
      GO TO 2
    3 CONTINUE
      GO TO 100
    2 LS=LL1*LL1
      IBASE=LS+(IVEC-1)*LL2
      DO 4 I=1,LL2
    4 CAU (I)=COPRO(IBASE+I)*1.003
C**** FIND STANDARD PHASES FOR REPS. WHOSE PARITY MATCHES LOW L VALUE
   10 LMAX=MAXO(LVAL1,LVAL2)
      DO 5 L=1,LMAX
      LDEG=2*L+1
      MPAR=-1
      IF (IODD.EQ.0.OR.2*(L/2).EQ.L) MPAR=1
      CALL ADAPT (L,LDEG,THLMK,KALC,MPAR,IDENT,0)
      IF (INTER.EQ.0) GO TO 45
      WRITE (IOUT,30)
      WRITE (IOUT,35) (IREPLB(J),J=1,NEVALB)
   45 NEVAL=NEVALP(1)
      IEV=0
      IF (2*(L/2).EQ.L.OR.IODD.EQ.0) GO TO 7
      IEV=NEVAL
      NEVAL=NEVALP(2)
    7 DO 8 J=1,NEVAL
      IREP=IREPL(J+IEV)
      IF (IFLAG(IREP).EQ.1) GO TO 8
      DO 9 K=1,NEVALB
      IF (IREPLB(K).NE.IREP) GO TO 9
      IRLOWL(IREP)=L
      IFLAG(IREP)=1
      IF (IODD.EQ.0) GO TO 20
      NRT=NR
      IF (2*(L/2).NE.L) NRT=-NR
      IFLAG(IREP+NRT)=1
   20 DO 14 IR=1,NR
      IF (IFLAG(IR).EQ.0) GO TO 9
   14 CONTINUE
      GO TO 15
    9 CONTINUE
    8 CONTINUE
    5 CONTINUE
C**** REPHASE PROTOTYPES AGAINST EACH OTHER
      DO 16 I=1,NR
      IF (IFLAG(I).EQ.0) GO TO 110
   16 CONTINUE
   15 DO 11 I=1,NREP
   11 IFLAG(I)=IRLOWL(I)+1
      IPHASE=2
      IF (IODD.EQ.0) GO TO 50
      IF (INTER.NE.0) WRITE (IOUT,150)
  150 FORMAT (//' FINAL PHASING OF EVEN-ODD PROTOTYPES.  ODD FUNCTIONS APHAS1110
     1RE "BASIS FNS." ')
      CALL PSYM (LVAL1,LL1,THLMK,LVAL2,LL2,AUFUN,1,-9,RES,B,NMAX)
      ISUB=LS
      DO 22 J=1,LL2
      DO 22 I=1,LL2
      ISUB=ISUB+1
   22 C(I,J)=COPRO(ISUB)
      IEV=NEVALP(1)
      NEVALB=NEVALP(2)
      DO 31 I=1,NEVALB
      IREP=IREPL(I+IEV)
   31 NDEGB(I)=NDEGA(IREP)
      CALL LINEUP (LL2,1)
      IPHASE=0
      ISUB=LS
      DO 23 J=1,LL2
      DO 23 I=1,LL2
      ISUB=ISUB+1
   23 COPRO(ISUB)=C(I,J)
C**** CLEAN UP FINAL VECTORS AND PRINT OUT
   50 DO 55 IPAR=1,2
      IF (IPAR.EQ.2) GO TO 56
      IBASE=0
      NEVAL=NEVALP(1)
      IEV=0
      IS1=0
      LL=LL1
      GO TO 57
   56 IF (IODD.EQ.0) RETURN
      IBASE=LS
      NEVAL=NEVALP(2)
      IS1=LL1
      LL=LL2
      IEV=NEVALP(1)
   57 ISUB=IBASE
      DO 59 I=1,NEVAL
      IREP=IREPL(I+IEV)
      NDEGB(I)=NDEGA(IREP)
   59 IREPLB(I)=IREP
      DO 58 J=1,LL
      EV(J)=EVP(J+IS1)
      DO 58 I=1,LL
      ISUB=ISUB+1
   58 C(I,J)=COPRO(ISUB)
      CALL CLEAN (C,LL,NDEGB,NEVAL,IREPLB,NMAX,CRTZRO,1)
      IF (INTER.EQ.0) GO TO 65
      ITITLE(8)=IHPAR(IPAR)
      CALL MATPRT (LL,1,ITITLE,C,EV,NMAX,LABEL,INDEX)
      WRITE (IOUT,105)
  105 FORMAT (//' REPRESENTATION INDEX FOR EACH SET OF VECTORS'/)
      WRITE (IOUT,35) (IREPLB(I),I=1,NEVAL)
   65 ISUB=IBASE
      DO 60 J=1,LL
      DO 60 I=1,LL
      ISUB=ISUB+1
   60 COPRO(ISUB)=C(I,J)
   55 CONTINUE
      RETURN
C**** ERROR MESSAGES
  100 ISTOP=1
      WRITE (IOUT,345)
  345 FORMAT (' A(1U) FUNCTION NOT FOUND.')
      RETURN
```

```
110 WRITE (IOUT,120)                                          PHAS1750
120 FORMAT (//' ERROR IN PHASING ROUTINE.')                   PHAS1760
    ISTOP=1                                                   PHAS1770
    RETURN                                                    PHAS1780
    END                                                       PHAS1790
```



```
      SUBROUTINE AUFUN (L,LL,FUNCT,IND)                       AUFU0010
C**** EVALUATES ODD-L SPHERICAL HARMONIC TIMES A(IU) FUNCTION AUFU0020
      IMPLICIT REAL*8 (A-H,O-Z)                               AUFU0030
      DIMENSION FUNCT (100),FUNCT1(33)                        AUFU0040
      COMMON/UNGRAD/CAU(33),LODD,LODEG                        AUFU0050
      CALL THLMK (L,LL,FUNCT,IND)                             AUFU0060
      IF (L.EQ.LODD) GO TO 3                                  AUFU0070
      CALL THLMK (LODD,LODEG,FUNCT1,IND)                      AUFU0080
      GO TO 4                                                 AUFU0090
    3 DO 5 I=1,LL                                             AUFU0100
    5 FUNCT1(I)=FUNCT(I)                                      AUFU0110
    4 SUM=0.                                                  AUFU0120
      DO 2 J=1,LODEG                                          AUFU0130
      IF (CAU(J).EQ.0.) GO TO 2                               AUFU0140
      SUM=SUM+FUNCT1(J)*CAU(J)                                AUFU0150
    2 CONTINUE                                                AUFU0160
      DO 1 I=1,LL                                             AUFU0170
    1 FUNCT(I)=FUNCT(I)*SUM                                   AUFU0180
      RETURN                                                  AUFU0190
      END                                                     AUFU0200
```

```
      SUBROUTINE CLEAN (C,NBASIS,NDEG,NEVAL,IREPLB,NMAX,CRTZRO,IOPT)
C**** UNDO ARBITRARY MIXING OF FUNCTIONS OF SAME SYMMETRY TYPE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)
      DIMENSION C(NMAX,NBASIS),NDEG(NEVAL),IREPLB(NEVAL)
      DIMENSION ISV(20),EM(20,20),R(20,20),TEMP(20),ED(20),DIAG(20),
     1 ICM(5)
      COMMON/SCAN1/IFLAG(50),NSC(50),SAMEC(50),JM,IC
      EQUIVALENCE (TEMP,SAMEC)
      DATA  DIAG/2.D0,3.D0,5.D0,7.D0,11.D0,13.D0,17.D0,19.D0,23.D0,
     1 29.D0,31.D0,37.D0,41.D0,43.D0,47.D0,53.D0,59.D0,61.D0,67.D0,
     2 71.D0/
      DO 1 IR=1,NREP
      NDIR=NDEGA(IR)
C**** FIND NUMBER OF SETS BELONGING TO THIS REP.
      NSETS=0
      I1=1
      DO 2 IEV=1,NEVAL
      NDI=NDEG(IEV)
      LABEL=IREPLB(IEV)
      IF (LABEL.NE.IR) GO TO 2
      NSETS=NSETS+1
      ISV(NSETS)=I1
    2 I1=I1+NDI
      IF (NSETS.LE.1) GO TO 1
C**** FIND VECTOR WITH SMALLEST NUMBER OF DISTINCT NONZERO COEFFICIENTS
      DO 20 M=1,NDIR
      JM=ISV(1)+M-1
      CALL SCAN (C,NBASIS,NMAX,CRTZRO)
   20 ICM(M)=IC
      ICMIN=NBASIS
      MINM=0
      DO 24 M=1,NDIR
      IF (ICM(M).GE.ICMIN) GO TO 24
      ICMIN=ICM(M)
      MINM=M-1
   24 CONTINUE
**** CALCULATE ICMIN*NSETS MATRIX
      DO 3 J=1,NSETS
      JM=ISV(J)+MINM
      CALL SCAN (C,NBASIS,NMAX,CRTZRO)
      DO 7 KC=1,IC
      FKC=NSC(KC)
    7 EM(KC,J)=DSQRT(FKC)*SAMEC(KC)
    3 CONTINUE
      IF (IC.EQ.NSETS) GO TO 9
      IF (IOPT.EQ.1) RETURN
**** IF IC.NE.NSETS, FORM SQUARE NSETS*NSETS MATRIX AND DIAGONALIZE
      DO 8 KC=1,NSETS
    8 ED(KC)=DIAG(KC)
      NP1=NSETS+1
      DO 10 KC=NP1,IC
   10 ED(KC)=DIAG(NSETS)
      DO 11 I=1,NSETS
      DO 11 J=1,I
      SUM=0.
      DO 12 K=1,IC
```



```
   12 SUM=SUM+EM(K,I)*ED(K)*EM(K,J)
      R(I,J)=SUM
   11 R(J,I)=SUM
      CALL EIGEN (R,EM,NSETS,TEMP,NSETS,SNORM,20)
      GO TO 13
C**** IF IC=NSETS, TRANSFORMATION MATRIX R = EM(TRANSPOSE)
    9 DO 14 I=1,NSETS
      DO 14 J=1,NSETS
   14 R(I,J)=EM(J,I)
C**** TRANSFORM ALL PARTNER VECTORS IN DEGENERATE SETS
   13 DO 15 M=1,NDIR
      DO 16 K=1,NBASIS
      DO 17 I=1,NSETS
      I1=ISV(I)+M-1
      SUM=0.
      DO 18 J=1,NSETS
      J1=ISV(J)+M-1
   18 SUM=SUM+C(K,J1)*R(J,I)
      IF (DABS(SUM).LE.CRTZRO) SUM=0.
   17 TEMP(I)=SUM
      DO 19 I=1,NSETS
      I1=ISV(I)+M-1
   19 C(K,I1)=TEMP(I)
   16 CONTINUE
   15 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SCAN (C,NBASIS,NMAX,CRTZRO)                            SCAN0010
      IMPLICIT REAL*8 (A-H,O-Z)                                         SCAN0020
      COMMON/SCAN1/IFLAG(50),NSC(50),SAMEC(50),JM,IC                    SCAN0030
      DIMENSION C(NMAX,NBASIS)                                          SCAN0040
      IC=0                                                              SCAN0050
      DO 23 K=1,NBASIS                                                  SCAN0060
      NSC(K)=0                                                          SCAN0070
      SAMEC(K)=0.                                                       SCAN0080
   23 IFLAG(K)=0                                                        SCAN0090
      DO 21 K=1,NBASIS                                                  SCAN0100
      IF (IFLAG(K).EQ.1) GO TO 21                                       SCAN0110
      AC=DABS(C(K,JM))                                                  SCAN0120
      IF (AC.LE.CRTZRO) GO TO 21                                        SCAN0130
      IC=IC+1                                                           SCAN0140
      NSC(IC)=1                                                         SCAN0150
      SAMEC(IC)=C(K,JM)                                                 SCAN0160
      KP1=K+1                                                           SCAN0170
      IF (KP1.GT.NBASIS) GO TO 21                                       SCAN0180
      DO 22 L=KP1,NBASIS                                                SCAN0190
      ACL=DABS(C(L,JM))                                                 SCAN0200
      IF (DABS(ACL-AC).GT.CRTZRO) GO TO 22                              SCAN0210
      IFLAG(L)=1                                                        SCAN0220
      NSC(IC)=NSC(IC)+1                                                 SCAN0230
   22 CONTINUE                                                          SCAN0240
   21 CONTINUE                                                          SCAN0250
      RETURN                                                            SCAN0260
      END                                                               SCAN0270
```



```
      SUBROUTINE SYMSUM (ISUM,JSUM)                                     SYMS0010
C**** SUMMARIZES SYMMETRY PROPERTIES OF MOLECULE AND GENERATES CROSS-   SYMS0020
C**** REFERENCE TABLES                                                  SYMS0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         SYMS0040
      DIMENSION LDEG(14),LDEGO(14),NDJJ(6),IHOL1(5),IHOL2(3)            SYMS0050
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 SYMS0060
      COMMON/FILES/INPUT,IOUT                                           SYMS0070
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    SYMS0080
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                        SYMS0090
      DATA  IHOL1/4H A,B,4H   E,4H   T,4H   U,4H   V/, IHOL2/3H(G),     SYMS0100
     1 3H(U),3H   /                                                     SYMS0110
      DATA  HYES/3HYES/, HNO/3HNO /,ILP /1H(/, IRP/1H)/                 SYMS0120
      ICMPLX=0                                                          SYMS0130
      IF (ISUM.EQ.13.AND.JSUM.EQ.6) ICMPLX=1                            SYMS0140
      IF (ISUM.EQ.15.AND.JSUM.EQ.5) ICMPLX=1                            SYMS0150
      IF (ISUM.EQ.15.AND.JSUM.EQ.10)ICMPLX=1                            SYMS0160
      IF (ISUM.EQ.17.AND.JSUM.EQ.9) ICMPLX=1                            SYMS0170
      IF (ISUM.EQ.17.AND.JSUM.EQ.18)ICMPLX=1                            SYMS0180
      IF (ISUM.EQ.23.AND.JSUM.EQ.14) ICMPLX=1                           SYMS0190
   50 IORDER=0                                                          SYMS0200
      DO 35 I=1,NREP                                                    SYMS0210
      IORDER=IORDER+NDEGA(I)**2                                         SYMS0220
      IF (NDEGA(I).EQ.2.AND.ICMPLX.EQ.1) IORDER=IORDER-2                SYMS0230
   35 CONTINUE                                                          SYMS0240
      IF (IODD.EQ.1) NR=NREP/2                                          SYMS0250
      IQN=0                                                             SYMS0260
      DO 20 I=1,NREP                                                    SYMS0270
      IDEG=NDEGA(I)                                                     SYMS0280
      IRTYP(I)=IHOL1(IDEG)                                              SYMS0290
      IRPAR(I)=IHOL2(1)                                                 SYMS0300
      IF (IODD.EQ.1.AND.I.GT.NREP/2) IRPAR(I)=IHOL2(2)                  SYMS0310
      IF (IODD.EQ.0) IRPAR(I)=IHOL2(3)                                  SYMS0320
      DO 20 ID=1,IDEG                                                   SYMS0330
      IQN=IQN+1                                                         SYMS0340
      IANGA(ID,I)=IQN                                                   SYMS0350
   20 IREPA(IQN)=I                                                      SYMS0360
      HCOMP=HNO                                                         SYMS0370
      IF (ICMPLX.EQ.1) HCOMP=HYES                                       SYMS0380
      WRITE (IOUT,10)                                                   SYMS0390
   10 FORMAT (1H1,'SUMMARY OF POINT GROUP PROPERTIES'/1X,33(1H-)/)      SYMS0400
      WRITE(IOUT,4)NREP,HCOMP,IORDER                                    SYMS0410
    4 FORMAT(5X,'NO. OF IRREDUCIBLE REPRESENTATIONS = ',I2/5X,'COMPLEX-CSYMS0420
     1ONJUGATE PAIRS OF I.R.S. - ',A3/5X,'ORDER OF THE GROUP = ',I3)    SYMS0440
C**** DETERMINE LABEL OF MOLECULAR POINT GROUP                          SYMS0450
      CALL PTGRUP (ICMPLX)                                              SYMS0460
      WRITE(IOUT,32)LVAL1,LVAL2                                         SYMS0470
   32 FORMAT(5X,'L VALUES OF REAL SPHERICAL HARMONICS USED TO SPAN ALL TSYMS0480
     1HE I.R.S IN THIS GROUP - ',2I5)                                   SYMS0490
  100 WRITE (IOUT,21)                                                   SYMS0500
   21 FORMAT (///33X,'REPRESENTATION   DEGENERACY   SPECIES    PHASE',  SYMS0510
     1 9X,'ANGULAR'/37X,'INDEX',22X,'LABEL    STANDARD   QUANTUM NUMBERSSYMS0520
     2'/33X,14('-'),3X,10('-'),3X,7('-'),3X,8('-'),3X,15('-')/)          SYMS0530
      DO 23 I=1,NREP                                                    SYMS0540
      IDEG=NDEGA(I)                                                     SYMS0550
      ILF=IHOL2(3)                                                      SYMS0560
      IRT=ILF                                                           SYMS0570
      IR=I                                                              SYMS0580
      IF (IRLOWL(I).GE.0) GO TO 24                                      SYMS0590
```

