

AD-A037 797

ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF PHYSICS  
THE ELECTRONIC STRUCTURE OF CRYSTALLINE SOLIDS: THEIR SURFACES --ETC(U)  
FEB 77 A KUNZ

F/G 20/11

F33615-72-C-1506

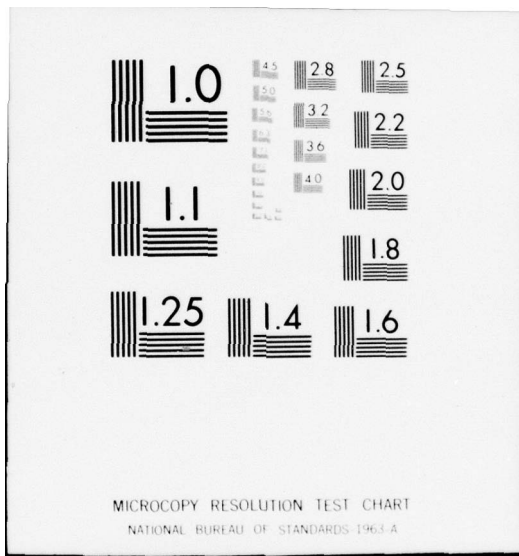
UNCLASSIFIED

AFAL-TR-76-140

NL

1 of 2  
ADA037797





ADA 037797

AFAL-TR-76-140

Handwritten scribbles and the number 12 circled.



# THE ELECTRONIC STRUCTURE OF CRYSTALLINE SOLIDS: THEIR SURFACES AND INTERFACES

DEPARTMENT OF PHYSICS AND MATERIALS RESEARCH LABORATORY  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  
URBANA, ILLINOIS 61801

FEBRUARY 1977

TECHNICAL REPORT AFAL-TR-76-140  
FINAL REPORT FOR PERIOD JANUARY 1975 - DECEMBER 1975

Approved for public release; distribution unlimited

DDC  
RECEIVED  
APR 5 1977  
A

AD NO. \_\_\_\_\_  
DDC FILE COPY

AIR FORCE AVIONICS LABORATORY  
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

DONALD C. REYNOLDS  
Project Scientist

FOR THE COMMANDER

ROBERT D. LARSON, Chief  
Electronic Research Branch  
Electronic Technology Div.  
AF Avionics Laboratory

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDP	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION.....	
BY.....	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and, or SPECIAL
A	

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
18 1. REPORT NUMBER AFAL-TR-76-140	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9	
6 4. TITLE (and Subtitle) The Electronic Structure of Crystalline Solids: Their Surfaces and Interfaces		5. TYPE OF REPORT & PERIOD COVERED Final Report, 1 Jan 1972 - 31 March 1976	
		6. PERFORMING ORG. REPORT NUMBER	
10 7. AUTHOR(s) A. Barry/Kunz		8. CONTRACT OR GRANT NUMBER(s) F33615-72-C-1506	
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 7885407-62	12 97
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory (AFAL/DHR) AFSC Aeronautical Systems Division Wright-Patterson AFB, Ohio 45433		12. REPORT DATE February 1977	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 163p.		13. NUMBER OF PAGES	
		15. SECURITY CLASS. (of this report) Unclassified	
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
Hartree-Fock		Semiconductors	
Self-consistent field method		Metals	
Charge densities		Oxides	
Compton Profiles			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)			
<p>→ This project was designed to study accurate, non-empirical theoretical techniques which permit one to have a predictive theoretical capability for systems of interest in electronic devices of a solid state nature. Thus, the classes of materials of interest include insulators (principally oxides), semiconductors, and some metals. Properties of bulk crystalline solids are needed as are properties of surfaces and interfaces. <del>It</del> desired to develop a unified technique for all classes of systems.</p> <p style="text-align: right;">→ It is cont.</p>			

176015 AB

cont

~~We conclude~~ the most general available technique is the Hartree-Fock Self-Consistent Field Method. In this case for most, if not all solids, this is best implemented by means of a local-orbital transformation. Since the Hartree-Fock method is a mean-field theory, inclusion of correlation corrections is important for studies of ground state properties (charge densities, Compton profiles, equilibrium lattice spacings). ~~We conclude~~ the best method overall in terms of economy and accuracy is the electronic-polaron model in its improved form. ~~We also conclude that~~ similar techniques are viable for surface state and interface state studies.

## FOREWORD

Contract Number F33615-72-C-1506, project AFAL-TR-76-140, The Electronic Structure of Crystalline Solids, performed by A. Barry Kunz, Associate Professor Physics, The University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801. This Contract monitored by 1st. Lt. Ronald Greene, AFAL/NP, Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio. Research performed during the period 1 January 1972 - 31 March 1976, by A. Barry Kunz, and submitted July 1976.

The research performed during this contract period was done with the assistance of Dr. T. C. Collins, Dr. R. N. Euwema, Dr. D. R. Jennison, and Dr. R. J. Blint. This is a final report.

## ABSTRACT

This project was designed to study accurate, non-empirical theoretical techniques which permit one to have a predictive theoretical capability for systems of interest in electronic devices of a solid state nature. Thus the classes of materials of interest include insulators (principally oxides), semiconductors, and some metals. Properties of bulk crystalline solids are needed as are properties of surfaces and interfaces. We desired to develop a unified technique for all classes of systems.

We conclude the most general available technique is the Hartree-Fock Self-Consistent Field method. In this case for most, if not all solids, this is best implemented by means of a local-orbital transformation. Since the Hartree-Fock method is a mean-field theory, inclusion of correlation corrections is important for studies of excitation properties and densities of states but less important for studies of ground state properties (charge densities, Compton profiles, equilibrium lattice spacings). We conclude the best method overall in terms of economy and accuracy is the electronic-polaron model in its improved form. We also conclude that similar techniques are viable for surface state and interface state studies.

## TABLE OF CONTENTS

SECTION	PAGE
I INTRODUCTION .....	1
II GENERAL EXPERIMENTAL CONSIDERATIONS .....	2
III GENERAL THEORETICAL CONSIDERATIONS .....	4
IV DEVELOPMENT OF SOLVABLE MODELS .....	6
1. Correlation Corrections .....	6
2. Relaxation Models .....	8
3. Electron-Hole Interaction .....	11
V ATOMIC ILLUSTRATION .....	13
VI SOLID-STATE SYSTEMS - an illustration .....	14
VII EXTENSION TO NARROW GAP SOLIDS AND METALS .....	23
VIII LOCAL ORBITAL COMPUTER PROGRAM .....	24
1. Use of the Code .....	24
2. Input Data Defined .....	25
3. List of Programs .....	30
IX BAND THEORY PROGRAM .....	54
1. Use of the Code .....	54
2. Input Data Defined .....	55
3. List of Programs .....	59
X ELECTRONIC POLARON PROGRAM .....	120
1. Use of the Code .....	120
2. Input Data Defined .....	121
3. List of Programs .....	122
APPENDICES	
A. APPLICATION OF THIS METHOD TO Ca SOLID .....	133
B. LIST OF STUDENTS AND POST DOCTORAL PERSONNEL SUPPORTED BY THIS CONTRACT.....	148

TABLE OF CONTENTS (continued)

APPENDICES	PAGE
C. LIST OF PAPERS SUPPORTED IN PART BY THIS CONTRACT .....	149
REFERENCES .....	152

LIST OF TABLES

TABLE	PAGE
1. A Summary of Calculations for the Atomic Li and Be Systems .....	15
2. A Summary of Calculations for Solids Ne and Ar .....	19
3. A Summary of Results for Solids LiF, LiCl .....	20
4. A Summary of Results for Solids NaF, NaCl .....	21
5. A Summary of Results for Solids, KF, KCl .....	22

## LIST OF SYMBOLS

$F$	Fock operator
$\rho$	charge density matrix
$\phi_i$	one electron orbital
$\epsilon_i$	one electron energy in Fock limit
$d\tau$	element of volume
$e_i$	exact one electron energy
$\Sigma_i^*$	"self energy"
$V_{\vec{q}}^{ab}$	matrix element coupling states a, b separated by $\vec{q}$ in reciprocal space
$h$	Planck's constant/ $2\pi$
$W$	exciton excitation energy
$\vec{k}, \vec{q}, \vec{k}$	wave vectors
$V$	crystal volume
$\epsilon_\infty$	optical dielectric constant
$\theta$	virtual space density operator
$A$	arbitrary Hermetian operator
$U$	environmental potential
$E$	Total System energy
$\psi$	many body wave function
$H$	many body hamiltonian
$\gamma$	second order density matrix
$K_{i\alpha}$	electron-hole interaction strength
$E_B$	exciton binding energy
$h$	Planck's constant
$m$	electron mass
$e$	electron charge
$\Lambda(E)$	density of states

## I. INTRODUCTION

At the start of this project, the available techniques for determining the electronic structure of crystalline solids were essentially limited to empirical techniques or highly approximate methods employing ill-defined approximations. These techniques were very limiting in scope in that to be able to use the very accurate empirical techniques one needed to have a great deal of spectroscopic data available for the system of interest. This restriction clearly limited the predictive ability of such theories. On the other hand the approximate techniques based upon local electronic density approximate operators were incapable of systematic improvement since the derivations were ill defined. This means that when such theories work, all is well, but when they fail there is no real remedy.

There was a third alternative available and already in extensive use in atomic and molecular physics. This method is that of Hartree-Fock. Preliminary results by Lipari and Kunz had demonstrated that Hartree-Fock studies were possible for systems such as solid rare gases and alkali-halides. They further demonstrated that the inadequacies of the Hartree-Fock method were capable of systematic improvement by way of including electron screening effects. In general the work of Lipari and Kunz was a substantial improvement on the local-density methods (now renamed the  $\alpha$  method by some users). This improvement is illustrated here for a typical case, KCl. Two quantities of extreme interest in any system are the magnitude of the band gap and the width of the valence band. In the case of KCl the experimental band gap is 8.4 eV and the valence width is 3.0 eV. The most accurate available  $\alpha$  results which are due to DeCicco find a band gap of 6.2 eV and a valence width of 0.67 eV. The Hartree-Fock case finds a band gap of 13.8 eV and a valence width of 3.4 eV. After correcting the Hartree-Fock case for correlation effects, we find a band gap of 9.2 eV and a valence width of 2.9 eV. In both cases the error is less than 10% for the corrected Hartree-Fock case and as much as 500% for the  $\alpha$  system. In addition we find the corrected Hartree-Fock method accurately reproduces trends from system to system. Typically then these Hartree-Fock corrected results are in fact more accurate than those quoted for KCl.

## II. GENERAL EXPERIMENTAL CONSIDERATIONS

The study of optical absorption edges in insulators has consumed a considerable amount of effort in the past 10 years. By this statement, the author includes soft or even hard x-ray edges in the class of optical absorption edges. Of course, in the recent few years most of the effort has been concentrated on the soft x-ray region of the spectrum, a region which, until the advent of convenient synchrotron radiation sources, was most difficult to study.

Most of the experimental work has concentrated upon ordinary optical absorption experiments. Especially in the soft x-ray region there has been an absence of any experiments using modulation spectroscopy techniques or two quantum absorption studies, which permit one to identify to some extent the symmetry and the character of the transitions in question. Thus, for example, in the soft x-ray region it has not been possible to unequivocally establish the presence or absence of excitonic transitions such as dominate the optical absorption spectrum of an insulator in the region of fundamental optical absorption.

Another class of interesting and useful experiments which are performed are ESCA (Electron Spectroscopy for Chemical Analysis) or the x-ray photo emission (XPE) experiments. In this type of experiment the sample absorbs an energetic photon and the kinetic energy distribution of the emitted electrons is analyzed. In such an experiment when subject to proper analysis, the energy distribution of the initial states may be deduced and also sometimes that of the conduction states, provided they overlie the ionization continuum. The essential differences between an ESCA or an XPE measurement involves the type of photon source and is of no concern for the present time.

Consider the following example. The crystal in question is LiF and we may wish to know experimentally the energy separation between say the LiK-shell (1s shell) and the Fluorine  $L_{II,III}$ -shell (2p shell or the valence band). One could imagine measuring this in two or three ways. In the first of them, which has been performed in part by Stephan<sup>1/</sup> and by Sonntag,<sup>2/</sup> one measures the onset of optical absorption from the F  $L_{II,III}$ -shell and also from the LiK-shell. One might then interpret the difference in these energies as being the energy difference between the K-shell and the  $L_{II,III}$ -shell. In the present case, the K-shell absorption begins at about 54 eV while the  $L_{II,III}$ -shell absorption begins at about 12 eV yielding a difference of 42 eV for the splitting between the K-shell and the  $L_{II,III}$ -shell. The alternate is to study the XPE spectra of these shells as has recently been done by Gudat, et al.<sup>3/</sup> In this case, one finds that the splitting of the K-shell and the  $L_{II,III}$ -shell is 50 eV. Clearly there is some difference here. In addition, one might also do a direct measurement by partly depopulating the K-shell and then measuring the energy of photons emitted as electrons from the  $L_{II,III}$ -shell fall into the K-shell. This measurement would in principle yield a still different result. The difference between these results is not, in the author's opinion, due to any lack of experimental precision, but is a direct measure of the strength of electron-hole or excitonic interaction in these crystals.

Here it is known from valence band excitations that the strength of the excitonic effect (binding energy) in this region of the spectrum is about .5 to 2 eV. This shall be shown to mean, in effect, that in the soft x-ray region of the spectrum, the strength of such exciton effects is about 10 eV or hence about the first 10 eV of the soft x-ray spectrum must be due to either localized atomic-like excitations or excitons.

The theoretical situation has been much more fragmentary. It was first observed by Brown et al. that the shapes of the soft x-ray absorption spectra bore great resemblance, in many but not all cases to the density of states of the conduction band.<sup>4/</sup> This idea has been extended by the present author for the alkali-halides,<sup>5/</sup> by Rössler<sup>6/</sup> for the rare gas solids,<sup>6/</sup> and by the author and Mickish for these cases.<sup>7/</sup> Most recently, Menzel et al. have advanced the idea of band structure for LiF being entirely responsible for the optical properties (in both fundamental and LiK-shell region).<sup>8/</sup> However, as we see from the preceding analysis of data for LiF, the theory of Menzel, et al. becomes impossible since if band theory were correct, the ESCA and absorption data would agree. Analysis of a type like that used in this article to sort exciton from band effects using experimental data rather than theory was done for RbCl by Scheifley and Brown.<sup>4/</sup>

Alternate studies of soft x-ray spectra in insulators, and particularly those intent upon predicting absorption peak positions have been either of a deep impurity type<sup>9-13/</sup> or of the more usual excitonic type.<sup>13,14/</sup> In such calculations or theories, the shape of absorption spectra is usually secondary to the question of line positions. In addition, there have been some exciton theories based upon critical point type analysis advanced by Onodera and Toyozawa<sup>15/</sup> in which splittings of exciton lines formed from band states at critical points are studied for both electrostatic and spin-orbit interactions. It shall be seen that for such crystals as considered in the present case the strength of the electron-hole interaction and the localization of the low lying exciton excited states are such that critical point analysis becomes pointless in many instances.

Finally, in this run-down of pertinent preliminary results, the author refers to the results of the calculations using the operator OAO of Collins and Kunz.<sup>16/</sup> In this, the strength of the coulomb-hole interaction was studied for atomic excitations. It was found, in general, that these effects were substantial. It was also found by a comparison of these results to experiment that simple atomic relaxation or polarization effects were exceptionally large and, in fact for many core excitations, these relaxations far exceeded the strength of the electron-hole interaction for such states. It was also seen then that a simple dielectric theory added to the electron-hole interaction in order to describe the polarization relaxation would involve a dielectric constant less than unity if such a description were to succeed. No one seems anxious to use such a dielectric constant. This indicates that simple, effective mass type theories for core exciton formation is pointless in most, or at least many, cases.

In the present report, in Section III we consider general theoretical considerations. The author examines this both from the standpoint of ordinary energy band theory and also from the viewpoint of a Heitler-London theory. In Section IV, the author takes the general considerations of Section III and reduces them to solvable models which are used in the present

system of attack. In Section V, the author provides a series of atomic system calculations which illustrate the theoretical features of the present ideas. In Section V, a number of solid state calculations are performed in which a quantitative value is placed on a variety of pertinent effects as well as experimental quantities. Throughout this report, the author neglects lattice relaxation or lattice dynamical effects. It is not, however, to be inferred from this neglect that the author considers such effects unimportant, but rather they serve simply to modify the dominant electronic effects discussed in this paper by amounts which are comparable in size to the errors introduced by the approximation to the larger electronic terms.

### III. GENERAL THEORETICAL CONSIDERATIONS

In this section the author constructs a brief theoretical review from rather general points of view. On the whole such viewpoints are not to be regarded as new but are, nonetheless, not so completely assimilated by the scientist as to make such a review superfluous. In the later stages of this work it shall be advantageous to adopt different viewpoints for purposes of economy in computing different contributions. Therefore, it is of some use here to study the question of band theory and exciton formation from both the molecular orbital (Bloch orbital) and from the Heitler-London (or general valence bond) point of view. There are any number of places where the information conveyed here is available in greater detail. Perhaps the most accessible is that of Knox.

In our method we view the solid as being described in the first approximation by a single Slater determinant of one-electron orbitals. If these orbitals are suitably chosen via the variational theorem, subject to constraints of orthonormality, one may obtain the one electron orbitals as being given by the canonical unrestricted Hartree-Fock formalism. In this case, the one electron eigenstates are symmetry adapted (i.e. they have the periodicity of the lattice) and generally each one spans all the infinite periodic lattice. It turns out that these orbitals are a convenient set for describing properties such as conductivity but much less convenient for describing localized phenomena. If the orbitals are chosen such that,

$$F(\rho) \phi_i(x) = \epsilon_i \phi_i(x), \quad (1)$$

where  $F(\rho)$  is the Fock operator for the system then one has

$$\int \phi_i^*(\vec{x}) \phi_j(x) d\tau = \delta_{ij}. \quad (2)$$

Now one allows  $\vec{x}$  to stand for both spatial dependence  $\vec{r}$  and spin dependence  $s$  of the orbitals and  $\int d\tau$  includes summation over spin coordinates. One recognizes that some solutions to Eq. (1) define orbitals which have electrons in them (occupied orbitals) in the ground state as well as unoccupied states (virtual orbitals). I assume here  $N$  electrons in the system and use the notations  $i, j \leq N$  with the  $i, j$  labeling occupied orbitals and the  $a, b$  being virtual orbitals.

In terms of this notation the first order density matrix is defined as

$$\rho(\vec{x}, \vec{x}') = \sum_{i=1}^N \phi_i(\vec{x}) \phi_i^+(\vec{x}'). \quad (3)$$

From this one has some immediate physical consequences which are summed up in Koopmans' theorem. These are, if all the orbitals  $\phi_k$  and  $\phi_c$  are chosen from a self consistent solution to Eq. (1) for the ground state then in this frozen orbital limit the eigenvalues  $\epsilon_k$  represent the energy needed to remove the electron in orbital  $k$  from the system. That is, one goes from an  $N$  to an  $N-1$  body system. The eigenvalue  $\epsilon_a$  represents the energy needed to add an electron to the system, putting it<sup>a</sup> in an orbital  $a$ . Thus one goes from an  $N$  to an  $N+1$  body system.

In much usual solid state physics, one also identifies the energy difference  $\epsilon_a - \epsilon_i$  as being the amount of energy needed to excite an electron from state  $i$  to state  $a$ . This identification is only justified if the orbitals span all of an infinite periodic lattice whereby the effect of changing the number of particles on the potential is total negligible. The above<sup>18/</sup> descriptions constitute what is generally regarded as Koopman's theorem. The consequence of this is to say that the usual Hartree-Fock virtual orbitals do not see any electron-hole interaction. It has been recently shown that the absence of electron-hole interaction in a simple Hartree-Fock system such as given by Eq. (1) is a consequence of an arbitrary choice for the virtual manifold rather than as a consequence of Hartree-Fock theory and there is a proper choice of  $F$  which permits one to obtain virtual states which see electron-hole interactions, possesses a real Koopmans' theorem for<sup>19/</sup> excitation even for localized states, and satisfy a variational principle.