```
      IF (IODD.EQ.0) GO TO 24                                           SYMS0600
      IR=I-NR                                                           SYMS0610
      IF (I.LE.NR) IR=I+NR                                              SYMS0620
      ILF=ILP                                                           SYMS0630
      IRT=IRP                                                           SYMS0640
   24 WRITE (IOUT,22) I,IDEG,IRTYP(I),IRPAR(I),ILF,IRLOWL(IR),IRT,(IANGASYMS0650
     1 (ID,I),ID=1,IDEG)                                                SYMS0660
   22 FORMAT (38X,I2,15X,I1,6X,2A4,3X,A1,3HL =,I2,A1,1X,5I5)            SYMS0670
   23 CONTINUE                                                          SYMS0680
      IF (ISTOP.EQ.0) RETURN                                            SYMS0690
      WRITE (IOUT,36)                                                   SYMS0700
   36 FORMAT (1H1,'ERROR IN DETERMINATION OF SYMMETRY PROPERTIES. CALCULSYMS0710
     1ATION ABANDONED.')                                                SYMS0720
      RETURN                                                            SYMS0730
      END                                                               SYMS0740
```



```
      SUBROUTINE PTGRUP (ICMPLX)                                        PTGR0010
C**** DETERMINES WHICH POINT GROUP MOLECULE BELONGS TO                  PTGR0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         PTGR0030
      INTEGER*2 INGRUP,INHOL,IGUESS                                     PTGR0040
      DIMENSION IPTGRP(6),IHOL(36)                                      PTGR0050
      DIMENSION INGRUP(36),INHOL(36)                                    PTGR0060
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 PTGR0070
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    PTGR0080
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                        PTGR0090
      COMMON/FILES/INPUT,IOUT                                           PTGR0100
      DATA  IHOL/4H    ,4H)   ,4H) OR,4H OR ,4HK(H),4HD(H),4HK   ,4HD(6HPTGR0110
     1,4HD(6D),4HT(H),4HT(D),4HO   ,4HD(5D,4HD(5H,4HD(4D,4HD(4H,4HC(6H,PTGR0120
     2 4HT   ,4HD(3D,4HD(3H,4H C(6,4HV) O,4HR D(,4H6)  ,4HC(5H,4HC(5V,PTGR0130
     3 4H D(5,4HD(2H,4HC(4H,4HC(8),4HS(8),4HD(2D,4H C(4,4HV) O,4HR D(,PTGR0140
     4 4H4)  ,4HC(7),4HS(6),4HC(3H,4H C(6,4HC(3V,4H D(3,4HC(5),4HC(2H,PTGR0150
     5 4HC(4),4HS(4),4HC(2V,4H D(2,4HC(3),4HC(1),4HC(2),4HC(S),4HC(1),PTGR0160
     6 4HD(8H,4HD(7H,4HD(8D/                                            PTGR0170
      DATA INGRUP/12101,6050,4901,2461,1230,741,642,531,321,221,110,    PTGR0180
     1 2490,2900,3341,3310,2081,1670,1261,2521,2080,1701,881,1282,1061, PTGR0190
     2 862,441,641,1040,630,440,2450,351,431,220,850,1260/              PTGR0200
      DATA INHOL/5,7,6,10,18,37,38,43,49,50,53,9,55,54,56,13,15,19,8,14,PTGR0210
     1 16,28,17,25,29,44,36,25,40,46,10,29,44,50,31,19/                 PTGR0220
      DO 51 I=1,6                                                       PTGR0230
   51 IPTGRP(I)=IHOL(I)                                                 PTGR0240
C**** COMPUTE INDEX CORRESPONDING TO POINT GROUP                        PTGR0250
      K=0                                                               PTGR0260
      IGUESS=100*IORDER+10*NREP+IODD+ICMPLX                             PTGR0270
      DO 1 I=1,36                                                       PTGR0280
      IF (INGRUP(I).NE.IGUESS) GO TO 1                                  PTGR0290
      IBASE=INHOL(I)                                                    PTGR0300
      IPTGRP(1)=IHOL(IBASE)                                             PTGR0310
      K=I                                                               PTGR0320
      GO TO 2                                                           PTGR0330
    1 CONTINUE                                                          PTGR0340
    2 IF (K.NE.0) GO TO 3                                               PTGR0350
      ISTOP=1                                                           PTGR0360
      RETURN                                                            PTGR0370
    3 IF (K.LE.11) GO TO 50                                             PTGR0380
      IF (K.GE.12.AND.K.LE.26) GO TO 6                                  PTGR0390
      IF (K.GE.27.AND.K.LE.30) GO TO 55                                 PTGR0400
      IF (K.GE.31.AND.K.LE.34) GO TO 45                                 PTGR0410
   65 J=0                                                               PTGR0420
      DO 66 I=1,6                                                       PTGR0430
      IF (I.EQ.2) GO TO 66                                              PTGR0440
      J=J+1                                                             PTGR0450
      IPTGRP(I)=IHOL(J+IBASE)                                           PTGR0460
   66 CONTINUE                                                          PTGR0470
      IPTGRP(2)=IHOL(3)                                                 PTGR0480
      GO TO 50                                                          PTGR0490
   55 DO 70 I=1,2                                                       PTGR0500
   70 IPTGRP(2*I-1)=IHOL(I+IBASE)                                       PTGR0510
      IPTGRP(2)=IHOL(3)                                                 PTGR0520
      IPTGRP(4)=IHOL(2)                                                 PTGR0530
      GO TO 50                                                          PTGR0540
   45 DO 46 I=1,2                                                       PTGR0550
   46 IPTGRP(2*I-1)=IHOL(I+IBASE)                                       PTGR0560
      IPTGRP(2)=IHOL(4)                                                 PTGR0570
      GO TO 50                                                          PTGR0580
```

```
    6 IPTGRP(2)=IHOL(2)                                                 PTGR0590
   50 WRITE (IOUT,100) IPTGRP                                           PTGR0600
  100 FORMAT(5X,'THE POINT GROUP FOR THIS NUCLEAR CONFIGURATION IS',1X,6PTGR0610
     1A4)                                                               PTGR0620
      RETURN                                                            PTGR0630
      END                                                               PTGR0640
```



```
      SUBROUTINE BASIS                                                  BASI0010
C**** READS IN OR LOOKS UP ATOMIC ORBITAL PARAMETERS AND GENERATES CROSSBASI0020
C**** REFERENCE TABLES                                                  BASI0030
      IMPLICIT REAL*8 (A-H,O-Z)                                         BASI0040
      COMMON/FILES/INPUT,IOUT                                           BASI0050
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE, BASI0060
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30),   BASI0070
     2 INVAT(30)                                                        BASI0080
      COMMON/OUTB/BCOEF(500),ZETA(20,10),NVALUE(20,10),LVALUE(20,10),   BASI0090
     1 NBAST(10),ICRK(10),NASB(10),IBLKA(5,10),NBLK,NBVB(50),NBIB(50),  BASI0100
     2 ITB(50),LVB(50),IBIT(20,50),IBCFST(50),NOR,IAT(120),IBT(120),    BASI0110
     3 IKT(120),IAOFST(20,30)                                           BASI0120
      COMMON/OUT2/ASCOEF(3000),ITRANT(20,10),IAOT(40,80),ITRAN(120),    BASI0130
     1 ISRUNT(120),IANGT(120),IRADT(120),IFIRST(80),ISOT(40,80),IFB(50),BASI0140
     2 ISOQ(32,30),NRADA(14)                                            BASI0150
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 BASI0160
      DIMENSION IHOL(5),IRUN(80)                                        BASI0170
      DATA  IHOL/1HS,1HP,1HD,1HF,1HG/, IHOLC1/2HC(/, IHOLC2/1H)/        BASI0180
      NMAX=50                                                           BASI0190
      IAO=0                                                             BASI0200
      ITI=0                                                             BASI0210
      IFI=1                                                             BASI0220
      IFIT=1                                                            BASI0230
      IBLK=0                                                            BASI0240
      CALL INBAS                                                        BASI0250
      IF (ISTOP.EQ.1) RETURN                                            BASI0260
C**** ASSIGN TRANSFORMATION BLOCK PARAMETERS                            BASI0270
   16 DO 150 IT=1,NATYPE                                                BASI0280
      LRF=0                                                             BASI0290
      NM=NUMT(IT)                                                       BASI0300
      NB=NBAST(IT)                                                      BASI0310
      NBR=0                                                             BASI0320
      DO 156 LPO=1,10                                                   BASI0330
      L=LPO-1                                                           BASI0340
      DO 156 IB=1,NB                                                    BASI0350
      IF (L.NE.LVALUE(IB,IT) ) GO TO 159                                BASI0360
    3 NBR=NBR+1                                                         BASI0370
      ITI=ITI+1                                                         BASI0380
      ITRANT(NBR,IT)=ITI                                                BASI0390
      IF (LPO.LE.LRF) GO TO 165                                         BASI0400
      LRF=LPO                                                           BASI0410
      IBLK=IBLK+1                                                       BASI0420
      IFI=IFIT                                                          BASI0430
      IFB(IBLK)=IFI                                                     BASI0440
      IFIT=IFI+(NM*(2*LRF-1))**2                                        BASI0450
  165 IFIRST(ITI)=IFI                                                   BASI0460
  159 IF (NBR.EQ.NB) GO TO 161                                          BASI0470
  156 CONTINUE                                                          BASI0480
C**** GENERATE BASIS FUNCTION REFERENCE TABLES                          BASI0490
  161 DO 150 IN=1,NM                                                    BASI0500
      IA=ITYPA(IN,IT)                                                   BASI0510
      DO 150 IB=1,NB                                                    BASI0520
      LDEG=2*LVALUE(IB,IT)+1                                            BASI0530
      IAOFST(IB,IA)=IAO+1                                               BASI0540
      DO 150 K=1,LDEG                                                   BASI0550
      IAO=IAO+1                                                         BASI0560
      IAT(IAO)=IA                                                       BASI0570
      IBT(IAO)=IB                                                       BASI0580
```

```
150 IKT(IAO)=K                                                          BASI0590
    NOR=IAO                                                             BASI0600
    IF (NOR.GT.120) GO TO 200                                           BASI0610
    IAI=1                                                               BASI0620
    ITR=0                                                               BASI0630
    DO 151 IT=1,NATYPE                                                  BASI0640
    NB=NBAST(IT)                                                        BASI0650
    DO 152 IB=1,NB                                                      BASI0660
    ITR=ITR+1                                                           BASI0670
    IARUN=0                                                             BASI0680
    DO 153 IAO=IAI,NOR                                                  BASI0690
    IA=IAT(IAO)                                                         BASI0700
    ITO=ITYPT(IA)                                                       BASI0710
    IBO=IBT(IAO)                                                        BASI0720
    ITRANS=ITRANT(IBO,ITO)                                              BASI0730
    IF (ITRANS.NE.ITR) GO TO 153                                        BASI0740
    IARUN=IARUN+1                                                       BASI0750
    IAOT(IARUN,ITRANS)=IAO                                              BASI0760
153 CONTINUE                                                            BASI0770
152 CONTINUE                                                            BASI0780
151 IAI=IAI+IARUN                                                       BASI0790
    NBLK=IBLK                                                           BASI0800
    IF (NBLK.GT.50) GO TO 200                                           BASI0810
    NCF=IF(-1                                                           BASI0820
    IF (NCF.LE.3000) GO TO 400                                          BASI0830
    WRITE (IOUT,201)                                                    BASI0840
201 FORMAT (1H1,' NUMBER OF TRANSFORMATION COEFFICIENTS EXCEEDS 3000')BASI0850
    ISTOP=1                                                             BASI0860
    GO TO 400                                                           BASI0870
200 WRITE (IOUT,202)                                                    BASI0880
202 FORMAT (1H1,'PARAMETER SIZE ERROR IN SUBROUTINE BASIS')             BASI0890
    ISTOP=1                                                             BASI0900
C**** PRINTOUT OF CROSS-REFERENCE TABLES                                BASI0910
400 WRITE (IOUT,5)                                                      BASI0920
  5 FORMAT (1H1,'ATOMIC BASIS SET AND SYMMETRY ADAPTATION'/1X,40(1H-)/BASI0930
   1)                                                                   BASI0935
    DO 7 IBLK=1,NBLK                                                    BASI0940
    IT=ITB(IBLK)                                                        BASI0950
    L=LVB(IBLK)                                                         BASI0960
    NB=NBIB(IBLK)                                                       BASI0970
    WRITE (IOUT,6) IBLK,IT ,IHOL(L+1),ITSYMB(IT)                        BASI0980
  6 FORMAT (/' TRANSFORMATION BLOCK ',I2//' ATOM-TYPE ',I2,',  ',       BASI0990
   1 A1,' FUNCTIONS FOR ELEMENT ',A2/)                                  BASI1000
    NSIZE=(2*L+1)*NUMT(IT)                                              BASI1010
    WRITE (IOUT,19) NSIZE,IFB(IBLK)                                     BASI1020
 19 FORMAT (' TOTAL OF ',I3,' BASIS FUNCTIONS FOR EACH BASIS INDEX'// BASI1030
   1 ' BLOCK OF SYMMETRY TRANSFORMATION COEFFICIENTS STARTS AT LOCATIOBASI1040
   2N ',I4,' OF ASCOEF'/)                                               BASI1050
    WRITE (IOUT,8)                                                      BASI1060
  8 FORMAT (' BASIS  TRANSF.  QUANTUM NUMBERS   BASIS VECTORS')         BASI1070
    NV=NBVB(IBLK)                                                       BASI1080
    WRITE (IOUT,9) (IHOLC1,IV,IHOLC2,IV=1,NV)                           BASI1090
  9 FORMAT (' INDEX   INDEX', 6X, 'N    ZETA',5X,10(4X,A2,I2,A1,1X))    BASI1100
    WRITE (IOUT,204)                                                    BASI1110
204 FORMAT (/)                                                          BASI1120
    DO 10 IB=1,NB                                                       BASI1130
    IBI=IBIT(IB,IBLK)                                                   BASI1140
    N=NVALUE(IBI,IT)                                                    BASI1150
```