One recognizes that the Hartree-Fock system so far described, possesses several defects even if electron-hole interaction is properly included. These defects are, of course, when one uses Koopmans' theorem, one neglects the relaxation of the other orbitals and hence the associated energy changes of the rest of the system. This correction is termed a relaxation correction and stems solely from not performing self-consistent calculations on both ground and excited state or alternately on both ground and ionized state. The second defect is that all dynamic correlations between electrons are neglected. The correlation and relaxation corrections are together taken to be the self-energy correction. It is normal for extended systems then to solve beyond Hartree-Fock by solving Dyson's equation, to which Hartree-Fock is the first approximation.<sup>20/</sup> Thus if  $e_i$  is the actual energy of the electron in state  $i$ , one has

$$e_i = \epsilon_i + \sum_i^* \quad (4)$$

$\sum_i^*$  being the self energy.

#### IV. DEVELOPMENT OF SOLVABLE MODELS

In this section, the author will derive a series of reasonably simple solvable models for the phenomena of optical absorption and for photo-emission experiments. In this case, development is made starting with both the band theoretical model (molecular orbitals) and the Heitler-London formalism (valence bond model). The author will, in the case of correlation, rely heavily on previous efforts.<sup>21/</sup>

##### IV-1. CORRELATION CORRECTIONS

One begins with the molecular orbital model and includes correlation as required. Traditionally this is done in two ways. The first way which is termed Screened-Exchange plus coulomb-hole was given useful information by Hedin<sup>22/</sup> and was used first on a calculation for Si.<sup>23/</sup> This model was later extended<sup>5,24-26/</sup> as an adjunct to a Hartree-Fock formalism by Lipari and the author.

The second model used is that of the electronic polaron (EPM) which was introduced for the case of conduction electrons by Toyozawa<sup>27/</sup> and was given a fundamental derivation and extended to the case of holes by the author.<sup>28/</sup> In this model, the correlation contribution to  $\sum^*$ , called  $\sum_{nc}^*(\vec{k})$  is found to be for conduction electrons

$$\sum_{ac}^*(\vec{k}) = \sum_{\vec{q}, b} \frac{|V_{\vec{q}}^{ab}|^2}{\epsilon_a(\vec{k}) - \hbar W_{ex} - \epsilon_b(\vec{k} - \vec{q})}, \quad (6)$$

and for holes

$$\sum_{nc}^*(\vec{k}) = \sum_{\vec{q}, m} \frac{|V_{\vec{q}}^{nm}|^2}{\epsilon_n(\vec{k}) + \hbar W_{ex} - \epsilon_m(\vec{k} - \vec{q})}, \quad (7)$$

where

$$V_{\vec{q}}^{nm} = e \left( \frac{2\pi\hbar W_{ex}(1-1/\epsilon_{\infty})}{V} \right)^{\frac{1}{2}} \frac{i}{|\vec{q}|} \int \phi_n^*(\vec{r}) \phi_m(\vec{r}) e^{i\vec{q} \cdot \vec{r}} d\vec{r}. \quad (8)$$

In this case the summation on  $\vec{q}$  is over the first Brillouin zone. For electrons the b summation extends over all conduction levels and for holes the m summation is over the occupied space. V is the crystal volume, e the electron charge,  $\epsilon_{\infty}$  the optical dielectric constant,  $\phi_i$  is the Wannier function for the i<sup>th</sup> band,  $\hbar W_{ex}$  is the quantum of the polarization field (in this instance, the longitudinal exciton field) and is assumed dispersionless. This model is valid in so far as the width of the band being correlated is less than  $\hbar W_{ex}$ . This method of correlation may be generalized and even extended to the case of metallic systems.<sup>29/</sup>

It is clear that it is also possible to develop a definition of the energy band model from a valence bond or Heitler-London point of view. To the best knowledge of the author, no such band calculations have been carried out at the present. It is noted that the structure of the energy band Hamiltonian as a function of wave vector here is similar to that for the molecular orbital case. Furthermore, in the unrelaxed orbital limit one would expect for insulators that the energy bands would be quantitatively as well as qualitatively similar to the molecular orbital case. Correlation corrections based upon the Heitler-London model are not fully developed. There is, however, an available limiting theory of such a model due to Fowler.<sup>26</sup> In this model (a semi-classical one), the electron charge is situated at either a cation or anion (or atomic) site and the atom or ion at this site is neglected. The remainder of the crystal is then treated as if it were consisting of a collection of non-overlapping polarizable ions (or atoms). The polarization of this assemblage is computed, as is the energy associated with this polarization. This energy is then identified with  $\sum_c$ . Clearly this model neglects  $k$  dependence. It does, however, distinguish between events on non-equivalent lattice sites. Most importantly, this model clearly and unequivocally neglects the effects of relaxation on the ion or atom at the origin, and, most importantly, this model is seen to be equivalent to the electronic polaron model if one takes a static limit of the electronic polaron (i.e., that limit in which the bands are dispersionless).

It is also necessary to make a brief comment about the agreement found among the various correlation models. At this point, there are a number of calculations of  $\sum_{ic}$ , using all three models for the same crystal. These models, firstly, are in excellent qualitative agreement. Secondly, for the number of crystals for which all three types of results are available and computed in a seemingly reliable manner, one finds that roughly  $\sum_{ic}$  serves to lower the band gaps by about  $4.0 \pm 1.0$  eV and that the agreement on this quantity among the models is of the order of 10%. For most practical purposes this level of agreement is adequate.

It is necessary at this point to say a word about the presumed local states which occur as virtual levels in the Heitler-London scheme. The virtual orbitals, if the  $\epsilon$  matrix for the virtual states is to be diagonal, are just the ordinary virtual Bloch functions. This is undesirable for a localized description and unnecessary. The virtual levels are arbitrary. Let one define

$$\hat{O} = 1 - \rho \quad (9)$$

$$A^{1-F} = \hat{O}\hat{A}\hat{O}, \quad (10)$$

then let one solve

$$\{ F + \rho\hat{U}\rho + \hat{O}\hat{A}\hat{O} \} \phi_{Ai} = \epsilon_{Ai} \phi_{Ai}. \quad (11)$$

Here both  $\hat{U}$  and  $\hat{A}$  are arbitrary Hermitian operators. However, if  $\hat{A}$  is chosen to be attractive and to interrupt the periodicity of the lattice, one can obtain localized virtual states from Eq. (11). Some particularly useful

choices for  $\hat{A}$  based upon physical considerations, are given by Kunz and Collins.<sup>16,19</sup> One may consider Eq. (11) to be the generalized Hartree-Fock equation.

#### IV-2. RELAXATION MODELS

The basic problem of relaxation can be handled, in principle, from the viewpoint of the molecular orbital approach. In the case of atoms or molecules, it is standard to include relaxation corrections totally by the simple expedient of solving for the ground and the excited or ionized states of the system in fully self-consistent calculations. In such calculations then the differences in total energies include not only Koopmans' terms but also complete relaxation information. In addition, for the case of excitation such a calculation also includes the electron hole interaction in a manner which is correct, save for the inclusion of correlation corrections. In the cases of deep core states especially, correlation corrections have been seen to be minor compared to other effects and the answer based upon the total energy difference of self-consistent calculations has been capable of great precision.<sup>30</sup> In the case of the infinite solid with its infinite total energy such a simple and direct calculation has thus far been impractical. Of course, such difficulties as infinities may be avoided in the direct approach with molecular orbitals by using a perturbation expansion in terms of unrelaxed orbitals and thereby directly cancel the infinities prior to calculation. To date no one has actually performed such a calculation for any real solid.

A more simple and direct method for constructing the effects of relaxation is possible if one considers the Heitler-London model. In what follows, the author relaxes the restrictions that the orbitals used in the excited state or for the ionized states be obtained from solving self-consistently Eq. (11) for the orbitals. Instead the author assumes that Eq. (11) is solved self-consistently for the ground state, as is true for a number of solids now. One proceeds by permitting the orbitals in the wave functions given by Eq. (11) to be chosen variationally by minimizing the energy of the Hamiltonian with respect to these wave functions.

To do this, one proceeds as follows, which is very similar to the derivation of the Multi-Configurational Self-Consistent-Field Equations by T. L. Gilbert,<sup>31</sup> except that in the present case the coefficients of the single Slater determinant (called patterns by Gilbert) are constrained by symmetry considerations within a multiplicative factor of modulus unity. The total energy is represented as

$$E = \langle \psi | \hat{H} | \psi \rangle + \langle f\rho \rangle + \langle\langle g\rho \rangle\rangle \quad (12)$$

where, in keeping with the usual usage in this type of theory, one has

$$\rho = M \langle \psi \psi^+ \rangle_{M-1} \quad (13)$$

$$\gamma = \left(\frac{M}{2}\right) \langle \psi \psi^+ \rangle_{M-2} \quad (14)$$

There are  $M$  electrons and  $\rho$  and  $\gamma$  are the reduced first and second order density operators. In proceeding, we could follow Gilbert in not constraining the orbitals to be orthogonal, but this complexity is not needed for the present case. If orthogonality is desired the correct variational equations are simply

$$\delta \left[ E - \sum_{A_i, B_j=1}^N \langle A_i | B_j \rangle \lambda_{A_i B_j} \right] / \delta \phi_{A_i}^+ = 0, \quad (15)$$

or

$$\delta \left[ E - \sum_{A_i B_j=1}^N \langle A_i | B_j \rangle \lambda_{A_i B_j} \right] / \delta \phi_{A_i} = 0 \quad (16)$$

The evaluation of these variations yields the equations for the orbitals  $\phi_{A_i}$  which become

$$F \phi_{A_i} - \sum_{B_j=1}^M \phi_{B_j} \lambda_{B_j A_i} = 0, \quad (17)$$

and

$$F \phi_{A_i}^* - \sum_{B_j=1}^M \lambda_{B_j A_i} \phi_{B_j}^* = 0. \quad (18)$$

Here the orbital operator  $F$  is found to be

$$F = f_\rho + 2 \langle g\gamma \rangle, \quad (19)$$

with

$$f_\rho = \sum_i \int \rho(\vec{x}, \vec{x}'') f_i(\vec{x}'', \vec{x}') d\vec{x}'' \quad (20)$$

and

$$\langle g\gamma \rangle = \sum_{ij} \iiint g_{ij}(\vec{x}_1, \vec{x}_1', \vec{x}_2, \vec{x}_2') \gamma(\vec{x}_1, \vec{x}_2, \vec{x}_1', \vec{x}_2') d\vec{x} d\vec{x}_2' d\vec{x}_1'. \quad (21)$$

Finally, the author also specifies that here operators are treated as local kernels, that is for example,  $f_i(\vec{x}\vec{x}') = f_i \delta(\vec{x}-\vec{x}')$  and  $g_{ij}(\vec{x},\vec{x}',\vec{x}_1,\vec{x}_1') = \delta(\vec{x}-\vec{x}_1)\delta(\vec{x}'-\vec{x}_1')g_{ij}$ . Equations of this nature of complexity are routinely solved in calculations for simple atoms and molecules. To date, implementation of a scheme such as outlined above has not been feasible for the general solid state case.

What is very beneficial from the standpoint of our present considerations are the specific approximation schemes which one might devise from these considerations and it is for this reason we explicitly include the preceding discussion on Gilbert's work. Therefore, the models being employed are at least correct in some realistic limit, which, in part, justifies the inclusion of this material. This limit is one in which one assumes that the atoms or ions comprising the system overlap slightly or not at all. (Such a condition is often met in the alkali-halides or solid rare gases, in which the ground state local orbitals overlap by about 0.05 or less). In such a non-overlapping limit, the energy expectation value of the wave function (11) is about the same as for the symmetry-adapted wave function. Therefore, one need not prefer at this level one wave function from the other in calculating energy differences.

In this limit consider first the case of optical excitation. Here one has the atom at site A being excited and the other atoms remain in their ground state. The ground state of the other atoms need not be the same as when the system as a whole were in the ground state. However, when one excites the atom at site A this atom doesn't alter its state of charge, since we consider here formation of excitons, rather than excitations which ionize the atom in question. If the atom is ionized relaxation effects due to the surroundings will increase the values of the relaxation energies given in this paper. Therefore, in the small overlap region, one may anticipate that the polarization of the surrounding atom is weak compared to that of the atom at site A. Therefore, if all atoms other than that at site A are fixed in their system ground state configuration, that at site A must be solved for in the ground state and the excited state configuration self-consistently. The total energy difference then represents the excitation energy and, by construction, contains the following contributions, that due to Koopmans' theorem, Hartree-Fock relaxation and electron-hole, but no correlation, since this limit is clearly a single determinant. One may subtract the Hartree-Fock contribution and obtain the magnitude of the relaxation plus electron-hole interaction from relaxation. There are two ways to do this. The first is to solve for the excited state at A in the canonical Hartree-Fock limit and then in the OAO limit, choosing A to properly represent the hole formed.<sup>16,19</sup> This is a simple procedure to implement and has been done here extensively. From this evaluation of electron-hole interaction one then subtracts from the effects of relaxation plus electron-hole to obtain the effects of relaxation alone. The second procedure is to use the ionization processes from (11) and, in the vanishing overlap limit, calculate self-consistently the energy to ionize the atom at A self-consistently in the field of the fixed other atoms.<sup>31</sup> From a comparison of the Hartree-Fock eigenvalue to the total energy one has  $\sum_{Ri}^*$  directly. That is, one has

$$-\sum_{Ri}^* = E_{I(i)}^{(A)} - E_g^{(A)} + \epsilon_{Ai}. \quad (22)$$

This has been tested for the rare gas systems: He, Ne, Ar. It is concluded directly for the rare gas systems that the relaxation phenomena is strictly a local atomic effect. Due to their cost and complexity, similar calculations were not made for the alkali-halides. However, one may infer a similar result from available information. Although the alkali-halides overlap more strongly than the solid rare gases ( $S_{nn} < 0.05$ ) the overlap is still quite small compared to unity. It is also well known that the cohesive energy of the alkali halides is essentially accounted for by the Madelung energy term with the detailed electron charge effects being quite insignificant<sup>18</sup> by comparison. Since the Madelung contribution is similar in the relaxed and unrelaxed state and since the relaxed state is less diffuse than the unrelaxed state so that the detailed electronic structure is even less important, one argues that here too the greatest part of the relaxation energy is due to the central cell atom (or ion) relaxation. It is true in the ionic crystal case the wave function of the atoms around the central cell atoms are also distorted by having dipoles induced when the state of ionization of central cell atom changes. However such effects are in part at least included in either a Mott-Littleton<sup>26</sup> or electron-polaron<sup>27,28</sup> model. Finally it is also worth noting that even though the relaxation model is derived from a Heitler-London model and the correlation model from a Bloch picture that both effects as given here are distinct and contributions are not being counted twice.<sup>26,29</sup>

#### IV-3 ELECTRON-HOLE INTERACTION

The highly approximate version of the self-consistent Heitler-London model as described in Section 3.2 provides one with electron-hole interaction strengths. In fact, in the limit described which is appropriate for either an "impurity" state, or a description akin to such a state, or for the more normal symmetry-adapted state, one finds the amount by which the electron-hole paired state (called hereafter an exciton whether it be an "impurity state" or a true crystal stationary state) lies below the continuum or band-like state. This energy difference can be expressed in terms of the OAO formalism as follows. Let  $i$  refer to the hole and  $a$  to the electron. Let  $\epsilon_a^i$  be the OAO energy of electron  $a$  in the presence of hole  $i$ , and let  $\epsilon_a$  be the canonical Hartree-Fock energy of orbital  $a$ . The electron hole interaction strength is then found to be

$$K_{ia} = \epsilon_a^i - \epsilon_a. \quad (23)$$

In the simplest limit one would predict that the greatest amount by which the exciton line would lie below the band state is just  $K_{ia}$ . This limit clearly neglects any dispersion in the energy levels. One might, in principle, solve for the exciton state in order to correct this deficiency. Such a calculation is not currently practical, although very desirable. There are other limits which permit one to take into account the band shape.

The most simple model is the effective mass model. In this limit, the exciton's electron is assumed to be in a very diffuse orbit so that the effective electron-hole interaction looks like that of a point charge. This is a hydrogenic model. Here one finds  $E_B$ , the binding energy below a band extremum, is

$$E_B = \frac{me^4}{2\epsilon^2} \frac{1}{h^2 n^2} \quad (24)$$

In Eq. (24),  $m$  is the electron-hole reduced mass,  $\epsilon$  is an appropriate dielectric constant ( $> 1$ ) and  $n$  the exciton principal quantum number. In the ionic system or the solid rare gases the radius of the principal exciton ( $n=1$ ) is too small for the assumption leading to (24) to be valid and the author never uses the effective mass theory for computing  $n=1$  exciton states in this paper. However, the higher states are often much more diffuse and (24) becomes tenable. One result is to use Eq. (24) to assign a limit to the position of  $n=2$  levels (s or p character) as being simply 25% of the binding energy for the  $n=1$  level in the effective mass limit. Thus in the calculations, one may assign the lower limit (the state would be raised in energy by dielectric screening far more than it is lowered by an effective mass for the electron less than an electron mass) to the  $n=2$  line as being  $0.25 K_{ij}$  as given in Eq. (23).

In these crystals, the effective mass idea is quite simple to apply to the case of core state excitations. The effective mass of the core holes is infinite (hence isotropic). The conduction band minima are  $\Gamma_1$  states and hence also perfectly isotropic. The author notes that the effective masses actually calculated in several different directions in  $\vec{k}$  space for the energy bands used here were found to be isotropic as the theory requires. In this article the effective mass approximation is only applied to formation of  $n=2, \ell=1$  excitons (p-excitons) obtained by exciting core electrons. It is believed this is reasonable here in that the p-excitons are far more diffuse in  $\vec{r}$  space than are the  $n=1, \ell=0$ , (s-excitons) which we compute in the one-band one-site model discussed below, and hence satisfy more nearly the criterion for validity of effective mass theory. It is also noted that due to the essential nature of the one-band one-site model, only a single bound state is found for any of the cases considered here so that effective mass ideas are the only simple way to construct p-excitons.

There is a second limit which is appropriate for positioning the  $n=1$  exciton line in many crystals for which Eq. (24) is inappropriate and which is always employed here. This is the tight-bound limit or Slater-Koster-One Band-One-Site model.<sup>33/</sup> This is a deep impurity model limit in which the crystal band structure is correctly taken into account. In this model, one assumes that the excited electron is attracted to the hole by a coulomb interaction and that the hole is local to an ion or atom site and that the wave function for the electron may be expanded in terms of the <sup>34/</sup>one-electron band orbitals. It is assumed the interaction is short range<sup>34/</sup> and only orbitals of a single band are used in the actual expansion. One further assumes from the band function one constructs the appropriate Wannier functions on a given site. It is assumed then that the matrix elements of the electron-

hole interaction in the Wannier representation are given by

$$\langle \vec{R}_\mu, n | V | \vec{R}_\nu, m \rangle = V_0 \delta_{\mu\nu} \delta_{nc} \delta_{mc} \delta_{\mu 0} \quad (25)$$

$V_0$  is the effective interaction and may be given by  $R_{ia}$  in an  $\hat{O}\hat{A}\hat{O}$  calculation. Here  $|R_\mu, m\rangle$  is the  $m^{\text{th}}$  Wannier function at site  $\vec{R}_\mu$  and site  $\vec{R}_0$  is the site of the "impurity". The quantum number  $c$  stands for the conduction band for which the sum is over. In this representation the probability that the electron is in the cell with the hole is given by  $\frac{35}{}$

$$|\psi_E(0)|^2 = |1 - V_0 \Lambda(E)|^{-2}, \quad (26)$$

$$\Lambda(E) \equiv \lim_{\delta \rightarrow 0^+} \int_{-\infty}^{\infty} dt N(t) / (E - t + i\delta), \quad (27)$$

In (27),  $N(t)$  is the state density for the bands defined by  $|\vec{R}_\nu, c\rangle$ .

One sees at once that in the non-overlapping limit the quantity  $V_0$  in (25) and (26) is identified with  $K_{ia}$  given in (23) for our model. Thus one defines the needed parameters. The author assumes that continuing to identify  $K_{ia}$  with  $V_0$  in the small overlap limit introduces negligible error in the cases on hand.

## V. ATOMIC ILLUSTRATION

The theory outlined in the previous two sections makes a number of definite predictions. For example, removing an electron from state 1 requires energy in the amount  $(-e_1)$ ,

$$e_1 = \epsilon_1 + \sum_1^* \quad (28)$$

The energy needed to excite an electron from a state 1 to a state 3 becomes

$$E_{13} = \epsilon_3 + \sum_3^* - \epsilon_1 - \sum_1^* + K_{13} \quad (29)$$

The expression (28) is pertinent for studies of ESCA or photoemission spectroscopy whereas it is (29) which is pertinent for optical absorption or emission. From that it is possible to see one other effect of interest. Suppose one is interested in the separation of two occupied levels say (1) and (2). This could be determined indirectly from either ESCA or optical absorption. For  $\Delta E_{12}$  by ESCA one has

$$\Delta E_{12} = \epsilon_2 + \sum_2^* - \epsilon_1 - \sum_1^* \quad (30)$$

whereas by optical absorption one has

$$\Delta E_{12} = \epsilon_2 + \sum_2^* - \epsilon_1 - \sum_1^* + K_{13} - K_{23}. \quad (31)$$

Furthermore, if direct optical absorption is possible from the level 1 to 2 (e.g. for a Li atom  $1s$  to  $2s$  is possible), one has

$$\Delta E_{12} = \epsilon_2 + \sum_2^* - \epsilon_1 - \sum_1^* + K_{12}. \quad (32)$$

In general there is no reason to expect these three expressions for  $\Delta E_{12}$  to agree. It is noted however, that if one uses Eq.(31) to determine the Rydberg series limit, then the values for  $\Delta E_{12}$  from Eq.(31) will agree with those from Eq.(30) since the K's vanish in this limit. However, they will not agree with the results from Eq.(32) in general.

The author suggests that a criterion such as this could serve as an experimental distinction between exciton levels and band levels. Such a criterion is often needed in the soft x-ray range since the excitons (or states analogous to deep impurities which we don't differentiate) are resonant states and hence such traditional tests as photoconductivity are not useful for making distinctions.

To test these ideas in a simple framework, a few simple atomic calculations were performed. These were done for the Li atom and the Be atom. In this case, the energies of the various states of single ionization were computed as were the energies of the lowest excited state for each occupied level and, for Li, the energy of the transition  $1S^2 2S \rightarrow 1S 2S^2$ . All calculations were performed in the unrestricted Hartree-Fock limit. A common basis set for all Li cases is used and likewise for Be. In all cases a basis of 10 gaussian orbitals is used. In Li several possible spin states are sometimes given. From these self-consistent total energies the author calculates  $\Delta E_{1S 2S}$  for these systems as pertinent for the types of experiment outlined above. These results are summarized in Table I. The disagreement among the various models to define  $\Delta E_{1S 2S}$  is of the order of 1 to 4 eV, an amount which is significant. In the next section, this study is extended to solid systems and the author shows how in solid state cases this disagreement may be greatly enhanced.

## VI. SOLID STATE SYSTEMS

In the event of excitation of solid state systems the author predicts that there should be two distinct types of behavior. The first of these is analogous to the atomic system discussed in the preceding section. In

Table I

A summary of calculations for the atomic Li and Be systems. The results of several calculations are given here for total energies in several ground and excited states. All calculations are by means of the unrestricted Hartree-Fock formalism. The energy differences between the 1s and 2s level as deduced by optical absorption, photoemission and for Li direct transition from 1s to 2s level are also given. Results are in Rydbergs.

	<u>State</u>	<u>Energy</u>
Li 1s <sup>2</sup> 2s	<sup>2</sup> S	14.8609
Li 1s <sup>2</sup>	<sup>1</sup> S	14.4710
Li 1s2s	<sup>3</sup> S	10.1008
Li 1s2s	<sup>1</sup> S	10.0417
Li 1s <sup>2</sup> 3s	<sup>2</sup> S	14.7499
Li 1s2s3s	<sup>2</sup> S	10.2129
Li 1s2s <sup>2</sup>	<sup>2</sup> S	10.6140
Be 1s <sup>2</sup> 2s	<sup>1</sup> S	29.1356
Be 1s <sup>2</sup> 2s	<sup>2</sup> S	28.5454
Be 1s2s <sup>2</sup>	<sup>2</sup> S	20.0570
Be 1s <sup>2</sup> 2s3s	<sup>1</sup> S	28.6988
Be 1s2s <sup>2</sup> 3s	<sup>1</sup> S	20.2651
Li ΔE(1s2s) for photoemission		4.3702, 4.4293
Li ΔE(1s2s) from optical absorption to 3s level		4.5370
Li ΔE(1s2s) for direct absorption		4.2469
Be ΔE(1s2s) for photoemission		8.4884
Be ΔE(1s2s) for optical absorption		8.4437

this type of system one expects the separation of core levels to be experiment dependent, but by about 1.0 eV or so. Such systems would be typified by the solid rare gases and the author expects will include most if not all non-ionic systems, that is systems in which exciton formation has the electron and hole on the same site. The second type of system would be the ionic crystals which are typified by the alkali-halides. In such systems one will have markedly different behavior depending upon whether it is the anion or cation which is excited.

Consider the limit of vanishing overlap: if the ion being excited is an anion (here a negative ion) it will generally not possess any bound states other than the ground state. Therefore the electron migrates from the anion and may be trapped on the surrounding cations. If this occurs due to the charged lattice there still exists a net coulomb attraction between the excited electron and the now neutral anion. In this case, the coulomb hole term is roughly but reasonably well described by the expression

$$K_{ij} = \frac{-e^2}{|\vec{R}_a - \vec{R}_c| \epsilon} \quad (33)$$

In Eq. (33)  $|\vec{R}_a - \vec{R}_c|$  is the average separation of anion and cation and  $\epsilon$  is put to include the effect of screening which reduces the size of the attraction. In the present model calculations, where pertinent, the author uses the optical of electron constant which serves to maximize  $K_{ij}$ . In general there is additional screening due to ionic motion which would enhance  $\epsilon$  above the optical limit. Thus, in general, the value given by the author for excitation of anions will be an upper bound on the exciton binding (lower bound on the exciton position).<sup>36/</sup>

In exciting the cation, things are much different. The cation itself has a coulomb attraction for the excited electron and hence an infinite number of bound states on the same ion as the hole. In this case the value of  $K_{ij}$  is given as described in Section IV-3. Due to these physically different mechanisms for ionic crystals the discrepancies in energy between core levels as determined by differing processes can vary substantially and are found to differ by as much as 10 eV or so. The author would anticipate that such effects would be present in polar molecules to a degree similar to the ionic crystals. In the remainder of this section, the author describes the results of specific calculations for the solid state systems, Ne, Ar, LiF, NaF, KF, LiCl, NaCl, and KCl. In these calculations the author will ignore the small differences between levels as typified by the atomic calculations in Section V. The reader should be aware of their presence. In the methods used here such splittings for these solids were of the order of 0.25 eV. He concentrates instead on the larger effect of charge transfer versus non-charge transfer and in calculating the absolute position of true band absorption edges, lower bounds for exciton absorption for the band edges, in some cases actual positions of exciton lines using the theory of Section IV-3, and the positions for ESCA type edges.

The actual models used here are the following: The bands are computed first in the restricted Hartree-Fock limit. For the substances here these calculations are available and hence no new calculations of this type are performed. The correlation part of  $\sum^*$  is evaluated by the EPM except for NaCl and KCl, for which the available coulomb hole plus screened exchange results are used and for occasional core levels where Fowler's results are used.<sup>26/</sup> The relaxation part of  $\sum^*$  is evaluated using the Heitler-London approximation given in Section IV-2. The  $K_{ij}$ 's come from

the  $\hat{O}\hat{A}\hat{O}$  of Section IV-3, except for the charge transfer states, for which Eq. (47) is used. In using the Lipari and/or Kunz<sup>5/</sup> results for LiCl, NaCl, and KCl the valence bands are recomputed in position to correct for poor convergence of the plane wave basis in the original calculation. It is noted in passing, such corrections aren't needed for LiBr and NaBr of Lipari and Kunz<sup>5/</sup> as these are better converged.

Finally, the author will discuss the practical aspects of applying the  $\hat{O}\hat{A}\hat{O}$  model to compute the electron hole interaction strength. In doing this, the author briefly summarizes a series of theorems proven in Ref. 19 by Kunz and Collins. According to Ref. 19, the proper choice of the operator  $\hat{O}\hat{A}\hat{O}$  is given by

$$\hat{O}\hat{A}\hat{O} = \hat{O} \left[ (-1) \langle n | g_{12} (1 - P_{12}) | n \rangle \right] \hat{O}, \quad (34)$$

where one defines  $P_{12}$  by

$$\langle a | \hat{O} \langle n | g_{12} P_{12} | n \rangle \hat{O} | a \rangle \equiv \langle a n | g_{12} | n a \rangle. \quad (35)$$

In Eq. (35)  $g_{12}$  is the two body operator ( $e^2 / |\vec{r}_{12}|$ ) and  $|n\rangle$  is the wave function for the hole vacated in the band of principal quantum number  $n$ . Ideally then, in terms of the Bloch functions  $\psi_{nk}$  for this band, one has

$$|n\rangle = \int_{B.Z.} a_{nk} \psi_{nk}(\vec{r}) d\vec{k}. \quad (36)$$

The coefficients  $a_{nk}$ , may be chosen such that the expectation value of the virtual orbital  $\phi_{Aa}^{nk}$  solving Eq. (11) is minimized. This prescription is reasonable, since the choice of virtual orbital in Eq. (11) has been shown to be variationally determined, and to possess a Koopmans' theorem for excitations of the  $N$  body system.<sup>16,19/</sup> Therefore the shape of hole which optimizes the energy of  $\phi_{Aa}$  also optimizes the excited state total energy. This of course neglects relaxation of core orbitals shape and correlation. These effects here are treated separately.

In practice, for the case of non-overlapping orbitals (i.e. the orbitals for core states in this calculation), the proper choice of  $|n\rangle$  for use in Eq. (11) is found to be the local orbital,  $\phi_{An}$  for the hole vacated. For simplicity, the author also uses this approximately for the case of valence excitations. Here, of course, overlap considerations dictate that this choice of  $|n\rangle$  is not optimal. However, here the overlaps are small (of the order of 0.05 or less) and this should not be a significant problem. The calculations are performed using the same code as is used in solving for the local orbitals, for the ground state. Two shells of neighboring atoms are included in the calculation and the remainder of the lattice is treated by a point ion model. This seems reasonable, in that the first excited levels for the cations or rare gas atoms lie mostly inside the first neighbor distance when  $\hat{O}\hat{A}\hat{O}$  as described here is used. The value of  $K_{na}$  is then obtained by solving (11) with  $A$  in  $\hat{O}\hat{A}\hat{O}$  set to zero and

by taking the difference of the expectation value for  $\epsilon_{Aa}$  in these two limits.

In practice one could also obtain the value of  $K_{na}$  for the anion excitations by this model. However the diffuseness of the first excited level requires that one use a more varied angular basis set than the author is able to provide at this time in order to have an accurate estimate of the energy. Therefore it is simple and about as accurate in this case to use Eq. (33) to determine  $K_{na}$ . In this case  $(\vec{R}_A - \vec{R}_C)$  is the nearest neighbor distance. The author has tested this approximation on several molecules for which he is able to provide accurate calculations of charge transfer type excitations or core states. He finds the use of (33) introduces an error of about 20% in the value of  $K_{na}$ . In the test cases the value of  $K_{ia}$  given by (33) was too large and thus in no way compromises the conclusion of this manuscript.

The basis set for the solid state calculations consisted of s, p, and d type Slater orbitals. The same basis set used for the local orbitals calculations on the ground state was employed and was augmented by four additional diffuse s-type orbitals for describing the excited state (note symmetry prevents mixing of p or d orbitals in the excited state). The orbitals were even tempered in these calculations. In sum then for all 1s excitons the one-band one-site model is used and for all 2p excitons the effective mass theory. For excitations of cation or rare gas atom the OAO theory is used to compute  $K_{ia}$  and Eq.(33) is used for anion excitation.

Band structures for substances Ne, Ar, LiF, NaF, KF, LiCl, NaCl, KCl are shown in Figs. 1-4. These bands are obtained by starting with the self-consistent Hartree-Fock results in the literature. To these results  $\sum^*$  is added.  $\sum^*$  here includes both correlation and relaxation. This had not been reported except for LiF. In addition the bands for NaCl and KCl are corrected for the convergence of the plane wave basis. (This is needed only for the 3p valence bands.) This correction was computed by the author by computing the  $\Gamma$  point in a self-consistent LCAO model.

In addition to this, the values deduced by actual calculation for the values of the constituents of  $\sum^*$  and for the  $K_{ij}$ 's for the various crystals studied are given in Tables II-V. From these tables one can deduce the great difference between the K's for on-site excitation and for charge transfer excitation. In addition, one can see quite quickly for deep excitations the great dominance of the relaxation terms in determining the edge positions.

TABLE II

For the rare gas systems Ne and Ar the various contributions to the energy gaps are given. The correlation contribution is symbolized by  $\sum^*(c)$  whereas the relaxation contribution is given as  $\sum^*(r)$ . The electron hole interaction is essentially constant for all pair excitations on the Ne or Ar atoms. For excitations to the 3s level we deduce the value of K to be  $3.45 \pm .1$  eV and for excitations to the Ar 4s level K is found to be 1.5 eV. Results in eV.

Level	$\sum^*(c)$	$\sum^*(r)$
1s Ne	1.08	22.89
2s Ne	1.08	2.86
2p Ne	1.08	2.95
Conduction band He	-1.85	- 0
3p Ar	1.58	1.29
Conduction band Ar	-1.67	- 0

TABLE III

The self energies and the electron-hole interactions are given for LiF and LiCl. Results are in eV.

Level	System	$\Sigma^*(c)$	$\Sigma^*(r)$	$K_{ia}$
Li 1s	LiF	2.92	1.52	-10.38
F 1s	LiF	1.79	24.75	- 3.74
F 2s	LiF	1.79	4.00	- 3.74
F 2p	LiF	1.79	3.56	- 3.74
Li 1s	LiCl	3.45	1.52	-10.38
Cl 1s	LiCl	2.42	33.44	- 2.04
Cl 2s	LiCl	2.42	11.51	- 2.04
Cl 2p	LiCl	2.42	12.54	- 2.04
Cl 3s	LiCl	2.42	1.99	- 2.04
Cl 3p	LiCl	2.42	1.48	- 2.04
c.b.	LiF	-2.38	-1.0	
c.b.	LiCl	-1.61	---	

TABLE IV

The Self-Energies and the electron-hole interactions are given for NaF and NaCl. Results are in eV.

Level	System	$\Sigma^*(c)$	$\Sigma^*(r)$	$K_{ia}$
Na 1s	NaF	2.13	22.70	-8.60
Na 2s	NaF	2.13	2.79	-8.60
Na 2p	NaF	2.13	3.10	-8.60
F 1s	NaF	1.99	24.75	-3.59
F 2s	NaF	1.99	4.00	-3.59
F 2p	NaF	1.99	3.56	-3.59
Na 1s	NaCl	2.43	22.70	-8.60
Na 2s	NaCl	2.43	2.79	-8.60
Na 2p	NaCl	2.43	3.10	-8.60
Cl 1s	NaCl	1.31	33.41	-2.28
Cl 2s	NaCl	1.31	11.51	-2.28
Cl 2p	NaCl	1.31	12.54	-2.28
Cl 3s	NaCl	1.31	1.99	-2.28
Cl 3p	NaCl	1.31	1.48	-2.28
c.b.	NaF	-1.11	----	
c.b.	NaCl	-2.16	----	

TABLE V

The self energies and the electron-hole interaction strength are given for KF and KCl. Results are in eV.

Level	Substance	$\Sigma^*$ (c)	$\Sigma^*$ (r)	$K_{ia}$
1s K	KF	1.75	32.14	-7.22
2s K	KF	1.75	9.98	-7.15
2p K	KF	1.75	11.21	-7.13
3s K	KF	1.75	1.36	-7.00
3p K	KF	1.75	1.20	-6.84
1s F	KF	1.93	24.75	-2.93
2s F	KF	1.93	4.00	-2.93
2p F	KF	1.93	3.56	-2.93
1s K	KCl	1.96	32.14	-7.22
2s K	KCl	1.96	9.98	-7.15
2p K	KCl	1.96	11.21	-7.13
3s K	KCl	1.96	1.36	-7.00
3p K	KCl	1.96	1.20	-6.84
1s Cl	KCl	1.06	33.14	-2.16
2s Cl	KCl	1.06	11.51	-2.16
2p Cl	KCl	1.06	12.54	-2.16
3s Cl	KCl	1.06	1.99	-2.16
3p Cl	KCl	1.06	1.48	-2.16
c.b.	KF	-1.14	----	
c.b.	KCl	-1.99	----	

## VII. EXTENSION TO NARROW GAP SOLIDS AND METALS

All the preceding theory applies with only trivial modifications to systems such as narrow gap solids like Si or to metals. The method of local orbitals used for getting localized descriptions of the charge density operator may be modified in three equally useful ways. The first is to construct atom based local orbitals in a RHF mode by averaging over the needed low lying excited configurations. For example, in the case of Si, in which the Si atom exists for the atomic configuration  $1s^2 2s^2 2p^6 3s 3p^3$  ( $^5S$ ) one includes the virtual  $3s$  and  $3p$  in localization and projects out the occupied space using the band theory code to perform the separation. The second possibility is to localize some electron on atoms and some in bonds. In the case of Si one would then have a Si atom with electron in a  $1s^2, 2s^2, 2p^6$  configuration and 4 doubly occupied  $sp^3$  covalent bond orbitals. The third is to use a UHF description in which the up spin electrons localize on the A sublattice from the  $n=3$  shell and the down spin ones are the B sublattice. There is a fourth possibility and this is to form non-local orbitals which localize say on a  $Si_5$  cluster. This latter prospect is not appealing in that it is both inefficient of time and cumbersome to implement.

The band theoretical techniques used herein go over unchanged for metals or semiconductors .

The electronic polaron needs only minor modification to be generally useful. This modification is to use for a metal the plasma frequency rather than the exciton frequency and to recognize that for wide bands the expressions (6), (7) are  $\infty$  for many cases and one solves this by expanding the bordered determinant in the electronic polaron derivative exactly rather than by use of perturbation theory. In this limit the exact answer is given as

$$\sum_{ac}^* (\vec{k}) = \sum_{\vec{q}, b} \frac{|V_{\vec{q}}^{ab}|^2}{\epsilon_a(\vec{k}) - \epsilon_b(\vec{k}-\vec{q}) - \hbar\omega_{ex} + \sum_{ac}^* (\vec{k})} \quad (37)$$

and

$$\sum_{nc}^* (\vec{k}) = \sum_{\vec{q}, m} \frac{|V_{\vec{q}}^{nm}|^2}{\epsilon_n(\vec{k}) + \hbar\omega_{ex} - \epsilon_m(\vec{k}-\vec{q}) + \sum_{nc}^* (\vec{k})} . \quad (38)$$

The defining  $V_{\vec{q}}^{ij}$  are given by (8).

A reprint detailing the extension in practice for a metal is given as Appendix 1.

## VIII. LOCAL ORBITAL COMPUTER PROGRAM

This is a series of four programs used successively to solve for a solid state self consistent first order density matrix. The use of these programs and the listing are given here.

### VIII-1. USE OF THE CODE

This set of calculations uses four programs. The first is LØPAS. It solves self consistently the equation

$$\left[ F_A + U_A - \epsilon_{Ai} \right] \phi_{Ai} = \left[ - \sum_j \sum_k \phi_{Aj} \langle Aj | A | Ak \rangle \phi_{Ak}^+ \right] \phi_{Ai}$$

for a given  $U_A$ .  $A$  is a square well of radius 5.0 Bohr and depth - 1.0 Ry. The remaining three programs form  $U_A$  for LØPAS. For a diatomic substance the first pass of LØPAS solves for both types of atoms in the solid. This program produces two disk or tape output files. These are KØUT on device 6 and KLØUT on device 7. It may also need two input files. This is a data option as shown below. These files are KLRP on device 8 and KSRP on device 9. In addition there is card data as given below.

The third program used is called WFN. This produces potentials for the fourth program. The potential is produced in file KINTER on device 5 and is

$$V(r) = \sum_j \int \frac{e^2 \phi_j^2(\underline{r}_2)}{|\underline{r}_{12}|} d\underline{r}_2 .$$

This program needs card input and a file input on device 7, KLØUT produced by LØPAS. Card input is specified below.

The second program is PIØN, it generates the point ion potential (spherical part) about both a positive and negative ion site for a fcc lattice it requires card input as specified below and produces the output file KLRP on device 8 which is used by LØPAS.