```
    ZT=ZETA(IBI,IT)                                                     BASI1160
    IFBC=IBCFST(IBLK)+IB-1                                              BASI1170
    ITRANS=ITRANT(IBI,IT)                                               BASI1180
    MBV=(NV-1)*NB+IFBC                                                  BASI1190
    WRITE (IOUT,11) IBI,ITRANS,N,ZT,(BCOEF(I),I=IFBC,MBV,NB)            BASI1200
 11 FORMAT (I4,6X,I2,7X,I2,F9.4,5X,9F10.5)                              BASI1210
 10 CONTINUE                                                            BASI1220
  7 WRITE (IOUT,204)                                                    BASI1230
    IF(INTFR.EQ.0)RETURN                                                BASI1240
    DO 21 I=1,80                                                        BASI1250
 21 IRUN(I)=0                                                           BASI1260
    WRITE (IOUT,20)                                                     BASI1270
 20 FORMAT (1H1,17X,' BASIS    CHEMICAL  ATOM   QUANTUM NUMBERS  JRBITBASI1280
   1AL   TRANS.  TRANS.  RUNNING INDEX  ATOM  BASIS'/18X,'FUNCTION   SBASI1290
   2YMBOL   INDEX      N   L   K     EXPONENT  BLOCK   INDEX   WITHIN TBASI1300
   3RANS.  TYPE  INDEX'/18X,96(1H-)/)                                   BASI1310
    IATT=1                                                              BASI1320
    DO 25 IAO=1,NOR                                                     BASI1330
    IA=IAT(IAO)                                                         BASI1340
    IF (IA.NE.IATT) WRITE (IOUT,204)                                    BASI1350
    IATT=IA                                                             BASI1360
    IT=ITYPT(IA)                                                        BASI1370
    IB=IBT(IAO)                                                         BASI1380
    LP1=LVALUE(IB,IT)+1                                                 BASI1390
    ITRANS=ITRANT(IB,IT)                                                BASI1400
    IRUN(ITRANS)=IRUN(ITRANS)+1                                         BASI1410
 25 WRITE(IOUT,26)IAO,ITSYMB(IT),IA,NVALUE(IB,IT),IHOL(LP1),IKT(IAO),ZBASI1420
   1ETA(IB,IT),IBLKA(LP1,IT),ITRANS,IRUN(ITRANS),IT,IB                  BASI1430
 26 FORMAT (18X,I5,8X,A2,6X,I2,6X,I2,3X,A1,I4,4X,F9.4,4X,I2,5X,I3,8X,   BASI1440
   1 I3,9X,I2,3X,I3/)                                                   BASI1450
    RETURN                                                              BASI1460
    END                                                                 BASI1470
```

```
      SUBROUTINE INBAS                                                  INBA0010
      IMPLICIT REAL*8 (A-H,O-Z)                                         INBA0020
      INTEGER*2 NBASL,ICRL,LVALUL,NVALUL,NBIL,NBVL,IBFL,NASL            INBA0030
      REAL*4 RADILL,RADIL1,RADIL2,RADIL3,RADIL4,BCOFL                   INBA0040
      REAL*4 BC1,BC2,BC3,BC31,BC4,BC5,BC6,BC7                           INBA0050
      COMMON/FILES/INPUT,IOUT                                           INBA0060
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE, INBA0070
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30),   INBA0080
     2 INVAT(30)                                                        INBA0090
      COMMON/OUT8/BCOEF(500),ZETA(20,10),NVALUE(20,10),LVALUE(20,10),   INBA0100
     1 NBAST(10),ICRK(10),NASB(10),IBLKA(5,10),NBLK,NBVB(50),NBIB(50),  INBA0110
     2 ITB(50),LVB(50),IBIT(20,50),IBCFST(50),NOR,IAT(120),IBT(120),    INBA0120
     3 IKT(120),IAOFST(20,30)                                           INBA0130
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 INBA0140
      DIMENSION NBASL(54,2),NASL(54,2),ICRL(54,2),LVALUL(210),RADILL    INBA0150
     1 ( 500),NBIL(210),NBVL(210),IBFL(210),NVALUL( 500),NBLV(4),NSLV(4)INBA0160
     2,RADIL1(110),RADIL2(184),RADIL3(178),RADIL4(13),BCT(20),BCOFL(1100INBA0170
     3),BC1(155),BC2(159),BC3(159),BC31(8),BC4(159),BC5(159),BC6(161),  INBA0180
     4 BC7(111)                                                         INBA0190
      EQUIVALENCE (RADILL,RADIL1),(RADILL(111),RADIL2),(RADILL(295),    INBA0200
     1 RADIL3),(RADILL(473),RADIL4),(BCOFL,BC1),(BCOFL(156),BC2),(BCOFL INBA0210
     2 (315),BC3),(BCOFL(474),BC31),(BCOFL(482),BC4),(BCOFL(641),BC5),  INBA0220
     3 (BCOFL(800),BC6),(BCOFL(961),BC7)                                INBA0230
C**** STORED BASIS SET TABLES                                           INBA0240
      DATA NBLV /1,111,486,0/                                           INBA0250
      DATA NSLV /1,111,249,0/                                           INBA0260
      DATA  NASL/4*1,6*2,2*1,6*2,2*1,10*2,6*3,2*1,10*2,6*3,             INBA0270
     1 4*1,16*2,34*3/                                                   INBA0280
      DATA NBASL /4*1,6*2,2*1,6*2,2*1,10*2,6*3,2*1,10*2,6*3,            INBA0290
     1 1,2,2*3,6*5,2*8,6*10,2*12,10*14,6*16,18*-99/                     INBA0300
      DATA ICRL /2*0,8*2,8*10,18*18,18*36,    54*0/                     INBA0310
      DATA NBIL /110*1,1,2,3*3,2,3,2,3,2,3,2,3,2,3,2,6,2,6,2,6,4,6,4,6, INBA0320
     1 4,6,4,6,4,6,4,8,4,8,4,8,4,2,8,4,2,8,4,2,8,4,2,8,4,2,8,4,2,8,4,2,  INBA0330
     2 8,4,2,8,4,2,8,4,2,8,6,2,8,6,2,8,6,2,8,6,2,8,6,2,8,6,2/            INBA0340
      DATA NBVL /110*1,2*1,3*2,1,2,1,2,1,2,1,2,1,2,1,3,1,3,1,3,2,3,2,3, INBA0350
     1 2,3,2,3,2,3,2,4,2,4,2,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,  INBA0360
     2 4,2,1,4,2,1,4,2,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1/INBA0370
      DATA  LVALUL/5*0,1,0,1,0,1,0,1,0,1,0,1,3*0,1,0,1,0,1,0,1,0,1,0,1,  INBA0380
     1 3*0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,1,0,1,    INBA0390
     2 2,0,1,2,0,1,2,3*0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,1,2,0,  INBA0400
     3 1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,                               INBA0410
     4 5*0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,  INBA0420
     5 1,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,  INBA0430
     6 1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2,0,1,2/                    INBA0440
      DATA IBFL /110*1,1,2,4,10,16,22,24,30,32,38,40,46,48,54,56,62,64, INBA0450
     1 82,84,102,104,122,130,148,156,174,182,200,208,226,234,252,260,292INBA0460
     2,300,332,340,372,380,382,414,422,424,456,464,466,498,506,508,540, INBA0470
     3 548,550,582,590,592,624,632,634,666,674,676,708,716,718,750,758, INBA0480
     4 760,792,810,812,844,852,854,896,914,916,948,966,968,1000,1018,   INBA0490
     5 1020,1052,1070/                                                  INBA0500
      DATA NVALUL /2*1,14*2,14*3,3*4,3,4,3,4,3,4,3,4,3,4,3,4,3,4,3,4,3,4INBA0510
     1,3,2*4,3,2*4,3,2*4,3,2*4,3,2*4,3,2*4,3,3*5,4,5,4,5,4,5,4,5,4,5,4, INBA0520
     2 5,4,5,4,5,4,5,4,2*5,4,2*5,4,2*5,4,2*5,4,2*5,4,2*5,4,             INBA0530
     3 5*1,2,2*1,2,2*1,3*2,2*1,3*2,1,4*2,1,4*2,1,4*2,1,4*2,2*1,2*2,2*3, INBA0540
     4 2*2,2*1,2*2,2*3,2*2,2*1,2*2,2*3,2*2,2*3,2*1,2*2,2*3,2*2,2*3,2*1, INBA0550
     5 2*2,2*3,2*2,2*3,2*1,2*2,2*3,2*2,2*3,2*1,2*2,2*3,2*2,2*3,2*1,2*2, INBA0560
     6 2*3,2*2,2*3,2*1,2*2,2*3,2*4,2*2,2*3,2*1,2*2,2*3,2*4,2*2,2*3,2*1, INBA0570
     7 2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2, INBA0580
```



```
     8 4*3,2*1,2*1,2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3, INBA0590
     9 2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,4*3,2*1,  INBA0600
     A 2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,4*3,2*1,2*2,2*3,2*4,2*2,  INBA0610
     B 2*3,2*4,2*3,2*1,2*2,2*3,2*4,2*2,2*3,2*4,2*3,2*1,2*2,2*3,2*4,2*2,  INBA0620
     C 2*3,2*4,2*3,2*1,2*2,2*3,2*4,2*2,2*3,2*4,2*3,2*1,2*2,2*3,2*4,2*2,  INBA0630
     D 2*3,2*4,2*3,2*1,2*2,2*3,2*4,2*2,2*3,2*4,2*3/                     INBA0640
C**** MINIMAL VALENCE-SHELL BASIS SET                                   INBA0650
C**** H - NE: BAGUS AND GILBERT, UNPUBLISHED WORK                       INBA0660
C**** NA - XF: CLEMENTI, ET AL., J.C.P. 38,2686 (1963); 47, 1300 (1967) INBA0670
      DATA  RADIL1/1.,1.6875,0.640,0.955,1.288,1.211,1.623,1.437,1.924, INBA0680
     1 1.917,2.246,2.227,2.564,2.550,2.879,2.879,0.8358,1.1025,1.3724,  INBA0690
     2 1.3552,1.6344,1.4284,1.8806,1.6288,2.1223,1.8273,2.3561,2.0387,  INBA0700
     3 2.5856,2.2547,0.8738,1.0995,1.1581,2.3733,1.2042,2.7138,1.2453,  INBA0710
     4 2.9943,1.2833,3.2522,1.3208,3.5094,1.3585,3.7266,1.3941,3.9518,  INBA0720
     5 1.4277,4.1765,1.4606,4.4002,1.4913,4.6261,1.7667,1.5554,5.0311,  INBA0730
     6 2.0109,1.6951,5.4171,2.2360,1.8623,5.7928,2.4394,2.0718,6.1590,  INBA0740
     7 2.6382,2.2570,6.5197,2.8289,2.4423,6.8753,0.9969,1.2141,1.2512,  INBA0750
     8 3.9896,1.2891,3.2679,1.3392,3.0796,1.3952,3.1110,1.4453,3.2205,  INBA0760
     9 1.4905,3.3470,1.5286,3.4937,1.5675,3.6476,1.6057,3.8064,1.6384,  INBA0770
     A 3.9692,1.9023,1.6940,4.2354,2.1257,1.8204,4.4925,2.3222,1.9989,  INBA0780
     B 4.7436,2.5076,2.1617,4.9900,2.6807,2.3223,5.2335,2.8436,2.4849,  INBA0790
     C 5.4733/                                                          INBA0800
C**** DOUBLE-ZETA BASIS SET                                             INBA0810
C**** HE - NE, BAGUS AND GILBERT, UNPUBLISHED WORK                      INBA0820
C**** HE - C, K - KR: CLEMENTI, ET AL., J.C.P. 40, 1944 (1964); 47, 1865INBA0830
      DATA RADIL2 /1.,2.905,1.453,2.414,4.353,0.651,3.386,5.781,0.975,  INBA0840
     1 4.353,7.177,1.317,2.218,1.004,5.332,8.698,1.647,2.730,1.255,5.395INBA0850
     2,8.280,1.967,3.247,1.497,7.380,9.569,2.300,3.692,1.656,8.366,     INBA0860
     3 10.885,2.627,4.180,1.848,9.354,12.230,2.952,4.683,2.053,9.7154,  INBA0870
     4 13.147,2.6039,3.9098,0.7549,1.2594,2.5713,5.4964,10.288,13.587,  INBA0880
     5 2.8947,4.3072,0.9076,1.5059,3.0421,6.2078,10.332,13.980,3.5342,  INBA0890
     6 5.0251,1.1595,1.8684,3.6214,7.1261,0.8952,1.6085,9.5094,14.537,  INBA0900
     7 4.1338,6.0540,1.3996,2.2394,4.0852,7.8130,1.0934,1.8626,13.541,  INBA0910
     8 17.774,5.2039,3.1986,1.6390,2.9205,4.5555,8.5171,1.2398,2.0813,  INBA0920
     9 13.717,17.691,3.0757,5.7486,1.8151,3.1596,4.9073,8.9026,1.3217,  INBA0930
     A 2.3336,12.059,17.650,4.9261,6.9833,2.0091,3.3416,5.3574,9.5670,  INBA0940
     B 1.6092,2.8587,14.438,19.131,4.6352,6.7813,2.1847,3.7491,5.0242,  INBA0950
     C 9.0786,1.8117,3.4255,17.290,22.797,7.0793,13.094,2.2893,3.2788,  INBA0960
     D 0.7685,1.2257,6.6382,11.956,3.4843,2.1131,16.716,21.774,7.6735,  INBA0970
     E 9.2638,2.9095,4.1028,0.8790,1.4652,7.2468,13.245,3.6852,2.3446,  INBA0980
     F 17.290,22.757,8.2528,10.858,3.4054,5.3205,1.0507,1.9277,7.6297,  INBA0990
     G 13.495,4.2055,2.6867,1.8554,4.5843,19.167,24.368,8.6061,13.012,  INBA1000
     H 3.1804,4.2714,0.9906,1.7265,8.2114,14.695,5.0016,2.9769,1.9830,  INBA1010
     I 4.6227,20.642,26.150,9.1890,12.722,3.3876,4.6018,0.9417,1.6209,  INBA1020
     J 8.7784,16.087,4.4957,2.8683,2.2166,5.1853,21.777,27.213,9.7064/  INBA1030
      DATA RADIL3 /13.426,3.4616,4.7943,1.1068,2.3994,9.0032,15.341,    INBA1040
     1 5.6266,3.3019,2.0675,5.1217,22.165,27.934,10.218,13.583,3.7964,  INBA1050
     2 5.2770,1.1029,2.0073,9.8608,18.300,5.5196,3.4156,2.5044,5.7542,  INBA1060
     3 23.002,28.980,10.618,14.750,3.9551,5.5099,1.1331,2.0826,10.504,  INBA1070
     4 19.654,5.9885,3.6793,2.6139,6.0576,22.964,29.105,10.938,16.068,  INBA1080
     5 4.1859,5.8357,1.1608,2.1474,10.775,19.074,6.0979,3.7573,2.7400,  INBA1090
     6 6.3762,24.341,30.404,11.542,15.960,4.4269,6.2680,1.2148,2.2966,  INBA1100
     7 11.347,20.265,6.6228,4.0208,2.8305,6.6304,25.652,32.003,12.272,  INBA1110
     8 15.693,4.5848,6.7711,1.0507,2.0583,11.701,20.535,6.7232,4.0324,  INBA1120
     9 2.7577,6.7784,26.434,32.604,12.445,17.458,4.5181,6.5112,1.0949,  INBA1130
     A 2.0327,12.101,20.598,6.7826,4.1606,3.1307,7.3279,24.259,32.432,  INBA1140
     B 12.922,15.645,4.8430,6.8633,1.3233,2.3174,13.046,24.784,7.3992,  INBA1150
     C 5.1345,1.3010,3.5541,3.6671,8.3093,25.178,33.502,13.324,17.041,  INBA1160
```

```
D 5.1905,7.3342,1.4598,2.5623,12.930,21.248,6.8434,4.4787,1.3857,   INBA1170
E 2.7424,3.8559,8.3987,24.928,34.076,13.720,16.130,5.4824,7.4731,   INBA1180
F 1.6823,2.8215,13.362,25.645,8.1667,5.5727,1.6232,3.1345,4.2259,   INBA1190
G 9.0546,25.203,34.967,13.913,18.000,5.8328,8.0342,1.8451,3.0598,   INBA1200
H 14.610,27.813,9.3272,6.0506,1.7714,3.4127,4.5354,9.3545,28.303,   INBA1210
I 37.064,15.316,18.803,6.6824,9.5189,2.3318,4.0670,15.061,28.333,   INBA1220
J 9.0037,6.2067,1.8959,3.5888,4.8816,9.9206,28.941,37.859,15.550/   INBA1230
DATA RADIL4 /19.868,5.5065,8.3700,2.4358,4.0461,15.460,29.224,      INBA1240
1 8.1315,6.0579,1.8095,3.2184,5.1816,10.326/                        INBA1250
DATA BC1 /1.0,0.18159,0.84289,0.86288,0.15286,0.00021,-0.18792,     INBA1260
1 -0.00093,1.02066,0.87859,0.13322,-0.00051,-0.21603,0.00249,1.0263INBA1270
2,0.88708,0.12235,0.00062,-0.24194,0.00684,1.03149,0.21506,0.84089,INBA1280
3 0.90077,0.10695,0.00187,-0.25677,0.01014,1.03413,0.25925,0.80295,INBA1290
4 1.08517,-0.09035,0.00235,-0.23530,-0.01893,1.03375,0.28840,0.7781INBA1300
5,1.07582,-0.08065,0.00313,-0.24017,-0.02125,1.03535,0.33255,.74468INBA1310
6,1.06848,-0.07301,.00379,-.24322,-.02281,1.03628,.35507,.72760,   INBA1320
7 1.06232,-.06661,.00439,-.24534,-.02408,1.03688,.36896,.71694,     INBA1330
9 .73919,.26678,.00088,.00143,.00016,-.00030,-.34000,.03768,.39901,INBA1340
A .68259,-.00049,.00052,.05260,-.00692,-.07599,-.10472,.79808,      INBA1350
B .25733,.73878,.33147,.60598,.40103,.00283,-.00183,.00040,-.00084,INBA1360
C -.34742,.02775,.33845,.75162,.00171,-.00572,.07001,-.00758,       INBA1370
D -.10297,-.14383,.64423,.43983,.74494,.31623,.39604,.61459,.00939,INBA1380
E -.01221,.00103,-.00223,-.32916,-.01558,.48867,.61647,.00149,      INBA1390
F -.00347,.07899,.00174,-.18830,-.12383,.64318,.45352,.77360,       INBA1400
G .26973,-.00662,.02313,-.14620,-.04726,.76411,.30883,.22699,       INBA1410
H .79217,.01514,-.02761,.00143,-.00327,-.31239,-.07524,.66752,      INBA1420
I .47864,.00246,-.00440,.08076,.01924,-.27440,-.08971,.67830,       INBA1430
J.42977,.77780,.26218,-.00473,.01824,-.17507,-.05446,.69577,.38129/INBA1440
DATA BC2 /.74977,.25361,.00143,.00249,.00012,-.00041,-.49236,       INBA1450
1 .12796,.98322,.19647,.04050,-.11416,.11559,-.02116,-.41650,       INBA1460
2 -.00731,.77817,.39127,.78393,.25344,-.00352,.01428,-.19689,       INBA1470
3 -.05826,.60660,.47850,.53980,.46677,.00369,-.00317,.00036,        INBA1480
4 -.00129,-.45693,.08381,.68067,.49502,.02803,-.09538,.12585,       INBA1490
5 -.01924,-.41437,-.07015,.72311,.48648,.75824,.28161,.00287,       INBA1500
6 .00204,-.20738,-.06876,.55335,.55074,.22143,.79375,.01808,        INBA1510
7 -.02554,.00143,-.00476,-.34337,-.06389,.43577,.72240,.00642,      INBA1520
8 -.01993,.10635,.01811,-.34780,-.14259,.69985,.48475,.76189,       INBA1530
9 .27656,.00498,-.00117,-.23282,-.06743,.67838,.43171,.35071,       INBA1540
A .65854,.00694,-.00912,.00065,-.00263,-.44690,.04291,.63699,       INBA1550
B .58355,.03133,-.12009,.13582,-.01047,-.45658,-.10603,.68854,      INBA1560
C .56846,.62615,.45922,.01945,-.06999,-.23934,-.11117,.72656,       INBA1570
D .43099,.77926,.22770,.00354,-.00462,.00101,-.00142,.00015,        INBA1580
E -.00031,-.39810,.05070,1.11024,-.06090,-.01632,.03337,-.00179,    INBA1590
F .00390,.10926,-.00170,-.48580,.06286,.38935,.71592,-.00365,       INBA1600
G .00922,-.01899,-.00100,.09788,-.01455,-.04442,-.17608,.74202,     INBA1610
H .32095,.28733,.04573, -.42376,-.67764,.84278,.17593,.02376,       INBA1620
I -.00576,.46278,.54811,.02112,-.02590,.00210,-.00361,.00020,       INBA1630
J -.00044,-.41612,.03117,1.21977,-.13724,-.01338,.03790,-.00097/    INBA1640
DATA BC3 /.00222,.14019,-.00826,-.72706,.23825,.75652,.36790,       INBA1650
1 -.00549,.01702,.03251,-.00137,-.18955,.06811,.19369,.11258,       INBA1660
2 -.61595,-.48659,.87022,.14168,.03427,-.01145,-.31473,-.03823,     INBA1670
3 .48646,-.61279,.43866,.57563,.02017,-.02835,.00190,-.00445,       INBA1680
4 .00017,-.00047,-.40680,.01604,1.18618,-.11670,-.00793,.05023,     INBA1690
5 -.00059,.00177,.13206,.00208,-.66635,.17686,.99639,.09647,        INBA1700
6 -.00991,.04843,-.03230,.00043,.15740,-.04054,-.29233,-.01502,     INBA1710
7 .84798,.25199,.85312,.16093,.02487,-.00560,-.32486,-.04357,       INBA1720
8 .38439,.71747,.81198,.30493,.56091,.44768,.00458,-.00820,         INBA1730
9 .00107,-.00156,.00009,-.00022,-.43033,.05288,1.16499,-.09783,     INBA1740
```



```
A -.02413,.04730,-.00153,.00398,.13437,-.00489,-.60087,.11632,      INBA1750
B .54854,.56966,-.00422,.01445,-.02953,.00031,.14467,-.03022,       INBA1760
C -.13567,-.15909,.67605,.43415,.86987,.13709,.02895,-.00309,       INBA1770
D -.35118,-.03204,.28760,.82391,.75478,.36564,.66513,.34106,        INBA1780
E .00510,-.00618,.00096,-.00150,.00008,-.00017,-.45273,.07801,      INBA1790
F 1.23751,-.17767,-.02032,.04578,-.00125,.00282,.15120,-.01914,     INBA1800
G -.66710,.17267,.58189,.54381,-.00528,.01433,-.03024,.00178,       INBA1810
H .15438,-.04900,-.09679,-.17711,.52578,.57466,.89362,.11307,       INBA1820
I .03560,-.01121,-.34713,-.03033,.53686,.57076,.76316,.35383,       INBA1830
J .67678,.32681,.00164,-.00026,.00034,-.00051,.00002,-.00008/       INBA1840
DATA BC31 /-.46634,.09153,1.25591,-.19999,-.02876,.05279,-.00095,INBA1850
1 .00615/                                                           INBA1860
DATA BC4 /.15438,-.02252,-.67793,.18395,.50633,.61280,-.00240,      INBA1870
1 .02360,-.03390,.00567,.13145,-.03353,-.19444,-.10244,.87695,      INBA1880
2 .26573,.84670,1.16559,.01755,.00062,-.35925,-.03851,.32197,       INBA1890
3 .80368,.73165,.40849,.60565,.40204,.00787,-.01079,.00124,         INBA1900
4 -.00197,.00010,-.00025,-.46254,.07740,1.28802,-.22353,-.01970,    INBA1910
5 .04773,-.00115,.00309,.15605,-.01825,-.73462,.22283,.63512,       INBA1920
6 .49965,-.00398,.01505,-.03217,.00322,.16597,-.05322,-.14590,      INBA1930
7 -.13103,.70162,.41307,.90656,.09190,.04112,-.00943,-.37082,       INBA1940
8 -.02072,.39236,.72134,.72813,.39159,.59668,.41187,.00733,         INBA1950
9 -.01137,.00127,-.00203,.00010,-.00025,-.46369,.07583,1.24500,     INBA1960
A -.17773,-.01770,.04412,-.00100,.00272,.15367,-.01493,-.70405,     INBA1970
B .18834,.61281,.52512,-.00371,.01398,-.03067,.00221,.15595,        INBA1980
C -.04458,-.13450,-.13631,.69836,.41776,.91337,.07491,.05562,       INBA1990
D -.01391,-.37776,-.01492,.34084,.76903,.71876,.40504,.44554,       INBA2000
E .56468,.00539,-.01174,.00084,-.00143,.00006,-.00015,-.45546,      INBA2010
F .05245,1.18610,-.09988,-.01256,.03593,-.00065,.00177,-.14376,     INBA2020
G -.00094,-.67009,.14521,.64109,.49954,-.00480,.01702,-.02713,      INBA2030
H -.00137,.14666,-.03524,-.13397,-.13100,.69261,.42361,.89679,      INBA2040
I .10142,.03915,-.00864,-.37991,-.02214,.41525,.70533,.71157,       INBA2050
J .41464,.49325,.51462,.00551,-.00931,.00071,-.00128,.00005/        INBA2060
DATA BC5 /-.00013,-.47476,.07494,1.24093,-.16369,-.01321,.04039,    INBA2070
1 -.00068,.00192,.15523,-.01146,-.72787,.19702,.69361,.45118,       INBA2080
2 -.00443,.01764,-.02993,.00141,.15584,-.04499,-.15270,-.11220,     INBA2090
3 .72226,.39760,.90375,.08931,.04431,-.00881,-.38885,-.01707,       INBA2100
4 .36604,.75500,.69757,.43271,.56740,.44064,.01005,-.01387,         INBA2110
5 .00087,-.00179,.00006,-.00014,-.48478,.08788,1.37427,-.30751,     INBA2120
6 -.01221,.04566,-.00065,.00164,.16148,-.01830,-.84612,.31280,      INBA2130
7 .73748,.41625,-.00511,.01735,.02138,-.00002,-.15160,.06305,       INBA2140
8 .09849,.09759,-.72048,-.39435,.89818,.10137,.03356,-.00537,       INBA2150
9 -.38846,-.02035,.43411,.69645,.69521,.44925,.51296,.49324,        INBA2160
A .00341,-.00582,.00035,-.00070,.00003,-.00006,-.49614,.09498,      INBA2170
B 1.24127,-.16224,-.01107,.03504,-.00062,.00147,.16362,-.01821,     INBA2180
C -.72478,.18966,.55056,.60534,-.00249,.00729,-.02346,-.00228,      INBA2190
D .15949,-.05125,-.05278,-.17695,.51599,.59447,.88755,.11397,       INBA2200
E .03143,-.00605,-.38770,-.02536,.48983,.64044,.69487,.43550,       INBA2210
F .26725,.74626,.02116,-.03121,.00146,-.00270,.00012,-.00026,       INBA2220
G -.44786,.00512,1.28168,-.15704,-.01318,.03804,-.00087,.00199,     INBA2230
H .17029,-.00169,-.86319,.28996,.58647,.57498,-.00411,.01166,       INBA2240
I -.03289,-.00322,.23348,-.09601,-.09340,-.19348,.49694,.60674,     INBA2250
J .92478,.05707,.07310,-.03168,-.00088,.00675,-.39101,-.01146/      INBA2260
DATA BC6 /.26343,.76006,-.00256,.10185,.05674,.00306,-.00522,       INBA2270
1 -.20708,.95615,.15095,.74031,.37327,.27375,.74022,.01531,         INBA2280
2 -.02605,.00138,-.00256,.00011,-.00025,-.45471,.01087,1.22777,     INBA2290
3 -.10323,-.01141,.03644,-.00069,.00164,.16452,.00300,-.81330,      INBA2300
4 .23365,.65816,.50750,-.00289,.01029,-.03668,-.00405,.23616,       INBA2310
5 -.08087,-.15364,-.18275,.45342,.66194,.87263,.13162,.03095,       INBA2320
```

```
      6 -.01119,-.00081,.00276,-.38302,-.03619,.62023,.49383,-.00213,  INBA2330
      7 .01453,.07385,.00740,-.12584,-.12484,.80427,.31649,.70033,.40948,INBA2340
      8 .19636,.81497,.01832,-.02646,.00103,-.00191,.00007,-.00017,    INBA2350
      9 -.45154,-.00998,1.21808,-.07289,-.01345,.03675,-.00080,.00189, INBA2360
      A .18269,-.00043,-.90776,.30463,.63255,.53181,-.00538,.01670,    INBA2370
      B .04606,.00233,-.29411,.11899,.16548,.20568,-.47504,-.63742,    INBA2380
      C .92139,.06221,.06160,-.02036,-.00126,.00386,-.40554,-.01242,   INBA2390
      D .24598,.82473,-.01145,.05798,.08935,.00298,-.05037,-.22726,    INBA2400
      E .82895,.27999,.71617,.38692,.17486,.83676,.00785,-.01682,.00040, INBA2410
      F -.00093,.00002,-.00005,-.46228,-.01232,1.11118,.05264,-.00530, INBA2420
      G .02422,-.00019,.00049,.17590,.00865,-.78204,.16523,.71794,     INBA2430
      H .45554,-.00385,.01469,.04711,.00497,-.25806,.06881,.22969,     INBA2440
      I .17806,-.47164,-.64988,.92089,.05183,.06775,-.01455,-.00097,   INBA2450
      J .00296,-.41231,-.00833,.13279,.92648,-.01164,.06609,.09860/    INBA2460
       DATA BCT /.00243,-.02336,-.28098,.82121,.29550,.70183,.39352,   INBA2470
      1 .33846,.67846,.03171,-.04440,.00313,-.00592,.00023,-.00069,    INBA2480
      2 -.45917,.02294,1.40547,-.31512,-.01672,.06469,-.00090,.00283,  INBA2490
      3 .17032,-.00080,-.95336,.37395,1.01774,.10306,-.00690,.04629,   INBA2500
      4 -.05884,.00491,.27763,-.09885,-.46058,.01625,.78349,.36510,    INBA2510
      5 .92724,.05055,.06919,-.02298,-.00138,.00427,-.41861,-.00867,   INBA2520
      6 .19213,.87604,-.00829,.05405,.10742,.00261,-.03805,-.29596,    INBA2530
      7 .73969,.34026,.71135,.37898,.30191,.71223,.02495,-.03566,      INBA2540
      8 .00393,-.00544,.00025,-.00075,-.47092,.02751,1.38570,-.27267,  INBA2550
      A -.03026,.06030,-.00133,.00422,.18287,-.00684,-.96946,.36941,   INBA2560
      B .73304,.41854,-.00322,.02389,-.06059,.00387,.31961,-.12187,    INBA2570
      C -.37484,-.10581,.70702,.45131,.93533,.05118,.07231,-.03686,    INBA2580
      D -.00221,.00572,-.42145,-.01096,.44406,.64246,-.00943,.00309,   INBA2590
      E .11673,.00227,-.14806,-.18834,.57945,.54963,.70696,.37915/     INBA2600
       MVSOPH=2                                                        INBA2610
       MZ=54                                                           INBA2620
       LV=LFLOW(4)                                                     INBA2630
       IBFST=1                                                         INBA2640
       IBLK=1                                                          INBA2650
C**** READ IN OR ASSIGN BASIS SET PARAMETERS                           INBA2660
       IF (LV.GT.MVSOPH.AND.LV.NE.9) GO TO 200                         INBA2670
       IF (LV.NE.9) GO TO 1                                            INBA2680
       NBT=0                                                           INBA2690
       DO 10 IT=1,NATYPE                                               INBA2700
       READ (INPUT,11) NASB(IT),ICRK(IT)                               INBA2710
    11 FORMAT (16I5)                                                   INBA2720
       NAS=NASB(IT)                                                    INBA2730
       IB=0                                                            INBA2740
       DO 3 IAS=1,NAS                                                  INBA2750
       ITB(IBLK)=IT                                                    INBA2760
       READ(INPUT,11) LVB(IBLK),NBIB(IBLK),NBVB(IBLK)                  INBA2770
       L=LVB(IBLK)                                                     INBA2780
       IBLKA(L+1,IT)=IBLK                                              INBA2790
       NBI=NBIB(IBLK)                                                  INBA2800
       DO 5 IBI=1,NBI                                                  INBA2810
       IB=IB+1                                                         INBA2820
       IBIT(IBI,IBLK)=IB                                               INBA2830
       LVALUE(IB,IT)=LVB(IBLK)                                         INBA2840
     5 READ (INPUT,6) NVALUE(IB,IT),ZETA(IB,IT)                        INBA2850
     6 FORMAT (I5,F10.5)                                               INBA2860
       NBV=NBVB(IBLK)                                                  INBA2870
       ISUB=0                                                          INBA2880
       DO 7 IBV=1,NBV                                                  INBA2890
       READ (INPUT,13) (BCT(IBI),IBI=1,NBI)                            INBA2900
```