The fourth program is DØUB, it takes the potential generated by WFN and expands it about other sites. As this deck is used for cubic solids we retain only the  $Y_0^0$  term of the expansion, the  $Y_1^m - Y_3^m$  are identically zero. The next term  $Y_4^{m0}$  is small by our tests and is neglected. This forms its output on file 9 and is KSRP. It uses file KINTER produced by WFN on device 5

and also card input. The potential generated here is only for the first two neighbors (i.e. the alkali and halogen are expanded about the halogen). We assume

$$\gamma(\mathbf{r}) = \sum_{\ell, m} a_{\ell m}(\mathbf{r}) Y_{\ell}^m(\theta, \phi)$$

$$a_{\ell m}(\mathbf{r}) = \int V(r-R_N) Y_{\ell}^m(\theta, \phi) d\Omega$$

In file KSRP is stored  $Y_{\ell}^m(\theta, \phi) a_{\ell m}(\mathbf{r})$ . This program is quite flexible and can generate other expansion members if desired as well as do other double integrals.

The standard way of using this program is to use LOPAS to solve for the free ion solutions or the solutions assuming  $U_A = V_{PI}(r)$ . Then use PION, WFN, and D0UB to get  $U_A$  sense LOPAS using input from these programs and cycle through this until self consistency. This is normally only one or two passes.

#### VIII-2. INPUT DATA DEFINED

The following set of data for each ion or atom needed.

Card 1: ZLAB(I), I=1,20 20A4  
Holorith data to provide a label or output

Card 2: Z, RHO, H, NTYPES, TOL, FRACN, MAXIT, KD1, KD2, 3F10.4, I4, F10.4, I10, 3I2, 6I2, 2F4.0, IATOM, NEF, NEG, INTER1, INTER2, INTER3, WT1, WT2

Z = AT number

we generate Rmesh from  $R(I) = \exp(\text{RHO} + (I-1)*H)/Z$

RHO is usually -3.0

doub assumes these values

H is usually 0.0625

NTYPES = no. of different ang. momenta present, i.e. for I=3  
if=0 calls exit. for  $\text{Na}^+ = 2$

TOL.=self consistency cryterion try 0.005, this is adequate usually

FRACN determines wave function for I+1st iteration from - INTEGER TRIGGERS PRATT IMPROVEMENT. USE 9999 to avoid

$$\psi^{I+1} = \text{FRACN} \psi^I + (1-\text{FRACN}) \psi^{I-1}$$

use 0.5 to 0.8. (If it works 1.0 is best) now it seems to work much of the time.

MAXIT = max. no. of interactions allowed if it is not specified program uses  
MAXIT = 21 and FRACN = 0.5

KD1 if = 0 gets estimate of trial wavefunctions from card data otherwise  
from file KOUT, almost always use 0

KD2 if 0 gets input data for potential from cards if not 0 gets data from  
files KSRP and KLFP use 0 for first pass of LOPAS and otherwise for  
subsequent runs for a given material.

IATOM  $\neq$  0 if we compute on atom

NEF = no. of extra  $F^k$  needed

NEG = no. of extra  $G^k$  needed

INTER1,2 and 3 for mixup of state

INTER 1 gives L values of functions interchanged as L+1

INTER 2 is one replaced by INTER 3 (allows one to get excited states for  
example)

WT1 is no. of atom in nearest neighbors,

WT2 is no. of atom in next nearest neighbors

Card 3: VEST(I), I=1,161 8F10.5  
this is the point ion potential.

Card 4: 1X,5F12.6 K(I) is short range potential.

Note: Read in card 3 and 4 only if KDZ  $\neq$  0 and then only from disk or tape.

Card 4\* : RMAX, DEPT, 2F10.4 Well part of rad RMAX and depth Dept with a  
tail  $R > RMAX$   $V = \frac{DEPT + RMAX}{R}$   
(Ignored in programs if cards 3 and 4 received in from disk or tape)

Card 5: 3I4  
NES = no of S functions expansion used  
NEP = no of P functions expansion used  
NED = no of D functions expansion used

Card 6: I4,F14.7 I=1, NTPES, J=1, NES, AJ(I,J), ZJ(I,J)  
NEP  
NED

wavefunction assumed to be of form

$$\psi_{n\ell m}(r) = Y_{\ell}^m(\theta, \phi) \sum_j C_{JNJ_j} r^{AJ_j + \ell} \exp(-ZJ_j R)$$

here the A+Z are readins. Program from the NJ's use A+Z from  
Freeman and Watson or Bagus papers usually or extrapolate

- Card 7:  $N\emptyset(I), I=1, 3, 3I4$   
 $N\emptyset(1)$  = no. of occ S orbitals (shells)  
 $N\emptyset(2)$  = no. of occ P orbitals (shells)  
 $N\emptyset(3)$  = no. of occ D orbitals (shells)  
i.e. for  $\text{Na}^+$   $N\emptyset(1)=2, N\emptyset(2)=1, N\emptyset(3)=0$
- Card 8: 6F12.5  $I=1, N\text{TYPES}, J=1, N\emptyset(I), K=1, 11 \text{ PSI}\emptyset(I, J, K)$ . Initial estimate of CJ coefficients can be rather bad makes little difference see Watson and Freeman, etc. Always on cards but if  $KD1 \neq 0$  also gets them from disk. Uses disk version if readin
- Card 9: 2I4,  $N\text{F}(I), N\text{G}(I), I=1, N\text{TYPES}$   
 $N\text{F}$  is no of  $F_k$  integrals needed for each type of orbital (see Hartree's book for definition)  
 $N\text{G}$  is no of  $G_k$  integrals needed for each type of orbital (see Hartree)
- Card 10: I4, F12.6  $I=1, N\text{TYPES}, J=1, N\text{F}(I) F(I, J), X\text{F}(I, J)$ ,  
 $F$  tells which function in core is used in the  $F^0$  integral,  $X\text{F}$  tells its weight see Hartree. Functions are stored as follows. All S functions then all P functions, then all D functions stored from lowest eigenvalue up.
- Card 11: 2I4, F12.6  $I=1, N\text{TYPES}, J=1, N\text{G}(I)$   
 $G(I, J, 1), G(I, J, 2), X\text{G}(I, J)$   
 $G(I, J, 1)$  gives the  $K$  of  $G^k$  to evaluate  
 $G(I, J, 2)$  gives the function used  
 $X\text{G}(I, J)$  gives the weight  
see Hartree
- Note in list sample input for LCR is given output is essentially the energies and the CJ's found for S then P then D functions.
- Card 12: 3I4, F18.10 if needed  
 $I, J, K, \text{WT}$  for extra  $F^k$ 's  
 $K$  is the  $k$  of  $F^k$   
 $I$  and  $J$  tell which function  
 $\text{WT} = 1.0$
- Card 13:  $I, J, K$  as card 12 except if needed for  $G^k$ 's

Input Information for WFN - set for each ion or atom

- Card 1:  $Z, N, \text{F10.4}, 14$   
 $Z$  = at.no  
 $N$  = no of occ sub. shells  
If  $Z = 0$  call exit
- Card 2:  $N$  times F10.4  $\text{WT}$   
 $\text{WT}$  is occ. number of this shell  
also file  $\text{KLRP}$   
only useful output is on  $\text{KINTER}$

Data for PIØN

(1 set for each atom)

- Card 1: NREG 14  
no of regions needed, use 6 if 0 call exit
- Card 2: IPT,RHØ,H,Z,SIGN,I\$, \$F12.6  
IPT = 161  
RHØ,H as in LØPAS  
Z = at/ mp  
SIGN= - ionicity i.e. for Na<sup>+</sup> is -1, for Cl<sup>-</sup> is +1
- Card 3: BØUND,CØNS,CØFT is NREG Terms 3F12.4  
Band is R max of region  
in Region we have V = CØNS+CØFT/R  
use for fcc lattice the set

BØUND	CØNS	CØFT
a/2	-6.99/a	0.0
$\frac{\sqrt{2}}{2}$ a/2	17.01/a	- 12.0
$\frac{\sqrt{3}}{2}$ a/2	-17.01/a	12.0
$\frac{2}{\sqrt{5}}$ a/2	1.445/a	-4.0
$\frac{\sqrt{5}}{2}$ a/2	-10.55/a	8.0
$\infty$	0.0	-3.74

a is latt. param in Bohr units useful output is on file KLRP for zinc blende lattice

- Card 1: 4
- Card 2: 161 -3.0 0.0625 Z  $\pm$  Ionicity
- Card 3:  $\frac{\sqrt{3}}{4}$ a -7.5658/a 0.0  
 $\frac{\sqrt{2}}{2}$ a 10.88/a -8.0  
 $\frac{\sqrt{11}}{4}$ a -23.0/a 16.0  
 $\infty$  0.0 -3.11

Input Data for DØUB

also gets data from KINTER

- Card 1: NØFUN 12  
no of types of atoms, i.e. for alkali halide = 2
- Card 2: Z(I), I=1,NØFUN 2ØF6.0  
Z = at. no. of each type of atom
- Card 3: I,J,K,L,M,N,A 6(2,F12.6)  
here for an diatomic system use the following 4 cards only
- |   |   |   |   |   |   |                |
|---|---|---|---|---|---|----------------|
| 3 | 3 | 2 | 2 | 2 | 1 | a/2            |
| 3 | 3 | 1 | 2 | 2 | 1 | $\sqrt{2}$ a/2 |
| 4 | 4 | 1 | 2 | 2 | 1 | a/2            |
| 4 | 4 | 2 | 2 | 2 | 1 | $\sqrt{2}$ a/2 |

General Input Data for DØUB for HF routine (if used as an integration program rather than extension program)

- Card 1: NØFUN 12  
no of functions read in
- Card 2: Z(I), 2ØF6.0 I=1, NØFUN  
Z at m
- Card 3: I,J,K,L,M,N,A, 6I2,F12.6  
I = fun at cert  
J = oth "  
K = fun at oth cent  
L = 1 = 1.0  
2 = Sσ  
3 = Pσ  
4 = Pπ  
5 = dσ  
6 = dπ  
7 = dS  
M  
N  
A = separation of origins

## VIII-3. LIST OF PROGRAMS

```

12:22 APR 05, 1976 JL=3DF-H11
!JOB RRL2427,KUNZ,1
LISTCL

!ASSIGN M:BB,(FILE,PF1),(SAVE)
!FORTRAN LS,B0
C WRITTEN BY A R KUNZ DEPARTMENT OF PHYSICS UNIVERSITY OF ILLINOIS 0020
C NOVEMBER 1969 IN FORTRAN IV FOR THE IBM 360-75 0030
DIMENSION ZLAB(20)
COMMON A(11,11),B(11,11),RR(11),SNL(11,161),PSI(3,11,161),
IK(161),NO(3,11),AJ(3,11),ZJ(3,11),
2PSII(3,11,11),PSIO(3,11,11),E(3,11),
2 BV(3,11,11),PNE(3,11,11),NO(3),NEXP(3), 0060
JNF(3),F(3,25),G(3,25,2),XF(3,25),XG(3,25),NG(3) 0070
COMMON /SKRAP/ S(161),SN(161),VEXT(161),VPRI(161)
COMMON /DIA/PSKRAP(11,11) 0090
DOUBLE PRECISION A,B,RR
REAL NJ 0100
INTEGER AJ,F,G 0110
1 FORMAT (34#ILEGAL ORBITALS PROGRAM OF AR KUNZ ) 0120
2 FORMAT (72#
1
3 FORMAT (3F10.4,14,F14.6,I10,9I2,2F4.0) 0140
4 FORMAT (3I4) 0160
5 FORMAT (14,F14.6) 0170
6 FORMAT (6F12.6) 0180
7 FORMAT (14,F12.6) 0190
8 FORMAT (2I4,F12.6) 0200
9 FORMAT (1#ENERGY IS,F14.6) 0210
10 FORMAT (1X,5F10.6) 0220
11 FORMAT (8F10.6) 0230
12 FORMAT (1X,5F12.6)
712 FORMAT (1H,19#CONVERGENCE TEST IS,2X,F10.4)
713 FORMAT (20A4)
714 FORMAT (1H,20A4)
900 FORMAT(1H,35# CONVERGED RESULTS FOLLOW
INPUT IYH
CALL CHKPT(1#LFF,ITYM)
30 CONTINUE
PRINT 1 0240
READ 713,(ZLAB(I),I=1,20)
PRINT 714,(ZLAB(I),I=1,20)
C OBTAIN DATA FOR FIRST ITERATION 0270
READ 3,Z,RHO,NTYPES,TEL,ITRIGG,MAXIT,KD1,KI2,IATUM,NEF,NEG,
IINTER1,INTER2,INTER3,WAT1,WAT2
IF (FRACN) 711,711,712
711 FRACN=0.5
702 CONTINUE
IF (MAXIT) 703,703,704
703 MAXIT=21
704 CONTINUE
IF (NTYPES.EQ.0) GO TO 804
DO 31 I=1,161,4
R(I)=EXP(RHO)/Z
31 RHO=RHO+4.*H
DO 401 I=1,157,4
DEL=(R(I+4)-R(I))/4.
R(I+1)=R(I)+DEL

```

```

R(I+2)=R(I)+DEL*DEL
401 R(I+3)=R(I)+3.*DEL
IF (KD2.EQ.0) GO TO 403
READ (8,11)(VEXT(I),I=1,161)
READ (9,12)(SN(I),I=1,161)
READ 3,RMAX,LEPT
DO 1102 I=1,161
VPR(I)=1.0
IF (R(I).E.0) 11 5,1102,1102
1100 VPR(I)=-1.0
1102 CONTINUE
DO 13 I=1,161
13 VEXT(I)=VEXT(I)+KAT1*SN(I)
DO 9999 I=1,161
9999 VPR(I)=VEXT(I)
GO TO 18 6
803 CONTINUE
READ 3,RMAX,LEPT
DO 18 5 I=1,161
IF (R(I)-RMAX) 8 6,806,807
806 VEXT(I)=DEPT
GO TO 18 5
807 VEXT(I)=DEPT+R(I)*Z/R(I)
1800 CONTINUE
DO 1103 I=1,161
1103 VPR(I)=-ABS(VEXT(I))
1800 CONTINUE
PRINT 12,(VEXT(I),I=1,161)
READ 4,NES,NEP,NEE 0340
DO 32 I=1,NES 0350
READ 5,AJ(1,I),ZJ(1,I) 0360
IS=2*AJ(1,I)+3 0370
ISS=IS
I=FACT0(IS) 0380
IS=ISS
32 NJ(1,I)=((2.*ZJ(1,I)**IS)/I)**0.5 0390
IF (NEP.EQ.0) GO TO 33 0400
DO 34 I=1,NEP 0410
READ 5,AJ(2,I),ZJ(2,I) 0420
IS=2+2*AJ(2,I)+3 0430
ISS=IS
I=FACT0(IS) 0440
IS=ISS
34 NJ(2,I)=((2.*ZJ(2,I)**IS)/I)**0.5 0450
33 IF (NEE.EQ.0) GO TO 35 0460
DO 35 I=1,NEE 0470
READ 5,AJ(3,I),ZJ(3,I) 0480
IS=4+2*AJ(3,I)+3 0490
ISS=IS
I=FACT0(IS) 0500
IS=ISS
35 NJ(3,I)=((2.*ZJ(3,I)**IS)/I)**0.5 0510
30 NEXP(1)=NES 0520
NEXP(2)=NEP 0530
NEXP(3)=NEE 0540
READ 4,N(1),N(2),N(3) 0550
NOC=N(1)+N(2)+N(3) 0560
DO 37 I=1,NTYPE 0570
N=N(1) 0580
DO 37 J=1,N 0590

```

37	READ 6,(PSIE(I,J,K),K=1,11)	0600
	IF (KD).EQ.0) GO TO 801	
	DO 802 I=1,NTYPES	
	N=NB(I)	
	DO 802 J=1,N	
802	READ (6,6)(PSIE(I,J,K),K=1,11)	
	REWIND 6	
801	CONTINUE	
C	INITIAL ESTIMATES ON HAND FILL IN DATA BLOCKS OBTAIN EQUATIONS	0610
	DO 38 I=1,NTYPES	0620
	N=NEXP(I)	0630
	DO 38 J=1,N	0640
	DO 38 K=1,161	0650
38	PSI(I,J,K)=NU(I,J)*R(K)**(I+AJ(I,J))+EZP(=ZJ(I,J)*R(K))	0660
35	FORMAT (1X,5F12.6)	
35	FORMAT (1X,1F8.4)	
	DO 41 I=1,NTYPES	0670
41	READ 4,NF(I),NG(I)	0680
	DO 42 I=1,NTYPES	0690
	N=NF(I)	0700
	DO 42 J=1,N	0710
42	READ 7,F(I,J),XF(I,J)	0720
	DO 43 I=1,NTYPES	0730
	N=NG(I)	0740
	DO 43 J=1,N	0750
43	READ 8,G(I,J,1),G(I,J,2),XG(I,J)	0760
C	ALL NECESSARY DATA ON HAND PROCEED TO FORM CONSTANT DATA BLOCKS	0770
	DO 44 I=1,NTYPES	0780
	N=NEXP(I)	0790
	DO 44 J=1,N	0800
	DO 44 K=J,N	0810
	CALL KU(J,K),ANS,NU,AJ,ZJ,R,Z)	0820
	UVE(I,J,K)=ANS	0830
	ONE(I,K,J)=ANS	0840
	CALL OVE(J,K),ANS,NU,AJ,ZJ,R)	0850
	OV(I,J,K)=ANS	0860
44	OV(I,K,J)=ANS	0870
	NINT=0	0880
C	ITERATION STARTS HERE	0890
45	CONTINUE	0900
	NINT=NINT+1	0910
	IF (NINT.EQ.NAXIT) GO TO 54	
C	FORM INITIAL SML	0930
	DO 40 I=1,11	0940
	DO 40 J=1,161	0950
40	SML(I,J)=.0	0960
	I=0	0970
	DO 39 J=1,NTYPES	0980
	M=NEXP(J)	0990
	N=NB(J)	1000
	DO 39 K=1,N	1010
	I=I+1	1020
	DO 39 L=1,N	
	DO 39 II=1,161	1040
39	SML(I,II)=SML(J,II)+PSIB(J,K,L)*PSI(J,L,II)	1050
	DO 101 I=1,NOG	1060
	ANS=0.0	1070
	DO 102 J=1,155,2	
102	ANS=ANS+(R(J+1)-F(J))/3.)*(SML(I,J)**2+4.*SML(I,J+1)**2+ SML(I,J+2)**2)	

	DO 101 J=1,N	1100
101	SNL(I,J)=SNL(I,J)/SQRT(ANS)	1110
	DO 46 I=1,NTYPES	1120
	N=NEXP(I)	1130
	DO 47 J=1,N	1140
	DO 47 K=J,N	1150
	A(J,K)=ONE(I,J,K)	1160
47	B(J,K)=B7(I,J,K)	1170
C	ONE BODY OPERATORS COMPLETE ADD IN F OPERATORS NEXT	1180
	DO 48 J=1,N	1190
	DO 48 K=J,N	1200
	NF0=NF(I)	1210
	DO 48 L=1,NF	1220
	LL=F(I,L)	1230
	CALL FK(R,SNL,PSI,0,LL,J,K,ANS,I)	1240
48	A(J,K)=XF(I,L)*ANS+A(J,K)	1250
C	ADD IN GK ELEMENTS	1260
	DO 49 J=1,N	1270
	DO 49 K=J,N	1280
	NG0=NG(I)	1290
	DO 49 L=1,NG0	1300
	LL=G(I,L,2)	1310
	KK=G(I,L,1)	1320
	CALL GK(R,SNL,PSI,KK,LL,J,K,ANS,I)	
49	A(J,K)=XG(I,L)*ANS+A(J,K)	1340
	IF (IATOM.GT.0) GO TO 1201	
	DO 501 J=1,N	
	DO 501 K=J,N	
	ANS=0.0	
	DO 10 (502,503,504),1	
502	NS=1	
	NE=ND(1)	
	DO 10 505	
503	NS=ND(1)+1	
	NE=ND(1)+ND(2)	
	DO 10 505	
504	NS=ND(1)+ND(2)+1	
	NE=ND(3)	
505	CONTINUE	
	DO 506 L=NS,NE	
	DO 506 LL=NS,LE	
	AN=0.0	
	CALL G0(SNL,PSI,0,LL,J,K,ANS,I)	
506	ANS=ANS+AN	
	ANS=-ANS	
501	A(J,K)=A(J,K)+ANS	
1201	CONTINUE	
	DO 50 J=1,N	1350
	DO 50 K=J,N	1360
	A(K,J)=A(J,K)	1370
50	B(K,J)=B(J,K)	1380
C	WE ARE READY TO OBTAIN EIGENVALUES AND EIGEN VECTORS	1390
	CALL DIAGN(R,I,N,RP)	1400
	M=NB(I)	1410
	DO 51 J=1,M	
	E(I,J)=RP(J)	1430
	DO 51 K=1,N	1440
51	PSI(I,J,K)=RP(K,J)	1450
45	CONTINUE	1460
	IF (INTER1.EQ.1) GO TO 1202	