```
    13 FORMAT (8F10.8)                                                 INBA2910
       DO 9 IBI=1,NBI                                                  INBA2920
       BCOEF(IBFST+ISUB)=BCT(IBI)                                      INBA2930
     9 ISUB=ISUB+1                                                     INBA2940
     7 CONTINUE                                                        INBA2950
       IBCFST(IBLK)=IBFST                                              INBA2960
       IBFST=IBFST+NBI*NBV                                             INBA2970
       NBT=NBT+NBI                                                     INBA2980
     3 IBLK=IBLK+1                                                     INBA2990
       NBAST(IT)=NBT                                                   INBA3000
    10 CONTINUE                                                        INBA3010
       RETURN                                                          INBA3020
     1 DO 157 IT=1,NATYPE                                              INBA3030
       NZ=NUCT(IT)                                                     INBA3040
    14 IF (NZ.GT.MZ) GO TO 20                                          INBA3050
       NBAST(IT)=NBASL(NZ,LV)                                          INBA3060
       IF (NBAST(IT).LT.0) GO TO 20                                    INBA3070
       NASB(IT)=NASL(NZ,LV)                                            INBA3080
       ICRK(IT)=ICPL(NZ,LV)                                            INBA3090
       MSUB=NBLV(LV)                                                   INBA3100
       NSUB=NSLV(LV)                                                   INBA3110
       DO 100 IZ=1,MZ                                                  INBA3120
       IF (IZ.NE.NZ) GO TO 101                                         INBA3130
       GO TO 110                                                       INBA3140
   101 NSUB=NSUB+NASL(IZ,LV)                                           INBA3150
   100 MSUB=MSUB+NBASL(IZ,LV)                                          INBA3160
   110  IB=0                                                           INBA3170
       NAS=NASB(IT)                                                    INBA3180
       DO 105 IAS=1,NAS                                                INBA3190
       ITB(IBLK)=IT                                                    INBA3200
       NS=NSUB+IAS-1                                                   INBA3210
       LVB(IBLK)=LVALUL(NS)                                            INBA3220
       L=LVB(IBLK)                                                     INBA3230
       IBLKA(L+1,IT)=IBLK                                              INBA3240
       NBIB(IBLK)=NBIL(NS)                                             INBA3250
       NBVB(IBLK)=NBVL(NS)                                             INBA3260
       IBFSL=IBFL(NS)                                                  INBA3270
       NBI=NBIB(IBLK)                                                  INBA3280
       NBV=NBVB(IBLK)                                                  INBA3290
       DO 120 IBI=1,NBI                                                INBA3300
       IB=IB+1                                                         INBA3310
       IBIT(IBI,IBLK)=IB                                               INBA3320
       MS=MSUB+IB-1                                                    INBA3330
       NVALUE(IB,IT)=NVALUL(MS)                                        INBA3340
       ZETA(IB,IT)=RADILL(MS)                                          INBA3350
   120 LVALUE(IB,IT)=LVB(IBLK)                                         INBA3360
       ISUB=0                                                          INBA3370
       DO 130 IBV=1,NBV                                                INBA3380
       DO 131 IBI=1,NBI                                                INBA3390
       BCOEF(IBFST+ISUB)=BCOFL(IBFSL+ISUB)                             INBA3400
   131 ISUB=ISUB+1                                                     INBA3410
   130 CONTINUE                                                        INBA3420
       IBCFST(IBLK)=IBFST                                              INBA3430
       IBFST=IBFST+NBI*NBV                                             INBA3440
   105 IBLK=IBLK+1                                                     INBA3450
   157 CONTINUE                                                        INBA3460
       RETURN                                                          INBA3470
   200 WRITE (IOUT,202)                                                INBA3480
```

```
202 FORMAT (1H1,'PARAMETER SIZE ERROR IN SUBROUTINE BASIN')          INBA3490
    ISTOP=1                                                          INBA3500
    RETURN                                                           INBA3510
20 WRITE (IOUT,22) NZ,LV                                             INBA3520
22 FORMAT (1H1//'  STORED BASIS SET OPTION IS NOT IMPLEMENTED FOR NUCINBA3530
   1LEAR CHARGE ',I2,' AND SOPHISTICATION LEVEL ',I2,'.'/'  RESUBMIT UINBA3540
   2SING BASIS SET INPUT OPTION.')                                   INBA3550
    ISTOP=1                                                          INBA3560
    RETURN                                                           INBA3570
    END                                                              INBA3580
```



```
                                                                     ORBT0010
      SUBROUTINE ORBTLS                                              ORBT0020
C**** CALCULATES SYMMETRY-ADAPTED BASIS ORBITALS                     ORBT0030
      IMPLICIT REAL*8 (A-H,O-Z)                                      ORBT0040
      REAL*8 LABEL                                                   ORBT0050
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)              ORBT0060
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE, ORBT0070
     1 NUMT(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30), ORBT0080
     2 INVAT(30)                                                     ORBT0090
      COMMON/OUTB/BCOEF(500),ZETA(20,10),NVALUE(20,10),LVALUE(20,10), ORBT0100
     1 NBAST(10),ICRK(10),NASB(10),IBLKA(5,10),NBLK,NBVB(50),NBIB(50), ORBT0110
     2 ITB(50),LVB(50),IBIT(20,50),IBCFST(50),NOR,IAT(120),IBT(120), ORBT0120
     3 IKT(120),IADFST(20,30)                                        ORBT0130
      COMMON/OUT2/ASCOEF(3000),ITRANT(20,10),IAOT(40,80),ITRAN(120), ORBT0140
     1 ISRUNT(120),IANGT(120),IRADT(120),IFIRST(80),ISOT(40,80),IFB(50),ORBT0150
     2 ISOQ(32,30),NRADA(14)                                         ORBT0160
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1, ORBT0170
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                     ORBT0180
      COMMON/WORK2/DUMM(4),JSUB,LDEG,NUMB,IT                         ORBT0190
      COMMON/FILES/INPUT,IOUT                                        ORBT0200
      COMMON/WORK1/DUMMY(8000)                                       ORBT0210
      DIMENSION LABEL(50),INDEX(50),HOLANG(11,6),IREPLB(50),IDENT(20), ORBT0220
     1 IHOL1(10),IHOL2(7),IANGB(50),RESL(50,50),VALU(50),C(50,50),   ORBT0230
     2 B(50,50),NDEGB(50),IHOL(2),IHHOL(2),IREPT(120)                ORBT0240
      EQUIVALENCE (RESL,DUMMY),(VALU,DUMMY(2501)),(C,DUMMY(2551)),   ORBT0250
     1 (B,DUMMY(5051)),(IANGB,DUMMY(7561)),                          ORBT0260
     2 (LABEL,DUMMY(7875)),(INDEX,DUMMY(7925)),(IREPLB,DUMMY(7950)), ORBT0270
     3 (NDEGB,DUMMY(7975)),(NEVALB,DUMMY(8000))                      ORBT0280
      DATA  IHOL/4H CRU,4HDE S/,  IHHOL/4H FIN,4HAL S/               ORBT0290
      DATA  IDENT/4H CRU,4HDE S,4HYMME,4HTRY ,4HVECT,4HORS ,4HFOR ,4HATOORBT0300
     1M,4H-TYP,11*4H    /, IHOL1/4HE  1,4HE  2,4HE  3,4HE  4,4HE  5,  ORBT0310
     2 4HE  6,4HE  7,4HE  8,4HE  9,4HE 10/, IHOL3/4H, L /,  IHOL2/4H= 0 ORBT0320
     3,4H= 1 ,4H= 2 ,4H= 3 ,4H= 4 ,4H= 5 /                           ORBT0330
      DATA  HOLANG/7H      S ,10*2H  ,7H      Z ,7H      Y ,7H      X ,8*2H ORBT0340
     1 ,7H 3Z2-1 ,7H     YZ ,7H     XZ ,7H     XY ,7H X2-Y2 ,6*2H  ,   ORBT0350
     2 7H 0 COS ,7H 1 SIN ,7H 1 COS ,7H 2 SIN ,7H 2 COS ,7H 3 SIN ,   ORBT0360
     3 7H 3 COS ,7H 4 SIN ,7H 4 COS ,2*2H  ,11*2H  /                 ORBT0370
      EXTERNAL BASFUN                                                ORBT0380
      NMAX=50                                                        ORBT0390
      DO 5 I=1,NREP                                                  ORBT0400
    5 NRADA(I)=0                                                     ORBT0410
      ISO=0                                                          ORBT0420
      DO 1 IBLK=1,NBLK                                               ORBT0430
C**** SYMMETRY-ADAPT SET OF BASIS FUNCTIONS FOR GIVEN L-VALUE, ATOM-TYPEORBT0440
      IT=ITB(IBLK)                                                   ORBT0450
      L=LVB(IBLK)                                                    ORBT0460
      LP1=L+1                                                        ORBT0470
      IDENT(1)=IHOL(1)                                               ORBT0480
      IDENT(2)=IHOL(2)                                               ORBT0490
      IDENT(10)=IHOL1(IT)                                            ORBT0500
      IDENT(11)=IHOL3                                                ORBT0510
      IDENT(12)=IHOL2(LP1)                                           ORBT0520
      LDEG=2*L+1                                                     ORBT0530
      NUMB=NUMT(IT)                                                  ORBT0540
      I=0                                                            ORBT0550
      KFLAG=0                                                        ORBT0560
      DO 30 IA=1,NUMB                                                ORBT0570
      IF (RADIUS(IA).GT.CRTZRO) KFLAG=1                              ORBT0580
      DO 30 N=1,LDEG
```

```
      I=I+1                                                             ORBT0590
      INDEX(I)=IA                                                       ORBT0600
   30 LABEL(I)=HOLANG(N,LP1)                                            ORBT0510
      NBASIS=LDEG*NUMB                                                  ORBT0620
      MPAR=0                                                            ORBT0630
      IF (IODD.EQ.0.OR.KFLAG.EQ.1) GO TO 40                             ORBT0640
      MPAR=1                                                            ORBT0650
      IF (2*(L/2).NE.L) MPAR=-1                                         ORBT0550
   40 CALL ADAPT (L,NBASIS,BASFUN,1,MPAR,IDENT,0)                       ORBT0670
      IF (ISTOP.EQ.1) RETURN                                            ORBT0680
      IDENT(1)=IHHOL(1)                                                 ORBT0690
      IDENT(2)=IHHOL(2)                                                 ORBT0700
      CALL MATPRT (NBASIS,1,IDENT,C,VALU,NMAX,LABEL,INDEX)              ORBT0710
C**** GENERATE CROSS-REFERENCE TABLES                                   ORBT0720
      IND=0                                                             ORBT0730
      DO 11 I=1,NEVALB                                                  ORBT0740
      NDI=NDEGB(I)                                                      ORBT0750
      IR=IREPLB(I)                                                      ORBT0760
      DO 11 J=1,NDI                                                     ORBT0770
      IND=IND+1                                                         ORBT0780
   11 IANGB(IND)=IANGA(J,IR)                                            ORBT0790
      WRITE (IOUT,10) (IANGB(I),I=1,NBASIS)                             ORBT0800
   10 FORMAT (///' ANGULAR QUANTUM NUMBER ASSIGNED TO EACH SYMMETRY VECTORBT0810
     10R'//20I5)                                                        ORBT0820
      NB=NBAST(IT)                                                      ORBT0830
      DO 6 I=1,NB                                                       ORBT0840
      LVI=LVALUE(I,IT)                                                  ORBT0850
      IF (LVI.NE.L) GO TO 6                                             ORBT0860
      ITRANS=ITRANT(I,IT)                                               ORBT0870
      IFI=IFIRST(ITRANS)                                                ORBT0880
      ISRUN=0                                                           ORBT0890
      ISR=IFI                                                           ORBT0900
      DO 8 J=1,NEVALB                                                   ORBT0910
      NDJ=NDEGB(J)                                                      ORBT0920
      IRJ=IREPLB(J)                                                     ORBT0930
C>>>> CHANGE NEXT CARD IF CORRESPONDING DIMENSION IS CHANGED            ORBT0940
      IF (NRADA(IRJ).GT.30) GO TO 20                                    ORBT0950
      NRADA(IRJ)=NRADA(IRJ)+1                                           ORBT0960
      DO 13 M=1,NDJ                                                     ORBT0970
      IF (ISRUN.GT.NMAX) GO TO 20                                       ORBT0980
      ISRUN=ISRUN+1                                                     ORBT0990
      LABT2=0                                                           VECU0910
      DO 12 K=1,NBASIS                                                  ORBT1000
      ASCOEF(ISR)=C(K,ISRUN)                                            ORBT1010
   12 ISR=ISR+1                                                         ORBT1020
      ISO=ISO+1                                                         ORBT1030
      IANG=IANGB(ISRUN)                                                 ORBT1040
      IRAD=NRADA(IRJ)                                                   ORBT1050
      IANGT(ISO)=IANG                                                   ORBT1060
      IRADT(ISO)=IRAD                                                   ORBT1070
      ISRUNT(ISO)=ISRUN                                                 ORBT1080
      ITRAN(ISO)=ITRANS                                                 ORBT1090
   13 CONTINUE                                                          ORBT1100
    8 CONTINUE                                                          ORBT1110
    6 CONTINUE                                                          ORBT1120
    1 CONTINUE                                                          ORBT1130
      DO 151 I=1,NOR                                                    ORBT1140
      IF (IANGT(I).EQ.1) GO TO 151                                      ORBT1150
```



```
      DO 150 J=I,NOR                                                    ORBT1160
      IF (IANGT(I).LT.IANGT(J)) GO TO 150                               ORBT1170
      IF (IANGT(I).EQ.IANGT(J).AND.IRADT(I).LT.IRADT(J)) GO TO 150      ORBT1180
      ITEMP=IANGT(I)                                                    ORBT1190
      IANGT(I)=IANGT(J)                                                 ORBT1200
      IANGT(J)=ITEMP                                                    ORBT1210
      ITEMP=IRADT(I)                                                    ORBT1220
      IRADT(I)=IRADT(J)                                                 ORBT1230
      IRADT(J)=ITEMP                                                    ORBT1240
      ITEMP=ISRUNT(I)                                                   ORBT1250
      ISRUNT(I)=ISRUNT(J)                                               ORBT1260
      ISRUNT(J)=ITEMP                                                   ORBT1270
      ITEMP=ITRAN(I)                                                    ORBT1280
      ITRAN(I)=ITRAN(J)                                                 ORBT1290
      ITRAN(J)=ITEMP                                                    ORBT1300
  150 CONTINUE                                                          ORBT1310
  151 CONTINUE                                                          ORBT1320
      DO 152 I=1,NOR                                                    ORBT1330
      IANG=IANGT(I)                                                     ORBT1340
      IRAD=IRADT(I)                                                     ORBT1350
      IREPT(I)=IREPA(IANG)                                              ORBT1360
      ISRUN=ISRUNT(I)                                                   ORBT1370
      ITRANS=ITRAN(I)                                                   ORBT1380
      ISOT(ISRUN,ITRANS)=I                                              ORBT1390
  152 ISOQ(IANG,IRAD)=I                                                 ORBT1400
C**** PRINTOUT OF TABLES                                                ORBT1410
      WRITE (IOUT,50) (NRADA(I),I=1,NREP)                               ORBT1420
   50 FORMAT (///' NO. OF TIMES EACH REPRESENTATION OCCURS'//14I5)      ORBT1430
      IF(INTER.EQ.0) RETURN                                             ORBT1440
      WRITE (IOUT,155)                                                  ORBT1450
  155 FORMAT (1H1,39X,'***  REFERENCE TABLES FOR SYMMETRY-ADAPTED BASIS ORBT1460
     1 ***'//)                                                          ORBT1470
      WRITE (IOUT,100)                                                  ORBT1480
  100 FORMAT (19X,'SYMMETRY    REP.   SPECIES    QUANTUM NUMBERS     ATOORBT1490
     1M    ATOM-TYPE   TRANS.   RUNNING INDEX'/19X,'FUNCTION   INDEX   ORBT1500
     2LABEL    RADIAL    ANGULAR   SYMBOL     INDEX     INDEX    WITHIN ORBT1510
     3TRANS.'/19X,91(1H-)/)                                             ORBT1520
      IAQNT=1                                                           ORBT1530
      DO 101 ISO=1,NOR                                                  ORBT1540
      ITRANS=ITRAN(ISO)                                                 ORBT1550
      IANG=IANGT(ISO)                                                   ORBT1560
      IREP=IREPA(IANG)                                                  ORBT1570
      IRAD=IRADT(ISO)                                                   ORBT1580
      IF (IANG.NE.IAQNT) WRITE (IOUT,104)                               ORBT1590
  104 FORMAT (1X)                                                       ORBT1500
      IAQNT=IANG                                                        ORBT1610
      DO 105 IT=1,NATYPE                                                ORBT1620
      NB=NBAST(IT)                                                      ORBT1530
      DO 102 IB=1,NB                                                    ORBT1640
      IF (ITRANT(IB,IT).NE.ITRANS) GO TO 102                            ORBT1650
      GO TO 101                                                         ORBT1660
  102 CONTINUE                                                          ORBT1670
  105 CONTINUE                                                          ORBT1680
  101 WRITE (IOUT,103) ISO,IREP,IRTYP(IREP),IRPAR(IREP),IRAD,IANG,      ORBT1690
     1 ITSYMB(IT),IT,ITRANS,ISRUNT(ISO)                                 ORBT1700
  103 FORMAT (19X,I5,7X,I2,4X,2A4,5X,I2,8X,I2,8X,A2,8X,I2,8X,I3,9X,I3/) ORBT1710
      RETURN                                                            ORBT1720
   20 WRITE (IOUT,21)                                                   ORBT1730
```

```
21 FORMAT (//' ARRAY LIMIT EXCEEDED IN SUBROUTINE ORBTLS')      ORBT1740
   ISTOP=1                                                      ORBT1750
   RETURN                                                       ORBT1760
   END                                                          ORBT1770
```



```
      SUBROUTINE BASFUN (L,NBASIS,FUNCT,IND)                    BASF0010
C**** CALCULATES FUNCTION VALUES OF REAL BASIS FUNCTIONS CENTERED BASF0020
C**** ON ATOMS                                                  BASF0030
      IMPLICIT REAL*8 (A-H,O-Z)                                 BASF0040
      COMMON/OUT1/POS(3,30),RADIUS(30),THETA(30),PHI(30),NATOMS,NATYPE, BASF0050
     1 NUM(10),NUCT(10),ITSYMB(10),ITYPA(30,10),ITYPT(30),ITYPI(30), BASF0060
     2 INVAT(30)                                                BASF0070
      COMMON/WORK2/XP,YP,ZP,RP,JSUB,LDEG,NUMB,IT                BASF0080
      COMMON/XYZ/XG(1440),YG(1440),ZG(1440),X(30),Y(30),Z(30)   BASF0090
      DIMENSION FUNCT(100)                                      BASF0100
      JSUB=0                                                    BASF0110
      DO 1 IN=1,NUMB                                            BASF0120
      IA=ITYPA(IN,IT)                                           BASF0130
      XP=XG(IND)-X(IA)                                          BASF0140
      YP=YG(IND)-Y(IA)                                          BASF0150
      ZP=ZG(IND)-Z(IA)                                          BASF0160
      RP=XP*XP+YP*YP+ZP*ZP                                      BASF0170
      CALL VALUE (L,FUNCT)                                      BASF0180
    1 JSUB=JSUB+LDEG                                            BASF0190
      RETURN                                                    BASF0200
      END                                                       BASF0210
```

```
      SUBROUTINE VALUE (L,FUNCT)                                        VALU0010
C**** RETURNS VALUES OF 2*L+1 BASIS FUNCTIONS AT POINT X,Y,Z            VALU0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         VALU0030
      COMMON/WORK2/X,Y,Z,R2,ISUB,LDEG,NUMB,IT                           VALU0040
      DIMENSION FUNCT(100)                                              VALU0050
      DATA  ROOT2/1.4142135623730950/,  ROOT3/1.7320508075688773/,      VALU0060
     1  HRT10/1.5911388300841896/,  HROOT6/1.2247448713915890/,         VALU0070
     2  ROOT15/3.8729833462074168/,  RT35/5.9160797830996160/,          VALU0080
     3  RT70/8.3666002653407554/                                        VALU0090
      LP1=L+1                                                           VALU0100
      GO TO (1,2,3,4,5), LP1                                            VALU0110
C**** S FUNCTION                                                        VALU0120
    1 FUNCT(ISUB+1)=1.D0/R2                                             VALU0130
      RETURN                                                            VALU0140
C**** P FUNCTIONS                                                       VALU0150
    2 R2=1.D0/R2                                                        VALU0160
      FUNCT(ISUB+1)=Z*R2                                                VALU0170
      FUNCT(ISUB+2)=Y*R2                                                VALU0180
      FUNCT(ISUB+3)=X*R2                                                VALU0190
      RETURN                                                            VALU0200
C**** D FUNCTIONS                                                       VALU0210
    3 R3=1.D0/(R2*R2)                                                   VALU0220
      FUNCT(ISUB+1)=0.5D0*(3.D0*Z*Z-R2)*R3                              VALU0230
      RZ=ROOT3*Z*R3                                                     VALU0240
      FUNCT(ISUB+2)=Y*RZ                                                VALU0250
      FUNCT(ISUB+3)=X*RZ                                                VALU0260
      FUNCT(ISUB+4)=ROOT3*X*Y*R3                                        VALU0270
      FUNCT(ISUB+5)=ROOT3*(X*X-Y*Y)*(.5D0*R3)                           VALU0280
      RETURN                                                            VALU0290
C**** F FUNCTIONS                                                       VALU0300
    4 X2=X*X                                                            VALU0310
      Y2=Y*Y                                                            VALU0320
      Z2=Z*Z                                                            VALU0330
      R4=R2*R2                                                          VALU0340
      FACT=5.D0*Z2-R2                                                   VALU0350
      FUNCT(ISUB+1)=Z*(FACT-2.D0*R2)/R4                                 VALU0360
      FACTOR=HROOT6*FACT/R4                                             VALU0370
      FUNCT(ISUB+2)=Y*FACTOR                                            VALU0380
      FUNCT(ISUB+3)=X*FACTOR                                            VALU0390
      FUNCT(ISUB+4)=ROOT15*2.D0*X*Y*Z/R4                                VALU0400
      FUNCT(ISUB+5)=ROOT15*Z*(X2-Y2)/R4                                 VALU0410
      FACTOR=3.D0*(R2-Z2)                                               VALU0420
      FUNCT(ISUB+6)=HRT10*Y*(FACTOR-4.D0*Y2)/R4                         VALU0430
      FUNCT(ISUB+7)=HRT10*X*(4.D0*X2-FACTOR)/R4                         VALU0440
      RETURN                                                            VALU0450
C**** G FUNCTIONS                                                       VALU0460
    5 X2=X*X                                                            VALU0470
      Y2=Y*Y                                                            VALU0480
      Z2=Z*Z                                                            VALU0490
      R5=R2*R2*R2                                                       VALU0500
      FUNCT(ISUB+1)=(17.5D0*Z2*Z2-15.D0*Z2*R2+1.5D0*R2*R2)/R5           VALU0510
      FACTOR=HRT10*Z*(14.D0*Z2-6.D0*R2)/R5                              VALU0520
      FUNCT(ISUB+2)=FACTOR*Y                                            VALU0530
      FUNCT(ISUB+3)=FACTOR*X                                            VALU0540
      FACTOR=ROOT2 *HRT10*(7.D0*Z2-R2)/R5                               VALU0550
      FUNCT(ISUB+4)=X*Y*FACTOR*2.D0                                     VALU0560
      FUNCT(ISUB+5)=(X2-Y2)*FACTOR                                      VALU0570
      CONST=RT70*Z/R5                                                   VALU0580
```



```
      FACTOR=3.D0*(R2-Z2)                                               VALU0590
      FUNCT(ISUB+6)=CONST*Y*(FACTOR-4.D0*Y2)                            VALU0600
      FUNCT(ISUB+7)=CONST*X*(4.D0*X2-FACTOR)                            VALU0610
      FACTOR=FACTOR/3.D0                                                VALU0620
      FUNCT(ISUB+8)=RT35  *X*Y*4.D0*(2.D0*X2-FACTOR)/R5                 VALU0630
      FUNCT(ISUB+9)=RT35  *(4.D0*X2*(X2-FACTOR)+FACTOR*FACTOR)/R5       VALU0640
      RETURN                                                            VALU0650
      END                                                               VALU0660
```

```
      SUBROUTINE VECUPL                                                 VECU0010
C**** CALCULATES FINITE POINT-GROUP VECTOR COUPLING COEFFICIENTS        VECU0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         VECU0030
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 VECU0040
      COMMON/PROTCO/COPRO(1802),EVP(56),IREPL(25),NEVALP(2)             VECU0050
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    VECU0060
     1 LVAL2,IRTYP(14),IPPAR(14),IODD,IRLOWL(14)                        VECU0070
      COMMON/OUT4/VCCOEF(2350),IVCCB(85,14),IFVCC(50)                   VECU0080
      COMMON/WIGNER/S3J(17,17),Z3J(17,17)                               VECU0090
      COMMON/FILES/INPUT,IOUT                                           VECU0100
      COMMON/WORK1/DUMMY(8000)                                          VECU0110
      DIMENSION VCTEMP(25),AA(1089),BB(1089),AO(961),BO(961),IREPLB(50),VECU0120
     1 NDEGS(50)                                                        VECU0130
      EQUIVALENCE (AA,DUMMY),(BB,DUMMY(1090)),(AO,DUMMY(5051)),         VECU0140
     1 (BO,DUMMY(6012)),(VCTEMP,DUMMY(6973)),(IREPLB,DUMMY(7950)),      VECU0150
     2 (NDEGS,DUMMY(7975)),(NEVALS,DUMMY(8000))                         VECU0160
      DATA IHS/1HS/,IHA/1HA/                                            VECU0170
      ROOT2=DSQRT(2.D0)                                                 VECU0180
      NEVAL=NEVALP(1)                                                   VECU0190
      LDEG=2*LVAL1+1                                                    VECU0200
      LP1=LVAL1+1                                                       VECU0210
      NDMAX=3                                                           VECU0220
      IF (IORDER.EQ.24.AND.NREP.EQ.6) NDMAX=2                           VECU0230
      IF (IORDER.EQ.12.AND.NREP.EQ.3) NDMAX=2                           VECU0240
      IANTI=0                                                           VECU0250
      NAQN=0                                                            VECU0260
      NR=NREP                                                           VECU0270
      IF (IODD.NE.0) NR=NREP/2                                          VECU0280
      NR2=NR                                                            VECU0290
C**** DETERMINE IF SOME V.C.C.S ARE ANTISYMMETRIC                       VECU0300
      DO 50 I=1,NP                                                      VECU0310
      IF (NDEGA(I).GT.NDMAX) NR2=NR+2                                   VECU0320
   50 NAQN=NAQN+NDEGA(I)                                                VECU0330
      IF (NAQN.GT.NR) IANTI=1                                           VECU0340
      IF (NDMAX.EQ.2) NR2=NR2-1                                         VECU0350
      ISUB=0                                                            VECU0360
      LODD=0                                                            VECU0370
      IF (IANTI.EQ.1) CALL ODDGA (LODD,LDDEG)                           VECU0380
      LOP1=LODD+1                                                       VECU0390
C**** STORE PROTOTYPE VECTORS IN CONVENIENT FORM                        VECU0400
      DO 1 J=1,LDEG                                                     VECU0410
      JJ=LP1+LDEG*(J-1)                                                 VECU0420
      AA(JJ)=COPRO(ISUB+1)                                              VECU0430
      BB(JJ)=0.                                                         VECU0440
      PHASE=1.D0                                                        VECU0450
      DO 2 M=1,LVAL1                                                    VECU0460
      JP=JJ+M                                                           VECU0470
      JM=JJ-M                                                           VECU0480
      ISUB2M=ISUB+2*M                                                   VECU0490
      PHASE=-PHASE                                                      VECU0500
      COEF=COPRO(ISUB2M+1)/ROOT2                                        VECU0510
      AA(JP)=PHASE*COEF                                                 VECU0520
      AA(JM)=COEF                                                       VECU0530
      COEF=COPRO(ISUB2M)/ROOT2                                          VECU0540
      BB(JP)=PHASE*COEF                                                 VECU0550
    2 BB(JM)=-COEF                                                      VECU0560
    1 ISUB=ISUB+LDEG                                                    VECU0570
      LD2=LDEG*LDEG                                                     VECU0580
```



```
      DO 85 I=1,LD2                                                     VECU0590
      IF (DABS(AA(I)).LE.CRTZRO) AA(I)=0.                               VECU0600
      IF (DABS(BB(I)).LE.CRTZRO) BB(I)=0.                               VECU0610
   85 CONTINUE                                                          VECU0620
C**** GENERATE REQUIRED 3-J SYMBOLS                                     VECU0630
   22 CALL WIG3J (LVAL1,LVAL1,LVAL1,ISTOP)                              VECU0640
      IF (ISTOP.EQ.1) RETURN                                            VECU0650
C**** TRIPLE LOOP OVER SYMMETRY SPECIES AND SUBSPECIES                  VECU0660
      IF (INTER.NE.0) WRITE (IOUT,1100)                                 VECU0670
 1100 FORMAT ('1 UNNORMALIZED V.C.C.S'/)                                VECU0680
      IFV=1                                                             VECU0690
      IVCB=0                                                            VECU0700
      ISUB3=-1                                                          VECU0710
      LABT3=0                                                           VECU0720
      MARK=0                                                            VECU0730
      DO 9 I3=1,NEVAL                                                   VECU0740
      LABEL3=IREPL(I3)                                                  VECU0750
      IR3=LABEL3                                                        VECU0760
      NDEG3=NDEGA(LABEL3)                                               VECU0770
      IF (LABT3.EQ.LABEL3) GO TO 9                                      VECU0780
      IF (NDEG3.LE.NDMAX) LABT3=LABEL3                                  VECU0790
      IF (MARK.EQ.1) LABT3=LABEL3                                       VECU0800
      IF (MARK.EQ.1) IR3=IR3+2                                          VECU0810
      IF (MARK.EQ.1.AND.NDMAX.EQ.2) IR3=IR3-1                           VECU0820
      ISUB1=-1                                                          VECU0830
      LABT1=0                                                           VECU0840
      DO 5 I1=1,NEVAL                                                   VECU0850
      LABEL1=IREPL(I1)                                                  VECU0860
      NDEG1=NDEGA(LABEL1)                                               VECU0870
      IF (LABT1.EQ.LABEL1) GO TO 5                                      VECU0880
      LABT1=LABEL1                                                      VECU0890
      ISUB2=-1                                                          VECU0900
      DO 7 I2=1,I1                                                      VECU0920
      LABEL2=IREPL(I2)                                                  VECU0930
      NDEG2=NDEGA(LABEL2)                                               VECU0940
      IF (LABT2.EQ.LABEL2) GO TO 7                                      VECU0950
      LABT2=LABEL2                                                      VECU0960
      IDPR=LABEL2+LABEL1*(LABEL1-1)/2                                   VECU0970
      IF (IODD.EQ.0) GO TO 21                                           VECU0980
      LAB1NR=LABEL1+NR                                                  VECU0990
      LAB2NR=LABEL2+NR                                                  VECU1000
      IRNR2=IR3+NR2                                                     VECU1010
      LTEM=LAB1NR*(LAB1NR-1)/2                                          VECU1020
      IDPR1=LAB2NR+LTEM                                                 VECU1030
      IDPR2=LABEL2+LTEM                                                 VECU1040
      IDPR3=LABEL1+LAB2NR*(LAB2NR-1)/2                                  VECU1050
   21 IVSUB=0                                                           VECU1060
      DO 10 IG3=1,NDEG3                                                 VECU1070
      ISG3=(ISUB3+IG3)*LDEG                                             VECU1080
      IF (MARK.EQ.0) GO TO 27                                           VECU1090
      IF (NDMAX.EQ.2) GO TO 60                                          VECU1100
      IF (NDEG1.LT.5.OR.NDEG2.LT.4) GO TO 13                            VECU1110
      IF (NDEG3.EQ.4.AND.NDEG2.EQ.4) GO TO 13                           VECU1120
      IF (NDEG3.EQ.4) GO TO 24                                          VECU1130
      GO TO 27                                                          VECU1140
   60 IF (NDEG1.LT.3.OR.NDEG2.LT.3) GO TO 13                            VECU1150
      GO TO 24                                                          VECU1160
C**** SUBSPECIES LOOP FOR SYMMETRIC VECTOR COUPLING COEFFICIENTS        VECU1170
```