	II=NEXP(INTER1)	
	DO 1203 I=1,II	
1203	PSII(INTER1,INTER2,I)=PSII(INTER1,INTER3,I)	
1202	CONTINUE	
C	POCK EQUATION SOLVED TEST FOR SELF CONSISTENCY	1470
	U=0.0	1480
	DO 52 I=1,NTYPES	1490
	N=NB(I)	1500
	M=NEXP(I)	1510
	DO 52 J=1,N	1520
	DO 52 K=1,M	1530
	DEL=ABS(PSII(I,J,K)-PSI(I,J,K))	1540
	IF (D=DEL) 53,52,52	1550
53	D=DEL	1560
52	CONTINUE	1570
	PRINT 712,U	
	IF (D=DEL) 54,54,55	1580
55	CALL EVEXTP(117166,NTYPES,NB,NEXP,PSII,PSI)	
	DO 805 I=1,NTYPES	
	N=NB(I)	
	DO 805 J=1,N	
805	WRITE (6,6)(PSI(I,J,K),K=1,11)	
	END FILE 6	
	REWIND 6	
	GO TO 45	1650
54	DO 57 I=1,NTYPES	1660
	PRINT 900	
	N=NB(I)	1670
	M=NEXP(I)	1680
	DO 57 J=1,N	1690
	PRINT 9,E(I,J)	1700
	IF (J.GT.N) GO TO 57	
	JJ=I-1	
	WRITE (7,4) M,JJ	
70	FORMAT (11I4)	
	WRITE (7,6) (PSI(I,J,K),K=1,M)	
	WRITE (7,701) (AU(I,K),K=1,M)	
	WRITE (7,6) (ZJ(I,K),K=1,M)	
57	PRINT 10,(PSI(I,J,K),K=1,M)	1710
	DO 601 I=1,NBC	
	DO 601 J=1,NBC	
	ANS=0.0	
	DO 602 K=1,1RS,2	
	D=(R(K+1)-R(K))/3.0	
602	ANS=ANS+D*(SNL(I,K)*SNL(J,K)*VPRI(K)+4.*SNL(I,K+1)	
	+SNL(J,K+1)*VPRI(K+1)+SNL(J,K+2)*SNL(I,K+2)*VPRI(K+2))	
601	PRINT 603,I,J,ANS	
603	FORMAT (1H0,2I4,F14.7)	
	II=0	
	DO 3901 I=1,NTYPES	
	DO 3901 J=1,NF(1)	
	II=II+1	
	IF (II.GT.11) GO TO 3900	
	DO 3902 K=1,161	
	SNL(II,K)=0.0	
	DO 3902 L=1,NEXP(I)	
3902	SNL(II,K)=SNL(II,K)+PSII(I,J,L)*PSI(I,L,K)	
3901	CONTINUE	
3900	CONTINUE	
	NSUM=NB(1)+NB(2)+NB(3)	

```

      II=0
      IF (NSUM.GT.11) GO TO 3903
      WRITE (1)(NSUM,NO,P)
      DO 3904 I=1,NTYPES
      DO 3904 J=1,NF(I)
      II=II+1
3904  WRITE (1)(SNL(II,K),K=1,161)
3905  CONTINUE
      CALL ZENERGY(Z,REF,NFG,NTYPES)
      GO TO 30
      1720
804  CONTINUE
      REWIND 8
      REWIND 9
      END FILE 7
      REWIND 7
      END FILE 1
      REWIND 1
      CALL EXIT
      67
      END
      1730
SUBROUTINE GHS(L,PSI,R,L,LL,J,K,ANS,N0)
DIMENSION SNL(11,161),PSI(3,11,161),R(161)
COMMON /SKRAB/ S(161),SN(161),VPRI(161),VEXT(161)
C
ADDS EXTERNAL EXCHANGE TERM TO MATRIX
A=0.0
B=0.0
C=0.0
DO 1 M=1,159,2
D=(R(M+1)-R(M))/2.0
A=A+D*(PSI(N0,J,M)*SNL(L,M)+4.*PSI(N0,J,M+1)*SNL(L,M+1)
+PSI(N0,J,M+2)*SNL(L,M+2))
C=C+D*(PSI(N0,K,M)*SNL(LL,M)+4.*PSI(N0,K,M+1)*SNL(LL,M+1)
+PSI(N0,K,M+2)*SNL(LL,M+2))
1  B=B+D*(SNL(L,M)*SNL(LL,M)*VEXT(M)+4.*SNL(L,M+1)*SNL(LL,M+1)
+VEXT(M+1)+SNL(L,M+2)*SNL(LL,M+2)*VEXT(M+2))
ANS=A*B*C
RETURN
END
FUNCTION FACTB(IS)
1740
IS=IS-1
1750
IF (IS) 1,1,2
1760
1  FACTB=1.0
1780
RETURN
1790
2  A=IS
1800
DO 3 I=1,100
1810
IS=IS-1
1820
IF (IS) 4,4,5
1830
3  B=IS
1840
3  A=A*B
1850
4  FACTB=A
1860
RETURN
1870
END
1880
FUNCTION EZP(X)
XLIM=-25.0
IF (X.LT.XLIM) GO TO 1
EZP=EXP(X)
RETURN
1  EZP=0.0
RETURN
END
SUBROUTINE BVE(L,P,N0,ANS,NJ,AJ,ZJ,R)

```

```

DIMENSION NU(3,11),AU(3,11),ZJ(3,11),R(161)
COMMON /SKRPF/ S(161),SN(161),VEXT(161),VPRI(161)
REAL NU
INTEGER AU
LE=NB+AU(NB,L)
ME=NB+AU(NB,M)
LT=LE+ME
A=FACT(LT)
B=(ZJ(NB,L)+ZJ(NB,M))**(LT+1)
ANS=NU(NB,L)*NU(NB,M)*A/B
RETURN
END
SUBROUTINE KU(M,L,NB,ANS,NU,AU,ZJ,R,Z)
DIMENSION NU(3,11),AU(3,11),ZJ(3,11),R(161)
REAL NU
INTEGER AU
COMMON /SKRPF/ S(161),SN(161),VEXT(161),VPRI(161)
LE=NB+AU(NB,L)
ME=NB+AU(NB,M)
C PE TERM FIRST
LT=LE+ME-1
A=FACT(LT)
B=(ZJ(NB,L)+ZJ(NB,M))**(LT+1)
C=-2.*Z*NU(NB,L)*NU(NB,M)*A/B
C ANG KE TERM NEXT
LT=LE+ME-2
A=FACT(LT)
B=(ZJ(NB,L)+ZJ(NB,M))**(LT+1)
D=NB*(NB-1)
E=NU(NB,L)*NU(NB,M)*D*A/B
C R KE TERM NEXT
D=-ME*(ME-1)
F=D*NU(NB,L)*NU(NB,M)*A/B
LT=LE+ME-1
A=FACT(LT)
B=(ZJ(NB,L)+ZJ(NB,M))**(LT+1)
D=ME
G=2.*D*ZJ(NB,M)*NU(NB,L)*NU(NB,M)*A/B
LT=LE+ME
A=FACT(LT)
B=(ZJ(NB,L)+ZJ(NB,M))**(LT+1)
H=-NU(NB,L)*NU(NB,M)*ZJ(NB,M)**2*A/B
ANS=C+E+F+G+H
LE=NB+AU(NB,L)+0.1
ME=NB+AU(NB,M)+0.1
DO 1 I=1,161
S(I)=NU(NB,L)*EZP(-ZJ(NB,L)*R(I))*R(I)**LE
1 SN(I)=NU(NB,M)*EZP(-ZJ(NB,M)*R(I))*R(I)**ME
AN=0.0
DO 2 I=1,159,2
A=S(I)*SN(I)*VEXT(I)
B=S(I+1)*SN(I+1)*VEXT(I+1)
C=S(I+2)*SN(I+2)*VEXT(I+2)
2 AN=AN+((R(I+1)-R(I))/3.)*(A+4.*B+C)
ANS=ANS+AN
RETURN
END
FUNCTION FACT(I)
J=I
N=J

```

```

IF (J.LE.0) GO TO 1
3 J=J-1
IF (J.EQ.0) GO TO 2
N=N*J
GO TO 3
2 FACT=N
RETURN
1 FACT=1.0
RETURN
END

SUBROUTINE INTF(S,R)
DIMENSION S(161),R(161)
DO 1 I=4,158,2
A=(R(I)-R(I-1))*(R(I)-R(I+1))*(R(I)-R(I+3))
B=(R(I)-R(I-3))*(R(I)-R(I+1))*(R(I)-R(I+3))
C=(R(I)-R(I-5))*(R(I)-R(I-1))*(R(I)-R(I+3))
D=(R(I)-R(I-7))*(R(I)-R(I-1))*(R(I)-R(I+1))
E=(R(I-3)-R(I-1))*(R(I-3)-R(I+1))*(R(I-3)-R(I+3))
F=(R(I-1)-R(I-3))*(R(I-1)-R(I+1))*(R(I-1)-R(I+3))
G=(R(I+1)-R(I-3))*(R(I+1)-R(I-1))*(R(I+1)-R(I+3))
H=(R(I+3)-R(I-1))*(R(I+3)-R(I+1))
1 S(I)=S(I-3)*A/E+S(I-1)*B/F+S(I+1)*C/G+S(I+3)*D/H
RETURN
END

SUBROUTINE FR(R,SNL,PSI,K,L,M,N,ANS,NB)
DIMENSION R(161),SNL(11,161),PSI(3,11,161)
COMMON /SKNAP/ S(161),SN(161),VEXT(161),VPR(161)
C COMPUTES FR INTEGRALS
ANS=0.0
DO 1 I=1,161
S(I)=0.0
1 SN(I)=1.0
DO 3 I=3,161,2
3 SN(I)=SN(I-2)+((R(I)-R(I-1))/3.)*(SNL(L,I)**2/R(I)**(K+1)
+4.*SNL(L,I-1)**2/R(I-1)**(K+1)+SNL(L,I-2)**2/R(I-2)**(K+1))
DO 4 I=3,161,2
4 S(I)=S(I-2)+((R(I)-R(I-1))/3.)*(SNL(L,I)**2*R(I)**K
+4.*SNL(L,I-1)**2*R(I-1)**K+SNL(L,I-2)**2*R(I-2)**K)
DO 5 I=1,161,2
5 S(I)=PSI(NB,M,I)*PSI(NB,N,I)*(S(I)/R(I)**(K+1)
+R(I)**K*(S(I-1)-SN(I)))
S(2)=0.5*(S(1)+S(3))
S(160)=0.5*(S(159)+S(161))
CALL INTF(S,R)
DO 6 I=1,159,2
6 ANS=ANS+((R(I+1)-R(I))/3.)*(S(I)+4.*S(I+1)+S(I+2))
ANS=2.*ANS
RETURN
END

SUBROUTINE GR(R,SNL,PSI,K,L,M,N,ANS,NB)
DIMENSION R(161),SNL(11,161),PSI(3,11,161)
COMMON /SKNAP/ S(161),SN(161),VEXT(161),VPR(161)
C COMPUTES GR INTEGRAL
ANS=0.0
DO 1 I=1,161
S(I)=0.0
1 SN(I)=1.0
DO 3 I=3,161,2
3 SN(I)=SN(I-2)+((R(I)-R(I-1))/3.)*(SNL(L,I)*PSI(NB,M,I)/R(I)**(K+1)
+SNL(L,I-1)*4.*PSI(NB,M,I-1)/R(I-1)**(K+1))

```

```

2*SNL(L,I-2)*PSI(NB,M,I-2)/R(I-2)**(K+1))
DO 4 I=3,161,2
4 S(I)=S(I-2)+((R(I)-R(I-1))/3.)*(SNL(L,I)*PSI(NB,M,I)*R(I)**K
+4.*SNL(L,I-1)*PSI(NB,M,I-1)*R(I-1)**K
2*SNL(L,I-2)*PSI(NB,M,I-2)*R(I-2)**K)
DO 5 I=1,161,2
5 S(I)=PSI(NB,M,I)*SNL(L,I)*(S(I)/R(I)**(K+1)
+R(I)**K*(SN(161)-SN(I)))
S(2)=0.5*(S(1)+S(3))
S(160)=0.5*(S(159)+S(161))
CALL INTP(S,F)
DO 6 I=1,159,2
6 ANS=ANS-((R(I+1)-R(I))/3.)*(S(I)+4.*S(I+1)+S(I+2))
ANS=2.0*ANS
RETURN
END
SUBROUTINE ZENERGY(Z,NEF,NEG,NTYPES)
COMMON A(11,11),B(11,11),RR(11),PSI(11,201),R(201),S(201),
IDEAD(9),X(16),NR(11),LR(11),WTR(11),IPBS(11),
ZNU(3,11),ZU(3,11),ZU(3,11),PSII(3,11,11),PSIO(396),
ZOV(3,11,11),ONE(3,11,11),N0(3),NEXP(3),NF(3),F(3,25),
+G(3,25,2),XF(3,25),XG(3,25),NG(3)
DOUBLE PRECISION DEAD,A,B,RR
DOUBLE PRECISION PSI,R,S,RHB,H,ENERGY,ANS,AAN,AN,ANN
REAL NU
INTEGER AD,F,G
3 FORMAT (3I4,F18.10)
4 FORMAT (17H) INITIAL ENERGY IS,2X,2E18.8)
5 FORMAT (20H) OFF BODY ENERGY IS,2X,2E18.8)
6 FORMAT (9H) EXTRA F,3I4,2E18.8)
7 FORMAT (9H) EXTRA G,3I4,2E18.8)
C SELF CONSISTENT SOLUTION ON HAND FORM
C AND NORMALIZE WAVE FUNCTIONS
RHB=-3.5
H=0.0525
DO 31 I=1,201,4
K(I)=DEXP(MH0)/2
31 RHB=RHB+4.*H
DO 32 I=1,197,4
H=(R(I+4)-K(I))/4.
K(I+1)=R(I)+H
K(I+2)=R(I+1)+H
32 K(I+3)=R(I)+3.*H
NT=N0(1)+N0(2)+N0(3)
DO 35 I=1,NT
DO 35 J=1,201
35 PSI(I,J)=0.0
L=0
DO 33 I=1,3
IF (N0(I).EQ.0) GO TO 33
NM=N0(I)
DO 34 M=1,NM
L=L+1
KK=NEXP(I)
DO 34 J=1,201
DO 34 K=1,KK
34 PSI(L,J)=PSI(L,J)+PSII(I,M,K)*NU(I,K)*R(J)**(I+AU(I,K))*
DEXP(-ZU(I,K)*R(J))
33 CONTINUE
DO 36 I=1,NT

```

```

ANS=0.0
DO 37 J=1,199,2
37 ANS=ANS+0.3333333333333333*(R(J+1)-R(J))*PSI(I,J)**2
1+4.*PSI(I,J+1)**2+PSI(I,J+2)**2)
DO 38 J=1,201
38 PSI(I,J)=PSI(I,J)/DSQRT(ANS)
36 S(I)=DSQRT(ANS)
K=0
DO 39 I=1,3
IF (NB(I).EQ.0) GO TO 39
JJ=NB(I)
N=NEXP(I)
DO 40 J=1,JJ
K=K+1
DO 40 L=1,N
40 PSII(I,J,L)=PSII(I,J,L)/S(K)
39 CONTINUE
DO 42 I=1,3
IF (NB(I).EQ.0) GO TO 42
JJ=NB(I)
DO 4200 J=1,JJ
KK=NEXP(I)
PRINT 1,(PSII(I,J,K),K=1,KK)
4200 CONTINUE
42 CONTINUE
DO 1801 I=1,NT
1801 PRINT 1,K(I),(PSI(J,I),J=1,NT)
C NORMALIZED FUNCTIONS ON HAND COMPUTE TOTAL ENERGY
1 FORMAT (1X,F12.6)
ENERGY=0.0
2 FORMAT (2I4,F18.6)
NT=NB(1)+NB(2)+NB(3)
DO 401 I=1,NT
401 IPBS(I)=I
I=1
DO 402 J=1,NTYPES
L=NB(J)
DO 403 K=1,L
NR(I)=J+K-1
LR(I)=J-1
WTR(I)=XF(I,I)
403 I=I+1
402 CONTINUE
202 FORMAT (2I4,F12.6)
C OBTAIN ONE BODY CONTRIBUTION TO TOTAL ENERGY
DO 204 I=1,NT
L=LR(I)+1
JJ=NEXP(L)
N=NR(I)-L+1
DO 204 J=1,JJ
DO 204 K=1,JJ
204 ENERGY=ENERGY+PSII(L,N,J)*PSII(L,N,K)*ONE(L,J,K)+WTR(I)
PRINT 5,ENERGY
C TWO BODY TERMS NOW
301 FORMAT (1X,2I4)
302 FORMAT (1X,F12.6)
DO 205 I=1,NT
DO 205 J=I,NT
IF (I.EQ.J) GO TO 206
II=IPBS(I)

```

```

      JJ=IP0S(J)
      L=LR(I)
      LL=LR(JJ)
      WT=WTR(I)*WTR(J)
      KEY=L+LL+L*LL+1
      CALL FGK(II,II,JJ,JJ,0,PSI,R,ANS,S)
      GO TO (207,208,209,210,210,211,211,211,212),KEY
207 CALL FGK(II,JJ,II,JJ,0,PSI,R,ANS,S)
      ENERGY=ENERGY+WT*ANS-0.5*WT*AN
      GO TO 205
208 CALL FGK(II,JJ,II,JJ,1,PSI,R,ANS,S)
      ENERGY=ENERGY+WT*ANS-WT*AN/6.0
      GO TO 205
209 CALL FGK(II,JJ,II,JJ,2,PSI,R,ANS,S)
      ENERGY=ENERGY+WT*ANS-WT*AN/10.
      GO TO 205
210 CALL FGK(II,JJ,II,JJ,2,PSI,R,ANN,S)
      CALL FGK(II,JJ,II,JJ,0,PSI,R,ANS,S)
      ENERGY=ENERGY-WT*ANN/15.+WT*ANS-WT*AN/6.
      GO TO 205
211 CALL FGK(II,JJ,II,JJ,1,PSI,R,ANS,S)
      CALL FGK(II,JJ,II,JJ,3,PSI,R,ANN,S)
      ENERGY=ENERGY-WT*ANN*3./70.+WT*ANS-WT*AN/15.
      GO TO 205
212 CALL FGK(II,JJ,II,JJ,0,PSI,R,ANS,S)
      CALL FGK(II,JJ,II,JJ,2,PSI,R,AAN,S)
      CALL FGK(II,JJ,II,JJ,4,PSI,R,ANN,S)
      ENERGY=ENERGY-WT*ANN/35.+WT*ANS-WT*AN/10.-WT*AAN/35.
      GO TO 205
206 II=IP0S(I)
      L=LR(I)+1
      IWT=WTR(I)+0.1
      WT=0.
      DO 213 III=1,IWT
213 WT=WT-1.0+FLOAT(III)
      GO TO (214,215,216),L
214 CALL FGK(II,II,II,II,0,PSI,R,ANS,S)
      ENERGY=ENERGY+WT*ANS
      GO TO 205
215 CALL FGK(II,II,II,II,0,PSI,R,ANS,S)
      ENERGY=ENERGY+WT*ANS
      CALL FGK(II,II,II,II,2,PSI,R,ANS,S)
      ENERGY=ENERGY-WT*ANS*2./25.
      GO TO 205
216 CALL FGK(II,II,II,II,0,PSI,R,ANS,S)
      CALL FGK(II,II,II,II,2,PSI,R,ANS,S)
      CALL FGK(II,II,II,II,4,PSI,R,ANN,S)
      ENERGY=ENERGY+WT*(ANS=AN*2./63.=ANN*2./63.)
205 CONTINUE
C TOTAL ENERGY ON HAND WE ARE DONE WITH THIS SET OF DATA
  PRINT 4,ENERGY
  IF (NEF.EQ.0) GO TO 48
  DO 46 L=1,NEF
  READ 3,I,J,K,WT
  CALL FGK(I,I,J,J,K,PSI,R,ANS,S)
  ANS=ANS*WT
  46 PRINT 6,I,J,K,ANS
  48 CONTINUE
  IF (NEG.EQ.0) GO TO 49
  DO 47 L=1,NEG