```
   27 SNORM=0.                                                          VECU1180
      IVS=1                                                             VECU1190
      DO 6 IG1=1,NDEG1                                                  VECU1200
      ISG1=(ISUB1+IG1)*LDEG                                             VECU1210
      DO 8 IG2=1,NDEG2                                                  VECU1220
      ISG2=(ISUB2+IG2)*LDEG                                             VECU1230
      SUM=AA(ISG1+LP1)*AA(ISG2+LP1)*AA(ISG3+LP1)*S3J(1,1)               VECU1240
      DO 11 K=LP1,LDEG                                                  VECU1250
      M1=K-LP1                                                          VECU1260
      K1=ISG1+K                                                         VECU1270
      LIM=LDEG-M1                                                       VECU1280
      DO 11 N=1,LIM                                                     VECU1290
      IF (M1.EQ.0.AND.N.LE.LP1) GO TO 11                                VECU1300
      N2=ISG2+N                                                         VECU1310
      M2=N-LP1                                                          VECU1320
      M3=M1+M2                                                          VECU1330
      PHASE=2.D0                                                        VECU1340
      IF (2*(M3/2).NE.M3) PHASE=-PHASE                                  VECU1350
      MSUB1=IABS(M2-M1)/2+1                                             VECU1360
      MSUB2=IABS(M3)+1                                                  VECU1370
      IF (MSUB2.GT.LP1) GO TO 11                                        VECU1380
      KN3=M3+LP1+ISG3                                                   VECU1390
      T1=0.                                                             VECU1400
      T2=0.                                                             VECU1410
      IF (AA(KN3).EQ.0.) GO TO 80                                       VECU1420
      T1=(AA(K1)*AA(N2)-BB(K1)*BB(N2))*AA(KN3)                          VECU1430
   80 IF (BB(KN3).EQ.0.) GO TO 82                                       VECU1440
      T2=(AA(K1)*BB(N2)+BB(K1)*AA(N2))*BB(KN3)                          VECU1450
   82 T3=T1+T2                                                          VECU1460
      IF (T3.EQ.0.) GO TO 11                                            VECU1470
      SUM=SUM+T3*S3J(MSUB1,MSUB2)*PHASE                                 VECU1480
   11 CONTINUE                                                          VECU1490
      IF (DABS(SUM).LE.CRTZRO) SUM=0.                                   VECU1500
      VCTEMP(IVS)=SUM                                                   VECU1510
      SNORM=SNORM+SUM*SUM                                               VECU1520
      IVS=IVS+1                                                         VECU1530
    8 CONTINUE                                                          VECU1540
    6 CONTINUE                                                          VECU1550
      IFLAG=IHS                                                         VECU1560
      IF (SNORM.GT.CRTZRO) GO TO 23                                     VECU1570
      IF (LABEL1.EQ.LABEL2.AND.NDEG1.GT.1) GO TO 24                     VECU1580
      IF (LABEL1.EQ.LABEL3.AND.NDEG1.GT.1) GO TO 24                     VECU1590
      IF (LABEL2.EQ.LABEL3.AND.NDEG2.GT.1) GO TO 24                     VECU1600
      GO TO 13                                                          VECU1610
C**** SUBSPECIES LOOP FOR ANTISYMMETRIC V.C.C.S                         VECU1620
   24 ISUB0=-1                                                          VECU1630
      DO 30 I=1,NEVALS                                                  VECU1640
      IF (IREPLB(I).EQ.LABEL3) GO TO 31                                 VECU1650
   30 ISUB0=ISUB0+NDEGS(I)                                              VECU1660
      GO TO 13                                                          VECU1670
   31 ISG3=(ISUB0+IG3)*LDEG                                             VECU1680
      SNORM=0.                                                          VECU1690
      IVS=1                                                             VECU1700
      DO 33 IG1=1,NDEG1                                                 VECU1710
      ISG1=(ISUB1+IG1)*LDEG                                             VECU1720
      DO 33 IG2=1,NDEG2                                                 VECU1730
      ISG2=(ISUB2+IG2)*LDEG                                             VECU1740
      SUM=0.                                                            VECU1750
```



```
      DO 34 K=1,LDEG                                                    VECU1760
      M1=K-LP1                                                          VECU1770
      K1=ISG1+K                                                         VECU1780
      PHASE=2.D0                                                        VECU1790
      DO 34 N=1,LDDEG                                                   VECU1800
      PHASE=-PHASE                                                      VECU1810
      M3=N-LDP1                                                         VECU1820
      N3=ISG3+N                                                         VECU1830
      M2=M3-M1                                                          VECU1840
      IF (IABS(M2).GT.LVALI) GO TO 34                                   VECU1850
      KN2=ISG2+LP1+M2                                                   VECU1860
      T1=0.                                                             VECU1870
      T2=0.                                                             VECU1880
      IF (BO(N3).EQ.0.) GO TO 87                                        VECU1890
      T1=(BB(K1)*BB(KN2)-AA(K1)*AA(KN2))*BO(N3)                         VECU1900
   87 IF (AO(N3).EQ.0.) GO TO 88                                        VECU1910
      T2=(BB(K1)*AA(KN2)+AA(K1)*BB(KN2))*AO(N3)                         VECU1920
   88 T3=T1+T2                                                          VECU1930
      IF (T3.EQ.0.) GO TO 34                                            VECU1940
      SUM=SUM+T3*SYMB3J(M1,M2,M3)*PHASE                                 VECU1950
   34 CONTINUE                                                          VECU1960
      IF (DABS(SUM).LE.1.0D-6) SUM=0.                                   VECU1970
      VCTEMP(IVS)=SUM                                                   VECU1980
      SNORM=SNORM+SUM*SUM                                               VECU1990
      IVS=IVS+1                                                         VECU2000
   33 CONTINUE                                                          VECU2010
      IFLAG=IHA                                                         VECU2020
      IF (SNORM.LE.1.0D-6) GO TO 13                                     VECU2030
   23 RTNORM=DSQRT(SNORM)                                               VECU2040
      IF (IG3.GT.1) GO TO 35                                            VECU2050
      IVCB=IVCB+1                                                       VECU2060
      IAVB=IVCB                                                         VECU2070
      IF (IFLAG.EQ.IHA) IAVB=-IVCB                                      VECU2080
      IVCCB(IDPR,IR3)=IAVB                                              VECU2090
      IFVCC(IVCB)=IFV                                                   VECU2100
   35 IVS=1                                                             VECU2110
      IF (INTER.EQ.0) GO TO 100                                         VECU2120
      ND12=NDEG1*NDEG2                                                  VECU2130
      WRITE (IOUT,1000) LABEL1,LABEL2,LABEL3,IFLAG,(VCTEMP(JVS),JVS=1,  VECU2140
     1 ND12)                                                            VECU2150
 1000 FORMAT (/3I5,4X,1H(,A1,1H)/(1P6D20.10))                           VECU2160
C**** STORE V.C.C.S AND THEIR ADDRESS TABLES                            VECU2170
  100 DO 12 IG1=1,NDEG1                                                 VECU2180
      DO 12 IG2=1,NDEG2                                                 VECU2190
      VCCOEF(IFV+IVSUB)=VCTEMP(IVS)/RTNORM                              VECU2200
      IVSUB=IVSUB+1                                                     VECU2210
      IVS=IVS+1                                                         VECU2220
   12 CONTINUE                                                          VECU2230
      IF (IG3.GT.1) GO TO 10                                            VECU2240
      IF (IODD.EQ.0) GO TO 10                                           VECU2250
      IVCCB(IDPR,IRNR2)=0                                               VECU2260
      IVCCB(IDPR1,IR3)=IAVB                                             VECU2270
      IVCCB(IDPR1,IRNR2)=0                                              VECU2280
      IVCCB(IDPR2,IRNR2)=IAVB                                           VECU2290
      IVCCB(IDPR2,IR3)=0                                                VECU2300
      IVCCB(IDPR3,IR3)=0                                                VECU2310
      IVCCB(IDPR3,IRNR2)=IAVB                                           VECU2320
      GO TO 10                                                          VECU2330
```

```
13 IVCCB(IDPR,IR3)=0                                    VECU2340
   IF (IODD.EQ.0) GO TO 7                               VECU2350
   IVCCB(IDPR,IRNR2)=0                                  VECU2360
   IVCCB(IDPR1,IR3)=0                                   VECU2370
   IVCCB(IDPR1,IRNR2)=0                                 VECU2380
   IVCCB(IDPR2,IRNR2)=0                                 VECU2390
   IVCCB(IDPR2,IR3)=0                                   VECU2400
   IVCCB(IDPR3,IR3)=0                                   VECU2410
   IVCCB(IDPR3,IRNR2)=0                                 VECU2420
   GO TO 7                                              VECU2430
10 CONTINUE                                             VECU2440
   IFV=IFV+NDEG1*NDEG2*NDEG3                            VECU2450
 7 ISUB2=ISUB2+NDEG2                                    VECU2460
 5 ISUB1=ISUB1+NDEG1                                    VECU2470
   IF (MARK.EQ.0) GO TO 91                              VECU2480
   MARK=0                                               VECU2490
   GO TO 9                                              VECU2500
91 IF (NDEG3.GT.NDMAX) MARK=1                           VECU2510
 9 ISUB3=ISUB3+NDEG3                                    VECU2520
   IF (IORDER.EQ.60.OR.IORDER.EQ.120) CALL ORTHOG       VECU2530
   CALL VPRINT (NDMAX)                                  VECU2540
   RETURN                                               VECU2550
   END                                                  VECU2560
```



```
      SUBROUTINE ODDGA (LODD,LL)                                        ODDG0010
C**** GENERATES SYMMETRY-ADAPTED SET OF ANTISYMMETRIC PROTOTYPE FUNCTIONODDG0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         ODDG0030
      REAL*8 LABEL                                                      ODDG0040
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVAL1,    ODDG0060
      COMMON/PROTCO/COPRO(1802),EVP(56),IREPL(25),NEVALP(2)             ODDG0070
    1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                         ODDG0080
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)                 ODDG0090
      COMMON/FILES/INPUT,IOUT                                           ODDG0100
      COMMON/WORK1/DUMMY(8000)                                          ODDG0110
      DIMENSION LABEL(50),INDEX(50),IDENT(20),A0(961),B0(961),VALU(50)  ODDG0120
      DIMENSION LABEL(50),INDEX(50),NDEG(50)                            ODDG0130
    1,C(50,50),IREPLB(50), NDEG(50)                                     ODDG0140
      EQUIVALENCE (VALU,DUMMY(2501)),(C,DUMMY(2551)),(A0,DUMMY(5051)),  ODDG0150
    1 (B0,DUMMY(6012)),(LABEL,DUMMY(7875)),(INDEX,DUMMY(7925)),         ODDG0160
    2 (IREPLB,DUMMY(7950)),(NDEG,DUMMY(7975)),(NEVAL,DUMMY(8000))       ODDG0165
      DIMENSION LAB(5)                                                  ODDG0170
      EXTERNAL THLMK                                                    ODDG0185
      DATA  HOL1/7H   SIN /, HOL2/7H   COS /                            ODDG0190
      DATA  IDENT/4H EVE,4HN PA,4HRITY,4H, OD,4HD-L ,4HFUNC,4HTION,     ODDG0200
    1 4HS AN,4HTISY,4HMMET,4HRIC ,4HIN P,4HARTI,4HCLE ,4HEXCH,4HANGE,   ODDG0210
    2 4*4H    /                                                         ODDG0215
      DATA LAB/2,1,3,5,4/                                               ODDG0220
      NMAX=50                                                           ODDG0230
      LODD=LVAL2                                                        ODDG0240
      IF (IODD.EQ.0) GO TO 2                                            ODDG0250
      NEVAL=NEVALP(2)                                                   ODDG0260
      IBASE=NEVALP(1)                                                   ODDG0270
      DO 11 I=1,NEVAL                                                   ODDG0280
      IREPLB(I)=IREPL(I+IBASE)-NREP/2                                   ODDG0290
      LABLE=IREPLB(I)                                                   ODDG0300
   11 NDEG(I)=NDEGA(LABLE)                                              ODDG0310
      LL=2*LODD+1                                                       ODDG0320
      LL1=2*LVAL1+1                                                     ODDG0330
      ISUB=LL1*LL1                                                      ODDG0340
      DO 1 J=1,LL                                                       ODDG0350
      VALU(J)=EVP(J+LL1)                                                ODDG0360
      DO 1 I=1,LL                                                       ODDG0370
      ISUB=ISUB+1                                                       ODDG0380
    1 C(I,J)=COPRO(ISUB)                                                ODDG0390
      GO TO 8                                                           ODDG0400
    2 IF (LVAL1.EQ.2) LODD=3                                            ODDG0410
      IF (LVAL1.EQ.4) LODD=5                                            ODDG0420
      IF (LVAL1.EQ.6) LODD=9                                            ODDG0430
      IF (LVAL1.GT.6) LODD=15                                           ODDG0440
    3 LL=2*LODD+1                                                       ODDG0450
    8 CALL WIG3J (LVAL1,LVAL1,LODD,ISTOP)                               ODDG0460
      IF (ISTOP.EQ.1) RETURN                                            ODDG0470
      INDEX(1)=0                                                        ODDG0480
      LABEL(1)=HOL2                                                     ODDG0490
      DO 20 M=1,LODD                                                    ODDG0500
      M2=2*M                                                            ODDG0510
      M2P1=M2+1                                                         ODDG0520
      INDEX(M2)=M                                                       ODDG0530
      INDEX(M2P1)=M                                                     ODDG0540
      LABEL(M2)=HOL1                                                    ODDG0550
   20 LABEL(M2P1)=HOL2                                                  ODDG0560
      IF (IODD.EQ.1) GO TO 7                                            ODDG0570
    9 CALL ADAPT (LODD,LL,THLMK,1,1,IDENT,0)
```

```
      IF (ISTOP.EQ.1) RETURN                                    DDDG0580
      IF (NREP.EQ.5.AND.IORDER.EQ.24) GO TO 30                  DDDG0581
      GO TO 7                                                   DDDG0582
   30 DO 35 I=1,NEVAL                                           DDDG0583
      ILAB=IREPLB(I)                                            DDDG0584
      DO 36 J=1,5                                               DDDG0585
      IF (ILAB.EQ.J) IREPLB(I)=LAB(J)                           DDDG0586
   36 CONTINUE                                                  DDDG0587
   35 CONTINUE                                                  DDDG0588
    7 IF (INTER.EQ.0) GO TO 4                                   DDDG0590
      CALL MATPRT (LL,1,IDENT,C,VALU,NMAX,LABEL,INDEX)          DDDG0500
      WRITE (IOUT,12)                                           DDDG0610
   12 FORMAT (//' REPRESENTATION INDEX ASSIGNED TO EACH SET OF VECTORS')DDDG0620
      WRITE (IOUT,10) (IREPLB(I),I=1,NEVAL)                     DDDG0630
   10 FORMAT (//1615)                                           DDDG0640
    4 ROOT2=DSQRT(2.D0)                                         DDDG0550
      LP1=LODD+1                                                DDDG0660
      ISUB=0                                                    DDDG0670
      DO 5 I=1,LL                                               DDDG0680
      JJ=ISUB+LP1                                               DDDG0690
      AO(JJ)=C(1,I)                                             DDDG0700
      BO(JJ)=0.                                                 DDDG0710
      PHASE=1.D0                                                DDDG0720
      DO 6 M=1,LODD                                             DDDG0730
      JP=JJ+M                                                   DDDG0740
      JM=JJ-M                                                   DDDG0750
      M2=2*M                                                    DDDG0760
      PHASE=-PHASE                                              DDDG0770
      COEF=C(M2+1,I)/ROOT2                                      DDDG0780
      AO(JP)=PHASE*COEF                                         DDDG0790
      AO(JM)= COEF                                              DDDG0800
      COEF=C(M2,I)/ROOT2                                        DDDG0810
      BO(JP)=PHASE*COEF                                         DDDG0820
    6 BO(JM)=-COEF                                              DDDG0830
    5 ISUB=ISUB+LL                                              DDDG0840
      RETURN                                                    DDDG0850
      END                                                       DDDG0860
```