```

```

READ 3, I, J, K, WT
CALL FGK(I, J, I, J, K, PSI, R, ANS, S)
ANS=ANS*WT
47 PRINT 7, I, J, K, ANS
49 CONTINUE
END
SUBROUTINE FGK(A, B, C, D, K, SNL, R, ANS, S)
DIMENSION SNL(11, 201), R(201), S(201), SN(201)
DOUBLE PRECISION SNL, R, S, ANS, SN
INTEGER A, B, C, D
ANS=0.0
DO 1 I=1, 201
S(I)=0.0
1 SN(I)=0.0
DO 3 I=3, 201, 2
SN(I)=SN(I-2)+((R(I)-R(I-1))/3.)*(SNL(A, I)*SNL(B, I)/R(I)
1***(K+1)+4.*SNL(A, I-1)*SNL(B, I-1)/R(I-1)***(K+1)
2+SNL(A, I-2)*SNL(B, I-2)/R(I-2)***(K+1))
3 S(I)=S(I-2)+((R(I)-R(I-1))/3.)*(SNL(A, I)*SNL(B, I)
1***(K+1)+4.*SNL(A, I-1)*SNL(B, I-1)*R(I-1)**K
2+SNL(A, I-2)*SNL(B, I-2)*R(I-2)**K)
DO 5 I=1, 201, 2
5 S(I)=SNL(C, I)*SNL(D, I)*(S(I)/R(I)**(K+1)+R(I)**K*(SN(201)-SN(I)))
S(2)=0.5*(S(1)+S(3))
S(200)=0.5*(S(199)+S(201))
CALL INT(S, R)
DO 6 I=1, 199, 2
6 ANS=ANS+((R(I+1)-R(I))/3.)*(S(I)+4.*S(I+1)+S(I+2))
ANS=ANS*2.0
RETURN
END
SUBROUTINE INT(S, R)
DIMENSION S(201), R(201)
DOUBLE PRECISION S, R, A, B, C, D, E, F, G, H
DO 1 I=4, 198, 2
A=(R(I)-R(I-1))*(R(I)-R(I+1))*(R(I)-R(I+3))
B=(R(I)-R(I-3))*(R(I)-R(I+1))*(R(I)-R(I+3))
C=(R(I)-R(I-3))*(R(I)-R(I-1))*(R(I)-R(I+3))
D=(R(I)-R(I-3))*(R(I)-R(I-1))*(R(I)-R(I+1))
E=(R(I-3)-R(I-1))*(R(I-3)-R(I+1))*(R(I-3)-R(I+3))
F=(R(I-1)-R(I-3))*(R(I-1)-R(I+1))*(R(I-1)-R(I+3))
G=(R(I+1)-R(I-3))*(R(I+1)-R(I-1))*(R(I+1)-R(I+3))
H=(R(I+3)-R(I-3))*(R(I+3)-R(I-1))*(R(I+3)-R(I+1))
1 S(I)=S(I-3)*A/E+S(I-1)*B/F+S(I+1)*C/G+S(I+3)*D/H
RETURN
END
SUBROUTINE DIAG(A, B, N, M, RT)
DIMENSION A(11, 11), R(11, 11), RT(11)
DIMENSION AA(121), RR(121), GG(121)
COMMON /DIA/GAM(11, 11)
DOUBLE PRECISION A, R, RT, AA, BB, GG
1 FORMAT (1H1, 4NDIAG)
2 FORMAT (7H ENERGY, F14.5)
3 FORMAT (12H EIGENVECTR )
4 FORMAT (1X, 9F10.5)
I=0
DO 101 J=1, N
DO 101 K=1, N
I=I+1
AA(I)=A(K, J)

```

```

101 BB(1)=B(K,J)
CALL NROOT(F,AA,BB,RT,GG)
I=0
DO 102 J=1,N
DO 102 K=1,N
I=I+1
102 GAM(K,J)=GG(I)
C EIGENVALUES IN RT EIGENVECTORS IN GAM
C ORDER ANSWERS IN ASCENDING SEQUENCE
CALL ORDERRT(GAM,1,N)
RETURN
END
SUBROUTINE ORDER(V,S,NDIM,N)
DIMENSION V(NDIM),S(NDIM,NDIM)
DOUBLE PRECISION V
M1 = N-1
DO 92 M1=1,M1
M2 = M1+1
DO 92 M3=M2,N
IF (V(M3) - V(M1)) 90, 92, 92
90 HI = V(M1)
V(M1) = V(M3)
V(M3) = HI
DO 91 M4 = 1,M
HI = S(M4,M1)
S(M4,M1) = S(M4,M3)
91 S(M4,M3) = HI
92 CONTINUE
DO 93 M1=1,N
IF(S(1,M1)) 94, 93, 93
94 DO 95 M2=1,M
95 S(M2,M1)=-S(M2,M1)
96 CONTINUE
RETURN
END
SUBROUTINE EVEXT(F,ITRIGG,NTYPES,NB,NEXP,PSII,PSI0)
DIMENSION NB(1),NEXP(1),PSII(3,11,11),PSI0(3,11,11)
DIMENSION PSI3(3,11,11)
LOGICAL LLFRST
DATA ICDUNT/1/,LLFRST/.TRUE./
IF (ITRIGG*LT.1.GR.(LLFRST)) GO TO 3
ICDUNT=ICDUNT+1
IF (ICDUNT*LT.ITRIGG) GO TO 3
ICDUNT=0
DO 2 I=1,NTYPES
N=NB(I)
M=NEXP(I)
DO 2 J=1,M
DO 2 K=1,M
PSII(I,J,K)=3.*(PSII(I,J,K)-PSI0(I,J,K))+PSI3(I,J,K)
PRINT 900
900 FORMAT(11,'EXTRAPOLATION EMPLOYED')
3 LLFRST=.FALSE.
DO 4 I=1,NTYPES
N=NB(I)
M=NEXP(I)
DO 4 J=1,M
DO 4 K=1,M
PSI3(I,J,K)=PSII(I,J,K)
4 PSI0(I,J,K)=PSII(I,J,K)

```

```

RETURN
END
SUBROUTINE NR0BT (M,A,B,XL,X)
DIMENSION A(1),B(1),XL(1),X(1)
C .....NR00 370
C IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE NR00 380
C C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION NR00 400
C STATEMENT WHICH FOLLOWS. NR00 420
C NR00 430
C DOUBLE PRECISION A,B,XL,X,SUMV NR00 440
C DOUBLE PRECISION A,B,XL,X,SUMV NR00 450
C NR00 460
C THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS NR00 470
C APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS NR00 480
C ROUTINE. NR00 490
C NR00 500
C NR00 510
C THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO NR00 520
C CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. SQRT IN STATEMENTS NR00 530
C 110 AND 175 MUST BE CHANGED TO DSQRT. ABS IN STATEMENT 110 NR00 540
C MUST BE CHANGED TO DABS. NR00 550
C .....NR00 570
C COMPUTE EIGENVALUES AND EIGENVECTORS OF B NR00 590
C NR00 600
C K=1 NR00 610
C DO 100 J=2,M NR00 620
C L=M*(J-1) NR00 630
C DO 100 I=1,J NR00 640
C L=L+1 NR00 650
C K=K+1 NR00 660
C 100 B(K)=B(L) NR00 670
C NR00 680
C THE MATRIX B IS A REAL SYMMETRIC MATRIX. NR00 690
C NR00 700
C MV=0 NR00 710
C CALL EIGEN (B,X,M,MV) NR00 720
C NR00 730
C FORM RECIPROCATS OF SQUARE ROOT OF EIGENVALUES. THE RESULTS NR00 740
C ARE PREMULTIPLIED BY THE ASSOCIATED EIGENVECTORS. NR00 750
C NR00 760
C L=0 NR00 770
C DO 110 J=1,M NR00 780
C L=L+J NR00 790
C 110 XL(J)=1.0/DSQRT(DABS(B(L))) NR00 800
C K=0 NR00 810
C DO 115 J=1,M NR00 820
C DO 115 I=1,M NR00 830
C K=K+1 NR00 840
C 115 B(K)=X(K)*XL(J) NR00 850
C NR00 860
C FORM (B**(-1/2))PRIME * A * (B**(-1/2)) NR00 870
C NR00 880
C DO 120 I=1,M NR00 890
C N2=0 NR00 900
C DO 120 J=1,M NR00 910
C N1=M*(I-1) NR00 920
C L=M*(J-1)+I NR00 930
C X(L)=0.0 NR00 940
C DO 120 K=1,M NR00 950
C N1=N1+1 NR00 960
C N2=N2+1 NR00 970

```

120	X(L)=X(L)+B(N1)*A(N2)	NR00 980
	L=L+1	NR00 990
	DO 130 J=1,M	NR001000
	DO 130 I=1,J	NR001010
	N1=I-M	NR001020
	N2=M*(J-1)	NR001030
	L=L+1	NR001040
	A(L)=0.0	NR001050
	DO 130 K=1,M	NR001060
	N1=N1+M	NR001070
	N2=N2+1	NR001080
130	A(L)=A(L)+X(N1)*B(N2)	NR001090
C		NR001100
C	COMPUTE EIGENVALUES AND EIGENVECTORS OF A	NR001110
C		NR001120
	CALL EIGEN (A,X,M,MV)	NR001130
	L=L+1	NR001140
	DO 140 I=1,M	NR001150
	L=L+I	NR001160
140	XL(I)=A(L)	NR001170
C		NR001180
C	COMPUTE THE NORMALIZED EIGENVECTORS	NR001190
C		NR001200
	DO 150 I=1,M	NR001210
	N2=0	NR001220
	DO 150 J=1,M	NR001230
	N1=I-M	NR001240
	L=M*(J-1)+I	NR001250
	A(L)=0.0	NR001260
	DO 150 K=1,M	NR001270
	N1=N1+M	NR001280
	N2=N2+1	NR001290
150	A(L)=A(L)+B(N1)*X(N2)	NR001300
	L=L+1	NR001310
	K=0	NR001320
	DO 180 J=1,M	NR001330
	SUMV=0.0	NR001340
	DO 170 I=1,M	NR001350
	L=L+1	NR001360
170	SUMV=SUMV+A(L)*A(L)	NR001370
175	SUMV=DSQRT(SUMV)	NR001380
	DO 180 I=1,M	NR001390
	K=K+1	NR001400
180	X(K)=A(K)/SUMV	NR001410
	RETURN	NR001420
	END	NR001430
	SUBROUTINE EIGEN(A,R,N,MV)	EIGE 410
	DIMENSION A(1),R(1)	EIGE 420
C	.....	EIGE 440
C	IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE	EIGE 460
C	C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION	EIGE 470
C	STATEMENT WHICH FOLLOWS.	EIGE 480
C	DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,	EIGE 500
C	1 CRSX2,SINCS,RANGE	EIGE 510
	DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,	EIGE 500
	1 CRSX2,SINCS,RANGE	EIGE 510
C	THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS	EIGE 530
C	APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS	EIGE 540
C	ROUTINE.	EIGE 550
C	THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO	EIGE 570

```

C      * MAINTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  SORT IN STATEMENT EIGE 580
C      40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT.  ABS IN STATEMENT EIGE 590
C      62 MUST BE CHANGED TO DABS.  THE CONSTANT 5 SHOULD EIGE 600
C      BE CHANGED TO 1.0D-12. EIGE 610
C      ..... EIGE 630
C      10 RANGE=1.0E-12 EIGE 670
C      IF(NV=1) 11,25,13 EIGE 680
C      11 IQ=N EIGE 690
C      DO 20 J=1,N EIGE 700
C      IQ=IQ+N EIGE 710
C      DO 20 I=1,N EIGE 720
C      IQ=IQ+1 EIGE 730
C      R(IJ)=0.0 EIGE 740
C      IF(I=J) 21,15,2 EIGE 750
C      15 R(IJ)=1.0 EIGE 760
C      21 CONTINUE EIGE 770
C      ..... EIGE 780
C      COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX) EIGE 790
C      ..... EIGE 800
C      25 ANORM=0.0 EIGE 810
C      DO 35 I=1,N EIGE 820
C      DO 35 J=1,N EIGE 830
C      IF(I=J) 30,35,3 EIGE 840
C      3 IA=I+(J-J)/2 EIGE 850
C      ANORM=ANORM+A(IA)*A(IA) EIGE 860
C      35 CONTINUE EIGE 870
C      IF (ANORM) 100,100,40 EIGE 880
C      40 ANORM=1.414*SQRT(ANORM) EIGE 890
C      ANORMX=ANORM*RANGE/FLP(4) EIGE 900
C      ..... EIGE 910
C      INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR EIGE 920
C      ..... EIGE 930
C      IND=0 EIGE 940
C      THR=ANORM EIGE 950
C      45 THR=THR/FLP(10) EIGE 960
C      50 L=1 EIGE 970
C      55 M=L+1 EIGE 980
C      ..... EIGE 990
C      COMPUTE SIN AND COS EIGE1000
C      ..... EIGE1010
C      60 MV=(M+1)/2 EIGE1020
C      LV=(L+L-L)/2 EIGE1030
C      LM=L+MV EIGE1040
C      65 IF (DABS(A(LM))>THR) 130,65,65 EIGE1050
C      65 IND=1 EIGE1060
C      LL=L+LV EIGE1070
C      MM=M+MV EIGE1080
C      X=J+5*(A(LL)-A(MM)) EIGE1090
C      65 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X) EIGE1100
C      IF(X) 70,75,75 EIGE1110
C      70 Y=-Y EIGE1120
C      75 SINX=Y/DSQRT(1.+(1.+(DSQRT(1.0-Y*Y)))) EIGE1130
C      SINX2=SINX*SINX EIGE1140
C      70 COSX=DSQRT(1.-SINX2) EIGE1150
C      COSX2=COSX*COSX EIGE1160
C      SINCX =SINX*COSX EIGE1170
C      ..... EIGE1180
C      ROTATE L AND M COLUMNS EIGE1190
C      ..... EIGE1200
C      ILW=N*(L-1) EIGE1210

```

IMQ=N*(N-1)	EIGE1220
DO 125 I=1,N	EIGE1230
IQ=(I+I-1)/2	EIGE1240
IF(I=L) 80,115,80	EIGE1250
80 IF(I=M) 85,115,90	EIGE1260
85 IM=I+MQ	EIGE1270
GO TO 95	EIGE1280
90 IM=M+IQ	EIGE1290
95 IF(I=L) 100,105,105	EIGE1300
100 IL=I+LW	EIGE1310
GO TO 110	EIGE1320
105 IL=L+IQ	EIGE1330
110 X=A(IL)*COSX-A(IM)*SINX	EIGE1340
A(IM)=A(IL)*SINX+A(IM)*COSX	EIGE1350
A(IL)=X	EIGE1360
115 IF(MV=1) 120,125,120	EIGE1370
120 ILK=ILW+I	EIGE1380
IMR=IMQ+I	EIGE1390
X=R(ILR)*COSX-R(IMR)*SINX	EIGE1400
R(IMR)=R(ILR)*SINX+R(IMR)*COSX	EIGE1410
R(ILR)=X	EIGE1420
125 CONTINUE	EIGE1430
X=2.0*A(LM)*SINCS	EIGE1440
Y=A(LL)*COSX2+A(MM)*SINX2-X	EIGE1450
X=A(LL)*SINX2+A(MM)*COSX2+X	EIGE1460
A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)	EIGE1470
A(LL)=Y	EIGE1480
A(MM)=X	EIGE1490
C	EIGE1500
C TESTS FOR COMPLETION	EIGE1510
C	EIGE1520
C TEST FOR M = LAST COLUMN	EIGE1530
C	EIGE1540
130 IF(M=N) 135,140,135	EIGE1550
135 M=M+1	EIGE1560
GO TO 60	EIGE1570
C	EIGE1580
C TEST FOR L = SECOND FROM LAST COLUMN	EIGE1590
C	EIGE1600
140 IF(L=(N-1)) 145,150,145	EIGE1610
145 L=L+1	EIGE1620
GO TO 55	EIGE1630
150 IF(IND=1) 160,165,160	EIGE1640
155 IND=0	EIGE1650
C	EIGE1670
C COMPARE THRESHOLD WITH FINAL NORM	EIGE1680
C	EIGE1690
160 IF(THR=ANRMX) 165,165,45	EIGE1700
C	EIGE1710
C SORT EIGENVALUES AND EIGENVECTORS	EIGE1720
C	EIGE1730
165 I0=N	EIGE1740
DO 185 I=1,N	EIGE1750
I0=I0+N	EIGE1760
LL=I+(I+I-1)/2	EIGE1770
J0=N*(I-2)	EIGE1780
DO 185 J=1,N	EIGE1790
J0=J0+N	EIGE1800
MM=J+(J+J-1)/2	EIGE1810
IF(A(LL)-A(MM)) 170,185,185	EIGE1820

170	X=A(LL)	EIGE1830
	A(LL)=A(MM)	EIGE1840
	A(MM)=X	EIGE1850
	IF(MV=1) 175,185,175	EIGE1860
175	DO 180 K=1,N	EIGE1870
	ILR=IQ+K	EIGE1880
	IMR=JQ+K	EIGE1890
	X=R(ILR)	EIGE1900
	R(ILR)=R(IMR)	EIGE1910
180	R(IMR)=X	EIGE1920
185	CONTINUE	EIGE1930
	RETURN	EIGE1940
	END	EIGE1950
C	GENERATES POINT ION POTENTIAL	
	DIMENSION R(500),V(500)	
1	FORMAT (14)	
2	FORMAT (3F12.4)	
3	FORMAT (14,4F12.6)	
4	FORMAT (1H,2F12.4)	
5	FORMAT (8F10.5)	
100	READ 1,NREG	
	IF (NREG.EQ.0) GO TO 201	
	READ 3,IPT,RHO,H,Z,SIGN	
	DO 101 I=1,161,4	
	R(I)=EXP(RHO)/Z	
101	RHO=RHO+4.*H	
	DO 102 I=1,157,4	
	DEL=(R(I+4)-R(I))/4.	
	R(I+1)=R(I)+DEL	
	R(I+2)=R(I+1)+DEL	
102	R(I+3)=R(I+2)+DEL	
	NREG=NREG+1000	
	PRINT 1,NREG	
	READ 2,BBUND,CENS,CPFT	
	DO 11 I=1,IPT	
	IF (R(I)-BBUND) 12,12,13	
12	V(I)=(CENS+CPFT/R(I))*SIGN	
	GO TO 11	
13	READ 2,BBUND,CENS,CPFT	
	V(I)=(CENS+CPFT/R(I))*SIGN	
11	CONTINUE	
	PRINT 4,(V(I),R(I),I=1,IPT)	
	WRITE (8,5)(V(I),I=1,161)	
	GO TO 100	
201	CONTINUE	
	END FILE 8	
	REWIND 8	
	CALL EXIT	
	END	
C	WRITTEN IN FORTRAN IV BY A B KUNZ	
	DIMENSION SNLB(441),RX(441),VX(441),SNX(11,441)	
101	FORMAT (F10.4,14)	
102	FORMAT (1X,71H	
	,	
104	FORMAT (1X,5F12.6)	
7563	CONTINUE	
	SNLB(1)=0.0	
	I=1	
	DEL=0.00250	
	DO 1 J=1,11	

```

      DO 2 K=1,4
      I=I+1
2     SNLB(I)=SNLB(I-1)+DEL
1     DEL=DEL+DEL
      READ 101,ZX,RX
      ZEROB=0.0
      IF (ZX.EQ.0) GO TO 7001
      VBL=AA*AA*AA*FAC
C     FORM R MFSH AND READ IN DATA
      CX=0.88534138/ZX**(1./3.)
      DO 3 I=1,441
      RX(I)=SNLB(I)*CX
3     CONTINUE
      CALL WFN(NX,SNX,RX)
      DO 14 I=1,441
14    SNLB(I)=0.0
      DO 15 I=1,NX
      READ 101,WT
      DO 15 J=1,441
15    SNLB(J)=SNLB(J)+SNX(I,J)*SNX(I,J)*WT
      VX(1)=-2.*ZX
      DO 18 I=2,441
      VX(I)=VX(I-1)+(PX(I)-RX(I-1))*(SNLB(I)+SNLB(I-1))
18    CONTINUE
      DO 27 I=2,441
      AN=0.0
      DO 28 J=1,441
28    AN=AN+(RX(J+1)+PX(J))*(SNLB(J)/RX(J)+SNLB(J+1)/RX(J+1))
27    VX(I)=VX(I)+AN*PX(I)
      VX(441)=VX(440)
      DO 26 I=1,441
26    VX(I)=VX(I)-VX(441)
      WRITE (5,1041)(VX(I),I=1,441)
      GO TO 7563
7001 REWIND 7
      END FILE 5
      REWIND 5
      CALL EXIT
      END
      SUBROUTINE WFN(NX,SN,R)
      DIMENSION SN(11,441),R(451),CJ(11),NJ(11),AJ(11),ZJ(11)
      REAL NJ
      INTEGER NJ
1     FORMAT (2I4)
2     FORMAT (6F12.6)
3     FORMAT(11I4)
4     FORMAT (1X,3F10.5)
      DO 5 I=1,NX
      READ (7,1)(R,I)
      READ (7,2)(CJ(I),J=1,N)
      READ (7,3)(AJ(I),J=1,N)
      READ (7,2)(ZJ(I),J=1,N)
      DO 6 J=1,N
      Z=2.*ZJ(J)
      IP=2*L+3+2*NJ(J)
      Z=Z**IP
      IP=IP-1
      I=FACTB(IP)
      NJ(J)=SQRT(Z/I)
      DO 7 J=1,441

```

```

7 SN(I,J)=0.0
  DO 8 J=1,441
    DO 8 K=1,N
8 SN(I,J)=SN(I,J)+CJ(K)*NJ(K)*EZP(-ZJ(K)*R(J))*R(J)**(L+1+AJ(K))
  SUM=0.0
  DO 9 J=1,439,2
9 SUM=SUM+((R(J+1)-R(J))/3.)*(SN(I,J)**2+4.*SN(I,J+1)**2+SN(I,J+2)
  I**2)
  SUM=SQRT(SUM)
  DO 10 J=1,441
10 SN(I,J)=SN(I,J)/SUM
  DO 11 J=1,N
11 CJ(J)=CJ(J)/SUM
5 PRINT 4,(CJ(J),J=1,N)
  RETURN
  END

FUNCTION FACT0(IS)
  IF (IS) 1,1,2
1 FACT0=1.0
  RETURN
2 A=IS
  DO 3 I=1,100
  IS=IS-1
  IF (IS) 4,4,5
5 B=IS
3 A=A*B
4 FACT0=A
  RETURN
  END

FUNCTION EZP(X)
  XL=-25.0
  IF (X.LT.XL) GO TO 1
  EZP=EXP(X)
  RETURN
1 EZP=0.0
  RETURN
  END

C DOUBLE INTEGRATIONS PERFORMED FOR CERTAIN SOLID STATE CASES 1
C WRITTEN FOR THE CDC 6600 FALL 1966 AT LEHIGH UNIVERSITY BY A B 2
C KUNZ IN FORTRAN 3
C EXTENDED TO INCLUDE D STATES BY E. CALABRESE, MAY 1968 4
COMMON SNLP(6,441),X(441),RI(6,441),SNL1(441),SNL3(441),
IRK1(441),RR3(441),XFUN(441),Z(20)
10 FORMAT (I2) 7
20 FORMAT (10F6.1) 8
30 FORMAT (1X,5F12.7)
40 FORMAT (6I2,F12.6,I4) 10
50 FORMAT (1H1,17X,20HTW0-CENTER INTEGRALS/) 11
60 FORMAT (1H ,5F12.6) 12
70 FORMAT (43H0THE ABOVE ARE AFTER INTEGRATION OVC.F. THETA,/) 13
80 FORMAT (16H FUNCTIONS USED ,3I2,1X,3I2,1H,,7X,3HA =,F10.6,2H , ,12X14
1,32HTHE RESULT OF THE INTEGRATION IS,3X,F14.7,/) 15
  DO 5001 I=1,161
5001 X(I)=0.0
  DO 5002 I=1,4
5002 WRITE (9,3007)(X(I),I=1,161)
  REWIND 9
  PRINT 50 16
  READ 10,NBFUN 17
  READ 20,(Z(I),I=1,NBFUN) 18

```

	DO 1 I=1,NDFUN	19
	READ (5,30)(SNLF(I,J),J=1,441)	
1	CONTINUE	21
C	GENERATE X AND R MESHES	22
	DELTA=0.00250	23
	I=1	24
	X(1)=0.0	25
	DO 2 J=1,11	26
	DO 3 K=1,40	27
	I=I+1	28
3	X(I)=X(I-1)+DELTA	29
2	DELTA=DELTA+DELTA	30
	DO 4 I=1,NDFUN	31
	C=0.88534138/Z(I)**(1./3.)	32
	DO 5 J=1,441	33
5	RI(I,J)=C*X(J)	34
4	CONTINUE	35
C	X AND R MESHES GENERATED	36
	DO 6 I=1,NDFUN	37
	SNLB(I,1)=0.0	38
	DO 7 J=2,441	39
7	SNLB(I,J)=SNLB(I,J)/RI(I,J)	40
6	CONTINUE	41
	DO 4043 I=1,NDFUN	
	Z(I+NDFUN)=Z(I)	
	DO 4043 J=1,441	
	SNLB(I+NDFUN,J)=1.0	
4043	RI(I+NDFUN,J)=RI(I,J)	
C	PROCEED TO SET UP THE INTEGRATION	42
	DO 14 K=1,441	44
	RI(5,K)=X(K)	
14	SNLB(5,K)=1.0	
	Z(5)=1.0	
16	READ 40,ICE,ICEN,IPTH,L1,L2,L3,A,INF	47
	IF (INF) 8,9,9	48
9	DO 11 I=1,441	49
	SNL1(I) = SNLF(ICE,I)*SNLB(ICEN,I)	50
	SNL3(I) = SNLF(IPTH,I)	51
	RR1(I)=RI(ICE,I)	52
12	RR3(I)=RI(IPTH,I)	53
11	CONTINUE	54
	CK=0.88534138/Z(IPTH)**(1./3.)	55
	CALL XPAN (L1,L2,L3,A,IPTH,CK,SNL3,RR1,RR3,XFUN)	56
	ZZ=Z(ICE)	
	RHO=-3.0	
	H=0.0625	
	DO 3001 I=1,161,4	
	RI(6,I)=EXP(RHO)/ZZ	
3001	RHO=RHO+4.*H	
	DO 3002 I=1,157,4	
	DEL=(RI(6,I+4)-RI(6,I))/4.	
	RI(6,I+1)=RI(6,I)+DEL	
	RI(6,I+2)=RI(6,I+1)+DEL	
3002	RI(6,I+3)=RI(6,I+2)+DEL	
C	INTERPOLATE XFUN AND STORE IN X	
	DO 3003 I=1,161	
	RR=RI(6,I)	
	DO 3004 J=1,441	
	K=J	
	IF (RR=RI(ICE,J)) 3005,3006,3004	

3004	CONTINUE	
3006	X(I)=XFUN(K)	
	GO TO 3003	
3005	X(I)=XFUN(K-1)+(RR-RI(ICE,K-1))*(XFUN(K)-XFUN(K-1))/(RI(ICE,K)-RI(ICE,K-1))	
3003	CONTINUE	
3007	FORMAT (1X,5F12.6)	
	WRITE (9,3007)(X(I),I=1,161)	
	DO 13 I=1,441	57
13	XFUN(I) = SNL(I)*XFUN(I)	58
	ANS=0.0	59
	CALL SIMPS (ANS,XFUN,RR1)	60
	IF (INF) 17,15,17	61
17	PRINT 60,(XFUN(I),I=1,441)	62
	PRINT 70	63
15	PRINT 80,ICE,ICEN,ITH,L1,L2,L3,A,ANS	64
	GO TO 16	65
8	CONTINUE	66
	REWIND 5	
	END FILE 9	
	REWIND 9	
	CALL EXIT	
	END	68
	SUBROUTINE XFUN(LL1,LL2,LL3,A,KL0,CK,SNL,RR1,RR3,XFUN)	69
C	SUBROUTINE XFUN(LL1,LL2,LL3,A,KL0,CK) USE FOR DOUBLE INTEGRATION	70
	DIMENSION SNL(441),RR1(441),RR3(441),XFUN(441),THFUN(101)	71
C	CHANGE VARIABLES NOW	72
	L1=LL1	73
	L2=LL2	74
	L3=LL3	75
	UTH=0.314159	76
	XFUN(1)=.	77
	DO 1 M=2,441	78
	J0=1	79
	XFUN(M)=.	80
	THETA=.	81
	RR=RR1(M)	82
30	DO 2 N=1,101	83
	YI=RR*SINF(THETA)	84
	ZI=RR*COSF(THETA)	85
	YK=YI	86
	ZK=ZI-A	87
	R=SQRT(YK**2+ZK**2)	88
	SIK=YK/R	89
	COK=ZK/R	90
C	GET ANGULAR COEFFICIENT	91
	GO TO (3,21,5,4,22,23,24),L1	92
3	TH1=1.0	93
	GO TO 6	94
21	TH1=1./2.**((1./2.**)	95
	GO TO 6	96
5	TH1=(3./2.**((1./2.**)*COSF(THETA)	97
	GO TO 6	98
4	TH1=(3./4.**((1./2.**)*SINF(THETA)	99
	GO TO 6	100
22	TH1=(5./2.**((1./2.***(3./2.*COSF(THETA)*COSF(THETA)-1./2.**)	101
	GO TO 6	102
23	TH1=15.**((1./2.***(1./2.***SINF(THETA)*COSF(THETA)	103
	GO TO 6	104
24	TH1=15.**((1./2.***(1./4.***SINF(THETA)*SINF(THETA)	105

6	CONTINUE	106
	GO TO (7,25,9,8,26,27,28),L2	107
7	TH2=1.0	108
	GO TO 10	109
25	TH2=1./2.**(.1./2.)	110
	GO TO 10	111
9	TH2=(3./2.)**(.1./2.)*COSF(THETA)	112
	GO TO 10	113
8	TH2=(3./4.)**(.1./2.)*SINF(THETA)	114
	GO TO 10	115
26	TH2=(5./2.)**(.1./2.)*(3./2.*COSF(THETA)*COSF(THETA)-1./2.)	116
	GO TO 10	117
27	TH2=15.**(.1./2.)/2.*SINF(THETA)*COSF(THETA)	118
	GO TO 10	119
28	TH2=15.**(.1./2.)/4.*SINF(THETA)*SINF(THETA)	120
10	CONTINUE	121
	GO TO (11,29,13,12,30,31,32),L3	122
11	TH3=1.0	123
	GO TO 14	124
29	TH3=1./2.**(.1./2.)	125
	GO TO 14	126
13	TH3=(3./2.)**(.1./2.)*COK	127
	GO TO 14	128
12	TH3=(3./4.)**(.1./2.)*SIK	129
	GO TO 14	130
30	TH3=(5./2.)**(.1./2.)*(3./2.*COK*COK-1./2.)	131
	GO TO 14	132
31	TH3=15.**(.1./2.)/2.*SIK*COK	133
	GO TO 14	134
32	TH3=15.**(.1./2.)/4.*SIK*SIK	135
14	CONTINUE	136
	IF (KLD = 20) 34,33,33	137
33	RAD = 1.0	138
	GO TO 19	139
34	DO 15 I=1,441,40	140
	JG=I	141
	J=I	142
	IF (RR3(I)=R) 16,17,18	143
16	CONTINUE	144
15	CONTINUE	145
17	RAD = SNL(J)	146
	GO TO 19	147
18	JAK=J/40	148
	TAK=2**(.JAK-1)	149
	KX=RR3(J)=R	150
	CXK=CX*TAK*0.00250	151
	JUG=RX/CXK	152
	J=J-JUG	153
	RAD = SNL(J-1)+(R-RR3(J-1))*(SNL(J)-SNL(J-1))/	154
	I(RR3(J)=RR3(J-1))	155
19	CONTINUE	156
	THFUN(N) = RAD*TH1*TH2*TH3*SINF(THETA)	157
2	THETA=THETA+DTH	158
	DO 20 N=1,99,2	159
	XFUN(N)=XFUN(N)+(DTH/3.)*(THFUN(N)+4.*THFUN(N+1)+THFUN(N+2))	160
20	CONTINUE	161
1	CONTINUE	162
	RETURN	163
	END	164
	SUBROUTINE SIMPS (ANS,XFUN,RR1)	165

	DIMENSION XFUN(441),RR1(441),TLFUN(441)	166
C	INTEGRATE	167
	DO 1 I=1,441	168
1	TLFUN(I) = XFUN(I)*RR1(I)*RR1(I)	169
	DO 2 I=1,439,2	170
	ANS=ANS+((RR1(I+1)-RR1(I))/3.)*(TLFUN(I)+4.*TLFUN(I+1)+TLFUN(I+2))	171
2	CONTINUE	172
	RETURN	173
	END	174
	FUNCTION SIN(X)	
	SINF=SIN(X)	
	RETURN	
	END	
	FUNCTION SQRT(X)	
	SQRTF=SQRT(X)	
	RETURN	
	END	
	FUNCTION COS(X)	
	COSF=COS(X)	
	RETURN	
	END	

## IX. THE BAND THEORY PROGRAM

This is a complicated program and the control option and data are spelled out including examples here.

### IX. USE OF THE CODE

#### CONTROL CARDS

```
!dob
!LIMIT
!ASSIGN F:5, (FILE, HSI DATA), (IN), (SAVE)
!ASSIGN F:6, (DEVICE, LO)
!ASSIGN F:13, (SAVE), (OUTIN), (FILE,HFILE 13),(SN,KUZ6)
!ASSIGN F:20, (SAVE), (OUTIN), (TRIES,20), (FILE, HFOPKMTHS), (SN, KUZ6)
!ASSIGN F:21, (SAVE), (OUTIN), (TRIES,20), (FILE, HF1RRSTR), SN, KUZ6)
!ASSIGN F:22, (SAVE), (OUTIN), (TRIES,20), (FILE, HF2CCTH), (SN,KUZ6)
!ASSIGN F:23, (SAVE),(OUTIN), (TRIES,20), (FILE, HF3 COEF), (SN,KUZ6)
!ASSIGN F:24, (SAVE), (OUTIN), (TRIES,20),(FILE, HF4BNLBTH), (SN,KUZ6)
!ASSIGN F:25, (SAVE), (OUTIN), (TRIES,20),(FILE, HF5PLMCTH), (SN,KUZ6)
!ASSIGN F:26, (FILE, ALPAASI), (OUTIN), (RANDOM), (RSTORE, 747), (SAVE),
(SN,KUZ6)
!ASSIGN F:27, (FILE, ALPXXSI), (OUTIN), (RANDOM), (RSTORE,690), (SAVE),
(SN,KUZ6)
!ASSIGN F:28, (FILE, ALPAXSI), (OUTIN), (RANDOM), RSTORE, 737), (SAVE),
(SN,KUZ6)
!ASSIGN F:29, (FILE,ALPXASI), (OUTIN), (RANDOM), (RSTORE, 681), (SAVE),
(SN,KUZ6)
!ASSIGN F:30, (FILE, NTGRLSI), (OUTIN), (SAVE), (TRIES,20), (SN, KUZ6)
!ASSIGN F:31, (FILE, SIEV), (OUTIN), (SAVE), (TRIES, 20), (SN, KUZ6)
!ASSIGN F:32, (DEVICE, NO)
!ASSIGN M:LM, (FILE, HLMNBANG), ISN, KUZ6)
!OLAY (EF, (HBANGROM), (CHCKPTBO, MRLIB)), (LMN,HLMNBANG), (PERM), (MAP)
!RUN (LMN,HLMNBANG), (SN, KUZ6)
!DATA
Checkpoint time here
!FIN
```

---

above, the files are:

HSIDATA, input data file, to be described later  
HFILE13, scratch  
HFOPKMTHS, HF1RRSTR, HF2CCTH, HF3COEF, HF4RNCRTH, HF5PLMCTH, all scratch  
ALPAASI, ALPXXSI, ALPAXSI, ALPXASI,  $\alpha$  expansion files: A functions about  
other A sites, X functions about X sites, A functions expanded about X sites,  
X functions expanded about A sites respectively.  
NTGRLSI, integral file  
SIEV, contains answers: labeling info, wavefunctions, eigenvalues

IX. - 2 INPUT DATA DEFINED

RANDOM  $\alpha$ -FILE DETERMINATION

for AA set:

# of words = MXPTS \* MXPAN \* (NSETS \* LADINX (LAMAX + 2) + 1)

MXPTS = # of radial grid points used

MXPAN = (# of L' values used in  $\alpha$ -expansions)-1

NSETS = # of neighbor shells of type we are expanding

LADINX = # of functions, both occupied and virtual, on site A, counting 0 and position only M values, and counting only functions below the value of  $\ell$  specified. ( $\ell$  is the subscript -1)

Example: have 2 s functions  
          3 p functions  
          1 d function

then LADINX(1) = 0

      LADINX(2) = 2

      LADINX(3) = 8

      LADINX(4) = 11 = LADINX(LAMAX + 2), if LAMAX = 2

for AX set (A orbitals expanded about site X) like above, but don't add 1 to # of words

for XX set, like above for AA, but with X

for AA like above but don't add 1

DATA FILE

Card 1. LLD TAL, NROUT, LLMTRX; 10X, L5, I5, L5  
LLDTAL: logical variable is TRUE if a debugging run is made, otherwise set it FALSE  
NROUT: set equal to 3  
LLMTRX: logical variable causes matrices to be printed out along with bands

Card 2. LLA $\emptyset$ A, NEIGHA, LLA $\emptyset$ X, NEIGHX; 2(L5, I5)  
LLA $\emptyset$ A: TRUE if orbitals exist at site A( true if only one type of atom)  
NEIGHA: # of shells of neighbors of A to be considered (up to 6)  
LLA $\emptyset$ X: TRUE if orbitals exist at site X also  
NEIGHX: # of shells of neighbors of X to be considered (up to 6)

Card 3: Z, ZION, ZR, RHO, H, IPTS, NØCC, NTØT (5F10.0, 3I5)  
 Z: atomic # of atoms at A site  
 ZION: ionicity (e.g. -1 for F in LiF)  
 ZR: integration parameter. Set at 15.0  
 RHO: integration parameter. Set at -2.5  
 H: integration parameter. Set at .05  
 IPTS: # of points in mesh. Use 161  
 NØCC: # of occupied orbitals at the center (e.g. 3 for F in LiF since F<sup>-</sup> is 1s<sup>2</sup> 2s<sup>2</sup> 2p<sup>6</sup>)  
 NTØT: total # of orbitals, incl. virtual at site A (For site A maximum is MXAFN in program, for site X maximum is MXXFN).

Card 4: L, NBASES, LØCFN, LTØFN: 4I5  
 L: l quantum number of atomic shell  
 NBASES: # of basic functions for this l (up to 15)  
 LØCFN: # of occupied shells for this l  
 LTØFN: # of total shells for this l (total # of functions we wish to build for this l)

Card 5: As many cards as needed. With 8 (F8,0,I2) read in exponent and principal quantum number of each of the NBASES basis functions (S.T.Ø.) The function we wish to make a block function out of has this form

$$\phi_{nlm} = Y_l^m(\theta, \phi) \sum_{j=1}^{NBASES} C_j^{nl} N_{lj} r^{\eta_{jl}-1} e^{-Z_{lj}r}$$

So for each l, for all j, we read in Z<sub>lj</sub> and η<sub>lj</sub>

Card 6: As many cards as needed. With 8F10.0, with one logical record per each of the LTØFN functions, read in the NBASES coefficients plus FR, FRSP, RCUT  
 Must give occupied functions first

FR: effective charge in units of -|e| for electrons in that s shell, if FR = 0, default is 1. Room must be left for FR whether or not shell is occupied

FRSP: fraction of the spin shell which we are solving for which is occupied for this shell (i.e., for a 2p shell, if there are 3 spin up p electrons and one spin down, FRSP = 1. and FR is .666666 for that shell if we are solving for the spin up solutions.