```
      SUBROUTINE ORTHOG                                         ORTH0010
C**** SCHMIDT ORTHOGONALIZES DOUBLY-OCCURRING SETS OF V.C.C.S FOR POINT ORTH0020
C**** GROUPS K AND K (H)                                        ORTH0030
      IMPLICIT REAL*8 (A-H,O-Z)                                 ORTH0040
      COMMON/CONTRL/CRTZRO,CRTDEG,ISTOP,INTER,LFLOW(10)         ORTH0050
      COMMON/OUT4/VCCOEF(2350),IVCCB(85,14),IFVCC(50)           ORTH0060
      DIMENSION A(25),B(25)                                     ORTH0070
      DIMENSION C(25,10),NDEG(2),IREPLB(2)                      ORTH0080
      NCOEF=20                                                  ORTH0090
      IND=14                                                    ORTH0100
    4 IVCB1=IABS(IVCCB(IND,5))                                  ORTH0110
      IVCB2=IABS(IVCCB(IND,7))                                  ORTH0120
      IFV1=IFVCC(IVCB1)-1                                       ORTH0130
      IFV2=IFVCC(IVCB2)-1                                       ORTH0140
      KK=0                                                      ORTH0150
      DO 2 K=1,5                                                ORTH0160
      SUM=0.                                                    ORTH0170
      DO 1 I=1,NCOEF                                            ORTH0180
      J=I+KK                                                    ORTH0190
      A(I)=VCCOEF(IFV1+J)                                       ORTH0200
      B(I)=VCCOEF(IFV2+J)                                       ORTH0210
    1 SUM=SUM+A(I)*B(I)                                         ORTH0220
      IF (K.GT.1) GO TO 5                                       ORTH0230
      SNORM=DSQRT(1.D0-SUM*SUM)                                 ORTH0240
    5 DO 3 I=1,NCOEF                                            ORTH0250
      J=I+KK                                                    ORTH0260
      TERM=(B(I)-SUM*A(I))/SNORM                                ORTH0270
      IF (DABS(TERM).LE.CRTZRO) TERM=0.                         ORTH0280
    3 VCCOEF(IFV2+J)=TERM                                       ORTH0290
    2 KK=KK+NCOEF                                               ORTH0300
      IF (IND.EQ.14) GO TO 6                                    ORTH0310
C**** SIMPLIFY SETS OF COEFFICIENTS FOR  V X V = V              ORTH0320
      DO 7 I=1,2                                                ORTH0330
      NDEG(I)=5                                                 ORTH0340
    7 IREPLB(I)=5                                               ORTH0350
      KK=0                                                      ORTH0360
      DO 8 K=1,5                                                ORTH0370
      DO 9 I=1,25                                               ORTH0380
      J=I+KK                                                    ORTH0390
      C(I,K)=VCCOEF(IFV1+J)                                     ORTH0400
    9 C(I,K+5)=VCCOEF(IFV2+J)                                   ORTH0410
    8 KK=KK+25                                                  ORTH0420
      CALL CLEAN (C,25,NDEG,2,IREPLB,25,CRTZRO,0)               ORTH0430
      KK=0                                                      ORTH0440
      DO 10 K=1,5                                               ORTH0450
      DO 11 I=1,25                                              ORTH0460
      J=I+KK                                                    ORTH0470
      VCCOEF(IFV1+J)=C(I,K)                                     ORTH0480
   11 VCCOEF(IFV2+J)=C(I,K+5)                                   ORTH0490
   10 KK=KK+25                                                  ORTH0500
      IF (IND.EQ.15) RETURN                                     ORTH0510
    6 IND=15                                                    ORTH0520
      NCOEF=25                                                  ORTH0530
      GO TO 4                                                   ORTH0540
      END                                                       ORTH0550
```

```
      SUBROUTINE VPRINT (NDMAX)                                         VPRI0010
C**** PRINTOUT OF SELECTION RULE AND V.C.C. TABLES                      VPRI0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         VPRI0030
      COMMON/OUT3/NREP,NDEGA(14),IANGA(5,14),IREPA(32),IORDER,LVALI,    VPRI0040
     1 LVAL2,IRTYP(14),IRPAR(14),IODD,IRLOWL(14)                        VPRI0050
      COMMON/OUT4/VCCOEF(2350),IVCCB(85,14),IFVCC(50)                   VPRI0060
      COMMON/FILES/INPUT,IOUT                                           VPRI0070
      DIMENSION IDP(14),LINE(2)                                         VPRI0080
      DATA  LINE/4H----,3H---/                                          VPRI0090
      NR=NREP                                                           VPRI0100
      IF (IODD.NE.0) NR=NR/2                                            VPRI0110
      NLARK=2                                                           VPRI0120
      LARK=0                                                            VPRI0130
      DO 20 I=1,NR                                                      VPRI0140
      IF (NDEGA(I).GT.NDMAX) LARK=1                                     VPRI0150
   20 CONTINUE                                                          VPRI0160
      IF (NDMAX.EQ.2) NLARK=1                                           VPRI0170
      WRITE (IOUT,10)                                                   VPRI0180
   10 FORMAT (1H1,'SELECTION RULES FOR DIRECT PRODUCTS OF IRREDUCIBLE REVPRI0190
     1PRESENTATIONS'/1X,66(1H-)//)                                      VPRI0200
      WRITE (IOUT,11)                                                   VPRI0210
   11 FORMAT (28X,'REP.3 -->'/)                                         VPRI0220
      WRITE (IOUT,12) (IRTYP(I),IRPAR(I),I=1,NR)                        VPRI0230
   12 FORMAT (28X,14(A4,A3))                                            VPRI0240
      WRITE(IOUT,13) (I,I=1,NR)                                         VPRI0250
   13 FORMAT (7X,'REP.1 X REP.2',5X,14I7)                               VPRI0260
      WRITE (IOUT,14) ((LINE(I),I=1,2),J=1,NR)                          VPRI0270
   14 FORMAT (1X,25(1H-),2X,14(A4,A3))                                  VPRI0280
      DO 16 IR1=1,NR                                                    VPRI0290
      I1=IR1*(IR1-1)/2                                                  VPRI0300
      DO 17 IR2=1,IR1                                                   VPRI0310
      IND=IR2+I1                                                        VPRI0320
      DO 15 I=1,NR                                                      VPRI0330
   15 IDP(I)=0                                                          VPRI0340
      ISEL=0                                                            VPRI0350
      DO 18 I=1,NR                                                      VPRI0360
      IR3=I                                                             VPRI0370
      IF (LARK.EQ.1.AND.I.GT.NR) IR3=IR3+NLARK                          VPRI0380
      IDPI=IVCCB(IND,IR3)                                               VPRI0390
      IF (IDPI.EQ.0) GO TO 18                                           VPRI0400
      IDP(I)=1                                                          VPRI0410
      IF (IDPI.LT.0) IDP(I)=-1                                          VPRI0420
      ISEL=1                                                            VPRI0430
      IF (LARK.EQ.0.OR.NDEGA(I).LE.NDMAX) GO TO 18                      VPRI0440
      IDPI=IVCCB(IND,IR3+NLARK)                                         VPRI0450
      IF (IDPI.NE.0) IDP(I)=2                                           VPRI0460
      IF (IDPI.LT.0) IDP(I)=-2                                          VPRI0470
   18 CONTINUE                                                          VPRI0480
      IF (ISEL.EQ.0) GO TO 17                                           VPRI0490
      WRITE (IOUT,19) IR1,IRTYP(IR1),IRPAR(IR1),IR2,IRTYP(IR2),IRPAR(IR2VPRI0500
     1),(IDP(I),I=1,NR)                                                 VPRI0510
   19 FORMAT (/1X,I2,1H(,A4,A3,4H) X ,I2,1H(,A4,A3,1H),2X,I4,13I7)      VPRI0520
   17 CONTINUE                                                          VPRI0530
   16 CONTINUE                                                          VPRI0540
      WRITE (IOUT,1)                                                    VPRI0550
    1 FORMAT (1H1,'VECTOR COUPLING COEFFICIENTS'/1X,28(1H-)//)          VPRI0560
      MARK=0                                                            VPRI0570
      DO 7 I3=1,NR                                                      VPRI0580
```



```
      IR3=I3                                                            VPRI0590
      ND3=NDEGA(I3)                                                     VPRI0600
    8 IF (MARK.EQ.1) IR3=IR3+2                                          VPRI0610
      IF (MARK.EQ.1.AND.NDMAX.EQ.2) IR3=IR3-1                           VPRI0620
      DO 2 I1=1,NR                                                      VPRI0630
      ND1=NDEGA(I1)                                                     VPRI0640
      DO 2 I2=1,I1                                                      VPRI0650
      ND2=NDEGA(I2)                                                     VPRI0660
      ND12=ND1*ND2                                                      VPRI0670
      IND=I2+I1*(I1-1)/2                                                VPRI0680
      IVCB=IABS(IVCCB(IND,IR3))                                         VPRI0690
      IF (IVCB.EQ.0) GO TO 2                                            VPRI0700
      WRITE (IOUT,3) I1,I2,I3                                           VPRI0710
    3 FORMAT (//' REPRESENTATIONS ',I2,' X ',I2,' --> ',I2/)            VPRI0720
      WRITE (IOUT,5) ((IANGA(I,I3),I=1,ND3)                             VPRI0730
    5 FORMAT (28X,'IANG3 -->'//4X,'IANG1  IANG2',14X,I2,4(18X,I2))      VPRI0740
      WRITE (IOUT,25)                                                   VPRI0750
   25 FORMAT (1X)                                                       VPRI0760
      IVSUB=0                                                           VPRI0770
      IFV=IFVCC(IVCB)                                                   VPRI0780
      DO 4 I=1,ND1                                                      VPRI0790
      IANG1=IANGA(I,I1)                                                 VPRI0800
      DO 4 J=1,ND2                                                      VPRI0810
      IANG2=IANGA(J,I2)                                                 VPRI0820
      INIT=IFV+IVSUB                                                    VPRI0830
      IND=INIT+(ND3-1)*ND12                                             VPRI0840
      WRITE (IOUT,6) IANG1,IANG2,(VCCOEF(K),K=INIT,IND,ND12)            VPRI0850
    6 FORMAT (5X,I2,5X,I2,4X,5F20.9)                                    VPRI0860
    4 IVSUB=IVSUB+1                                                     VPRI0870
    2 CONTINUE                                                          VPRI0880
      IF (MARK.EQ.0.AND.ND3.GT.NDMAX) GO TO 9                           VPRI0890
      MARK=0                                                            VPRI0900
      GO TO 7                                                           VPRI0910
    9 MARK=1                                                            VPRI0920
      GO TO 8                                                           VPRI0930
    7 CONTINUE                                                          VPRI0940
      RETURN                                                            VPRI0950
      END                                                               VPRI0960
```

PROGRAM: GPTHEORY ROUTINE: VIBROT PAGE 75

```
      SUBROUTINE VIBROT                                                 VIBR0010
      RETURN                                                            VIBR0020
      END                                                               VIBR0030
```

PROGRAM: GPTHEORY ROUTINE: APPLY PAGE 76

```
      SUBROUTINE APPLY                                                  APPL0010
      RETURN                                                            APPL0020
      END                                                               APPL0030
```

PROGRAM: GPTHEORY ROUTINE: FILOUT PAGE 77

```
      SUBROUTINE FILOUT                                                 FIL00010
C**** DUMP OUTPUT COMMON BLOCK INFORMATION ONTO TAPE OR DISK            FIL00020
      IMPLICIT REAL*8 (A-H,O-Z)                                         FIL00030
      COMMON/OUT1/DUM01(391)                                            FIL00040
      COMMON/OUT8/DUM08(2046)                                           FIL00050
      COMMON/OUT2/DUM02(7092)                                           FIL00060
      COMMON/OUT3/DUM03(81),IDUM03                                      FIL00070
      COMMON/OUT4/DUM04(2970)                                           FIL00075
      LDUMP=20                                                          FIL00080
      WRITE(LDUMP)DUM01                                                 FIL00090
      WRITE(LDUMP)DUM08                                                 FIL00100
      WRITE(LDUMP)DUM02                                                 FIL00110
      WRITE(LDUMP)DUM03,IDUM03                                          FIL00120
      WRITE(LDUMP)DUM04                                                 FIL00130
      RETURN                                                            FIL00140
      END                                                               FIL00150
```

ACKNOWLEDGMENTS

We are grateful to the Chemistry Division and the Applied Mathematics Division for their support and cooperation in the implementation of this method. We thank Dr. A. C. Wahl for helpful comments and suggestions, and are particularly indebted to Dr. A. L. H. Chung, whose concurrent work at this laboratory had considerable influence on the development of this work. The algorithms for "unscrambling" sets of vectors having the same symmetry and for their phase alignment are due to her.

REFERENCES

1. E. P. Wigner, *Group Theory and Its Applications to the Quantum Mecahnics of Atomic Spectra*, Academic Press, New York (1959).
2. H. Bethe, Ann. Physik [5] *3*, 133 (1929).
3. R. S. Mulliken, Phys. Rev. *40*, 55; *41*, 49, 751 (1932); *43*, 279 (1933).
4. E. U. Condon and G. H. Shortley, *The Theory of Atomic Spectra*, Cambridge University Press (1935).
5. C. J. Ballhausen, *Introduction to Ligand Field Theory*, McGraw-Hill Book Co., New York (1962).
6. C. C. J. Roothaan, Rev. Modern Phys. *23*, 69 (1951).
7. G. Racah, Phys. Rev. *62*, 438 (1942); *63*, 367 (1943); *76*, 1352 (1949).
8. A. R. Edmonds, *Angular Momentum in Quantum Mechanics*, Princeton University Press (1957).
9. Y. Tanabe and S. Sugano, J. Phys. Soc. Japan *9*, 753 (1954).
10. Y. Tanabe and H. Kamimura, J. Phys. Soc. Japan *13*, 394 (1958).
11. J. S. Griffith, *The Theory of Transition Metal Ions*, Cambridge University Press (1961).
12. J. S. Griffith, *The Irreducible Tensor Method for Molecular Symmetry Groups*, Prentice-Hall, Inc., Englewood Cliffs, N. J. (1962).
13. B. D. Joshi, J. Chem. Phys. *43*, S40 (1965).
14. A. C. Wahl, J. Chem. Phys. *41*, 2600 (1964).
15. A. D. McLean and M. Yoshimine, IBM J. Res. Devel. *12*, 206 (1968).
16. E. Clementi and D. R. Davis, J. Comp. Phys. *2*, 223 (1967).
17. I. G. Csizmadia, M. C. Harrison, J. W. Moscowitz, and B. T. Sutcliffe, Theoret. Chim. Acta *6*, 191 (1966).
18. Z. Gershgorn and I. Shavitt, Int. J. Quantum Chem. *1S*, 403 (1967).
19. S. H. Glarum, private communication.
20. P. Pulay, Gy. Borossay, and F. Török, J. Mol. Struct. *2*, 336 (1968), and references therein.
21. S. Flodmark and E. Blokker, Int. J. Quantum Chem. *1S*, 703 (1967).
22. J. R. Gabriel, J. Math. Phys. *5*, 494 (1964); *9*, 973 (1968).
23. J. R. Gabriel, J. Math. Phys. *10*, 1932 (1969).
24. R. Moccia, Theoret. Chem. Acta *7*, 85 (1967).
25. R. McWeeny, *Symmetry - An Introduction to Group Theory*, Pergamon Press, Oxford, p. 195 (1963).
26. A. L. H. Chung and G. L. Goodman, to be published.
27. T. D. Bouman, A. L. H. Chung, and G. L. Goodman, in *Sigma Molecular Orbital Theory*, edited by O. Sinanoglu and K. Wiberg, Yale University Press, New Haven, pp. 333-349 (1970).

28. M. Gussoni and G. Zerbi, J. Mol. Spectry. *26*, 185 (1968).

29. S. L. Altmann, Proc. Cambridge Phil. Soc. *53*, 343 (1957).

30. S. L. Altmann and C. J. Bradley, Proc. Roy. Soc. *A255*, 199 (1962).

ARGONNE NATIONAL LAB WEST
3 4444 00011254 0

X

B