RCUT: function will be truncated for distances beyond this value. (if 0. - no truncation).

Repeat cards 4, 5, and 6 for each  $\ell$  value on first center (A).

If LLA $\emptyset$ X is true, repeat card 3 and set of 4, 5, and 6 for center X.

Card 7: A, UNIT, NXPAN, ITHPTS, LLGENA, LCHALL, LLBUF; F10.0, F10.5,  
2I5, 3L5.

A: cube edge in Bohr

UNIT: use .25 for diamond or z inc blend lattice

NXPAN: # of  $\alpha$  functions per expansion (max = 30)

ITHPTS: Set at 101 (angular integration grid for  $\alpha$  function generation)

LLGENA: Set TRUE if  $\alpha$  functions need to be generated this run  
Set FALSE if  $\alpha$  functions have already been calculated and are on hand in their files

LLCHAL: Must be set TRUE if LLGENI (see below) is true.  
Otherwise, set TRUE if presence and size of  $\alpha$  files are to be checked.

LLBUF: TRUE if Buffer IN/OUT Sigma-5 subroutines are to be used. Set FALSE for other computer.

Cards 8-10 pertain to site A one center integrals

Card 8: NAF(I), NAG(I), I=1, LAMAX1 = 3; 2I4 (three cards)

NAF(I): # of F type integrals to be calculated for  $\ell$ value =  
I - 1 type electrons feeling presence of each shell in turn

NAG(I): # of G type integrals (See Hartree's book for definition of F and G integrals)

Card 9: FA(I,J), XAF(I,J), I = 1, LAMAX + 1; J = 1, NAF(I); IH, F12.9  
(as many cards as needed)

FA(integer) tells which shell is causing the potential being considered in the present integral and XF tells its weight. If I = 1, 2, or 3, then we are finding contributions to the potential of s, p, or d electrons respectively. Always list shells as follows: first all occupied s shells, then all occupied p shells, then all occupied d shells.

Card 10: GA(I,J,1), GA(I,J,2), XAG(I,J), I=1, LAMAX + 1, J = 1, NAG(I);  
2I4, F12.9

GA(I,J,1): gives K value for the G(K) integral

GA(I,J,2): tells which shell produces potential

XAG(I,J): is multiplier for that integral

Repeat cards 8, 9, and 10 for site X if LLA $\emptyset$ X = TRUE

Card 11: NBLCHA, NBLCHX; 2I5

NBLCHA: # of Bloch functions made up of orbitals at center A  
(counts all m values and uses all functions built up  
in cards 4, 5, and 6.)

NBLCHX: as above for site X

Card 12,a,b: for each function read in through card 6 we assign  
angular quantum # LA (as before) and label with principal  
quantum number NA( no relation to the N read in for STO  
evaluation) beginning with LA, LA + 1, etc. until all func-  
tions from card 6 have been assigned in order.

Card 12a NA, LA: 2I5

Card 12b: Using format 8F10.0 supply the complex coefficients of  
 $Y_l^m$ ,  $m = 0, +1, -1, \dots$  for the Bloch function whose NA and  
LA values are read in just before

Repeat Cards 12 NBLCHA times

Repeat Cards 12 NBLCHX times

Card 13: LLETE, NLETE; L5, I5

LLETE: set TRUE, if some Bloch functions are to be deleted  
from already calculates H and S matrices due to  
linear dependences

NLETE: # of basis functions to be deleted

Card 14: KLETE; 16I5  
(skip card 14 if LLETE = FALSE)

KLETE: the serial # of each basis function to be deleted  
(will be numbers anywhere from 1 to NBLCHA + NBLCHX)

Card 15: XKX, XKY, XKZ, ITW, LLGENI, NEIGHA, NEIGHX, NVEC; eF10.0,  
I5, L5, 3I5

XKX, XKY, XKZ: the X, Y, and Z components of k vector

ITW: # of equivalent k's in B.Z.

LLGENI: set TRUE if all integrals are to be generated

NEIGA.: # of shells of neighbors at A (overrides previous  
specification unless 0)

NEIGX: as NEIGA, for center X

NVEC: Indicates # of eigenvectors to be printed; if zero,  
none will be printed

Repeat Card 15 for each k point desired. LLGENI will automatically be  
set FALSE for all successive reads

IX-3. LIST OF PROGRAMS

13:39 APR 08, '76 ID=060A-H01  
 !JOB MRL2427,HALL,7  
 PCL

COPY	MSIDATA	TB	LP(K)
1	1.000		F,3,F
2	2.000	TRUE	6TRUE 6
3	3.000	14.0,0,15.0,2.5,0.05,161,5,8,	
4	4.000	0,8,3,4,	
5	5.000	16.102,1,14.057,2,8.303,3,4.412,2,2.012,3,1.265,3,0.6,3,2.22,3,	
6	6.000	.952536,304373,005721,-.000140,000039,-.000030,000005,000000	
7	7.000	1,1,1,	
8	8.000	.223177,1.23767,-.186877,-.948639,-.005748,001474,-.000268,000000	
9	9.000	1,1,1,	
10	10.000	.063185,042032,-.024748,-.370712,.751079,540439,002511,000000	
11	11.000	.5,1,1,	
12	12.000	.063185,042032,-.024748,-.370712,000000,000000,000000,751079	
13	13.000		
14	14.000	1,6,2,3,	
15	15.000	11.347,2,6.416,2,3.822,2,1.743,3,1.023,3,0.5,3,	
16	16.000	.050129,508422,859623,006365,-.001111,000089,1,1,1,	
17	17.000		
18	18.000	.007980,103307,160440,-.560568,-.805777,003210,5,1,	
19	19.000		
20	20.000	.007980,103307,160440,-.805777,000000,000000,	
21	21.000		
22	22.000	2,1,0,1,	
23	23.000	1.32,3,	
24	24.000	1.000	
25	25.000	14.0,0,15.0,2.5,0.05,161,5,9,	
26	26.000	0,10,3,6,	
27	27.000	16.102,1,14.057,2,8.303,3,4.412,2,2.012,3,1.265,3,0.6,3,2.22,3,	
28	28.000	.855,3,1.125,4,	
29	29.000	.952536,304373,005721,-.000140,000039,-.000030,000005,000000	
30	30.000	0.0,0.0,1.0,1.0,	
31	31.000	.223177,1.23767,-.186877,-.948639,-.005748,001474,-.000268,000000	
32	32.000	0.0,0.0,1.0,1.0,	
33	33.000	.063185,042032,-.024748,-.370712,.751079,540439,002511,000000	
34	34.000	0.0,0.0,5.0,	
35	35.000	.063185,042032,-.024748,-.370712,000000,000000,000000,751079	
36	36.000	0.0,0.0,	
37	37.000		
38	38.000	1.0,	
39	39.000		
40	40.000	0.0,1.0,	
41	41.000	1,7,2,3,	
42	42.000	11.347,2,6.416,2,3.822,2,1.743,3,1.023,3,0.5,3,1.125,4,	
43	43.000	.050129,508422,859623,006365,-.001111,000089,0.0,1,	
44	44.000	1,	
45	45.000	.007980,103307,160440,-.560568,-.805777,003210,0.0,5	
46	46.000		
47	47.000	0.0,0.0,0.0,0.0,0.0,0.0,1.0,	
48	48.000		
49	49.000	10.2612,25,30,101, FAKE TRUE TRUE	
50	50.000	5,5,	
51	51.000	5,7,	
52	52.000	5,7,	
53	53.000	1,2,	
54	54.000	2,2,	
55	55.000	3,1,	

56 - 56.000 4,6.  
57 - 57.000 5,3.  
58 - 58.000 1,2.  
59 - 59.000 2,2.  
60 - 60.000 3,1.  
61 - 61.000 4,6.  
62 - 62.000 5,3.  
63 - 63.000 1,2.  
64 - 64.000 2,2.  
65 - 65.000 3,1.  
66 - 66.000 4,6.  
67 - 67.000 5,3.  
68 - 68.000 0,1,1.  
69 - 69.000 0,2,1.  
70 - 70.000 0,3,1.  
71 - 71.000 1,4,1.  
72 - 72.000 1,5,1.  
73 - 73.000 1,1,3333333  
74 - 74.000 1,2,3333333  
75 - 75.000 1,3,3333333  
76 - 76.000 0,4,1.  
77 - 77.000 2,4,4  
78 - 78.000 0,5,1.  
79 - 79.000 2,5,4  
80 - 80.000 2,1,2  
81 - 81.000 2,2,2  
82 - 82.000 2,3,2  
83 - 83.000 1,4,4  
84 - 84.000 3,4,2571429  
85 - 85.000 1,5,4  
86 - 86.000 3,5,2571429  
87 - 87.000 5,5,  
88 - 88.000 5,7,  
89 - 89.000 5,7,  
90 - 90.000 1,2.  
91 - 91.000 2,2.  
92 - 92.000 3,1.  
93 - 93.000 4,6.  
94 - 94.000 5,3.  
95 - 95.000 1,2.  
96 - 96.000 2,2.  
97 - 97.000 3,1.  
98 - 98.000 4,6.  
99 - 99.000 5,3.  
100 - 100.000 1,2.  
101 - 101.000 2,2.  
102 - 102.000 3,1.  
103 - 103.000 4,6.  
104 - 104.000 5,3.  
105 - 105.000 0,1,1.  
106 - 106.000 0,2,1.  
107 - 107.000 0,3,0.  
108 - 108.000 1,4,1.  
109 - 109.000 1,5,0.  
110 - 110.000 1,1,3333333  
111 - 111.000 1,2,3333333  
112 - 112.000 1,3,0.  
113 - 113.000 0,4,1.  
114 - 114.000 2,4,4  
115 - 115.000 0,5,0.





236	=	235.000	.625000	.250000	.000000	24	F	0	0	33
237	=	236.000	.625000	.250000	.125000	48	F	0	0	33
238	=	237.000	.625000	.250000	.250000	24	F	0	0	33
239	=	238.000	.625000	.375000	.000000	24	F	0	0	33
240	=	239.000	.625000	.375000	.125000	48	F	0	0	33
241	=	240.000	.625000	.375000	.250000	48	F	0	0	33
242	=	241.000	.625000	.375000	.375000	24	F	0	0	33
243	=	242.000	.625000	.500000	.000000	24	F	0	0	33
244	=	243.000	.625000	.500000	.125000	48	F	0	0	33
245	=	244.000	.625000	.500000	.250000	48	F	0	0	33
246	=	245.000	.625000	.500000	.375000	48	F	0	0	33
247	=	246.000	.625000	.625000	.000000	12	F	0	0	33
248	=	247.000	.625000	.625000	.125000	24	F	0	0	33
249	=	248.000	.625000	.625000	.250000	24	F	0	0	33
250	=	249.000	.750000	.000000	.000000	6	F	0	0	33
251	=	250.000	.750000	.125000	.000000	24	F	0	0	33
252	=	251.000	.750000	.125000	.125000	24	F	0	0	33
253	=	252.000	.750000	.250000	.000000	24	F	0	0	33
254	=	253.000	.750000	.250000	.125000	48	F	0	0	33
255	=	254.000	.750000	.250000	.250000	24	F	0	0	33
256	=	255.000	.750000	.375000	.000000	24	F	0	0	33
257	=	256.000	.750000	.375000	.125000	48	F	0	0	33
258	=	257.000	.750000	.375000	.250000	48	F	0	0	33
259	=	258.000	.750000	.375000	.375000	24	F	0	0	33
260	=	259.000	.750000	.500000	.000000	24	F	0	0	33
261	=	260.000	.750000	.500000	.125000	48	F	0	0	33
262	=	261.000	.750000	.500000	.250000	48	F	0	0	33
263	=	262.000	.750000	.625000	.000000	24	F	0	0	33
264	=	263.000	.750000	.625000	.125000	48	F	0	0	33
265	=	264.000	.750000	.750000	.000000	12	F	0	0	33
266	=	265.000	.875000	.000000	.000000	6	F	0	0	33
267	=	266.000	.875000	.125000	.000000	24	F	0	0	33
268	=	267.000	.875000	.125000	.125000	24	F	0	0	33
269	=	268.000	.875000	.250000	.000000	24	F	0	0	33
270	=	269.000	.875000	.250000	.125000	48	F	0	0	33
271	=	270.000	.875000	.250000	.250000	24	F	0	0	33
272	=	271.000	.875000	.375000	.000000	24	F	0	0	33
273	=	272.000	.875000	.375000	.125000	48	F	0	0	33
274	=	273.000	.875000	.375000	.250000	48	F	0	0	33
275	=	274.000	.875000	.500000	.000000	24	F	0	0	33
276	=	275.000	.875000	.500000	.125000	48	F	0	0	33
277	=	276.000	.875000	.625000	.000000	24	F	0	0	33
278	=	277.000	1.000000	.000000	.000000	3	F	0	0	33
279	=	278.000	1.000000	.125000	.000000	12	F	0	0	33
280	=	279.000	1.000000	.125000	.125000	12	F	0	0	33
281	=	280.000	1.000000	.250000	.000000	12	F	0	0	33
282	=	281.000	1.000000	.250000	.125000	24	F	0	0	33
283	=	282.000	1.000000	.250000	.250000	12	F	0	0	33
284	=	283.000	1.000000	.375000	.000000	12	F	0	0	33
285	=	284.000	1.000000	.375000	.125000	24	F	0	0	33
286	=	285.000	1.000000	.500000	.000000	12	F	0	0	33

## IX-3. LIST OF PROGRAMS

13:58 APR 08, '76 ID=060C-H01

JOB MKL2427,HALL,1

PCL

COPY HCOMPLX1 TO LP(K)

```

1 - 1.000 C!ASSIGN M:BM, (FILE,MMAIND), (SAVE)
2 - 2.000 C!FORTRAN LS,BD,ADP
3 - 3.000 C MAIN PROGRAM FOR BAND CALCULATIONS OF CLOSED SHELL IONIC SOLIDS
4 - 4.000 C OR NOBEL GAS SOLIDS
5 - 5.000 C D. J. MICKISH, UNIVERSITY OF ILLINOIS, SPRING 1972
6 - 5.500 C MODIFIED BY K. M. HALL TO INCLUDE OPEN SHELL SYSTEMS VIA
7 - 5.600 C FRACTIONAL OCCUPATION APPROXIMATION
8 - 6.000 C METHOD USED IS LCAO WITH LOCALIZED ORBITALS TO 1ST ORDER IN
9 - 7.000 C OVERLAP)
10 - 8.000 C COMPLEX CA,CX
11 - 9.000 C COMMON ALPHA(161,31)
12 - 10.000 C COMMON RANL(1449),DRANL(1449),RAMESH(161),VNCAFN(161),VCZAFN(161)
13 - 11.000 C COMMON RXNL(1449),DRXNL(1449),RXMESH(161),VNCXFN(161),VCZXFN(161)
14 - 12.000 C DIMENSION CA(19,5),L0CAFN(3),LTOAFN(3),LADINX(4),LAMINX(4),
15 - 13.000 C 1 NA(19),LA(19),FRA(3,15),FRX(3,15),NAF(3),NAQ(3),
16 - 13.500 C 2 XAF(3,25),XAG(3,25),FRASP(3,15),FRXSP(3,15)
17 - 14.000 C DIMENSION CX(15,5),L0CXFN(3),LTOCFN(3),LXDINX(4),LXMIX(4),
18 - 15.000 C 1 NX(15),LX(15),NXF(3),NXG(3),XXF(3,25),XXG(3,25)
19 - 16.000 C LOGICAL LLDTAL,LLAMA,LLNABA,LLABX,LLNABX,LLGENA,LLCKAL,LLGENI,
20 - 17.000 C 1 LLBUF,LLETE,LLFRST,LLMTRX
21 - 18.000 C DIMENSION KLETE(34)
22 - 19.000 C DIMENSION RRANL(1),RRXNL(1)
23 - 20.000 C EQUIVALENCE (ALPHA(1,1),RRANL(1),RRXNL(1))
24 - 21.000 C COMMON/CRY/ RU(258),RUB(240),AADIST(6),AXDIST(6),
25 - 22.000 C 1 NAASET(6),NAXSET(6)
26 - 23.000 C DIMENSION IRU(258),IRUB(240)
27 - 24.000 C DATA NAASET/12,6,24,12,24,8/
28 - 25.000 C INTEGER RU110(36),RU200(18),RU211(72),RU220(36),RU310(72),
29 - 26.000 C 1 RU222(24),FA(3,25),GA(3,25,2),FX(3,25),GX(3,25,2)
30 - 27.000 C DATA RU110/ 2, 2, 0, 2, -2, 0, -2, 2, 0, -2, -2, 0, 2, 0, 2,
31 - 28.000 C 1 2, 0, -2, -2, 0, 2, -2, 0, -2, 0, 2, 2, 0, 2, -2,
32 - 29.000 C 2 0, -2, 2, 0, -2, -2/
33 - 30.000 C DATA RU200/ 4, 0, 0, -4, 0, 0, 0, 4, 0, 0, -4, 0, 0, 0, 4,
34 - 31.000 C 1 0, 0, -4/
35 - 32.000 C DATA RU211/ 4, 2, 2, 4, 2, -2, 4, -2, 2, 4, -2, -2, 2, 4, 2,
36 - 33.000 C 1 2, 4, -2, -2, 4, 2, -2, 4, -2, 2, 2, 4, 2, 4, -2, 4,
37 - 34.000 C 2 -2, 2, 4, -2, -2, 4, -4, 2, 2, -4, 2, -2, -4, 2, 2,
38 - 35.000 C 3 -4, -2, -2, 2, -4, 2, 2, -4, -2, -2, -4, 2, -2, -4, -2,
39 - 36.000 C 4 2, 2, -4, 2, -2, -4, -2, 2, -4, -2, -2, -4/
40 - 37.000 C DATA RU220/ 4, 4, 0, 4, -4, 0, -4, 4, 0, -4, -4, 0, 4, 0, 4,
41 - 38.000 C 1 4, 0, -4, -4, 0, 4, -4, 0, -4, 0, 4, 4, 0, 4, -4,
42 - 39.000 C 2 0, -4, 4, 0, -4, -4/
43 - 40.000 C DATA RU310/ 6, 2, 0, 6, -2, 0, -6, 2, 0, -6, -2, 0, 6, 0, 6,
44 - 41.000 C 1 6, 0, -2, -6, 0, 2, -6, 0, -2, 0, 6, 2, 0, 6, -2,
45 - 42.000 C 2 0, -6, 2, 0, -6, -2, 2, 6, 0, 2, -6, 0, -2, 6, 0,
46 - 43.000 C 3 -2, -6, 0, 2, 0, 6, 2, 0, -6, -2, 0, 6, -2, 0, -6,
47 - 44.000 C 4 0, 2, 6, 0, 2, -6, 0, -2, 6, 0, -2, 6/
48 - 45.000 C DATA RU222/ 4, 4, 4, 4, 4, -4, 4, -4, 4, -4, 4, 4, -4, 4, 4,
49 - 46.000 C 1 -4, 4, -4, 4, -4, 4, -4, -4, -4/
50 - 47.000 C EQUIVALENCE (RU110(1),IRU(1)),(RU200(1),IRU(37)),
51 - 48.000 C 1 (RU211(1),IRU(55)),(RU220(1),IRU(127)),
52 - 49.000 C 2 (RU310(1),IRU(163)),(RU222(1),IRU(235))
53 - 50.000 C DATA NAXSET/4,12,12,16,24,12/
54 - 51.000 C INTEGER RUB100(12),RUB111(36),RUB210(36),RUB221(48),RUB311(72),
55 - 52.000 C 1 RUB320(36)

```

```

56 - 53.000      DATA RUB100/ 1, 1, 1, 1,-1,-1, -1, 1,-1, -1,-1, 1/
57 - 54.000      DATA RUB111/-3,-1,-1, -3, 1, 1, 3,-1, 1, 3, 1,-1, -1,-3,-1/
58 - 55.000      1      -1, 3, 1, 1,-3, 1, 1, 3,-1, -1,-1,-3, -1, 1, 3/
59 - 56.000      2      1,-1, 3, 1, 1,-3/
60 - 57.000      DATA RUB210/ 3, 3, 1, 3,-3,-1, -3, 3,-1, -3,-3, 1, 1, 3, 3/
61 - 58.000      1      1,-3,-3, -1, 3,-3, -1,-3, 3, 3, 1, 3, 3,-1,-3/
62 - 59.000      2      -3, 1,-3, -3,-1, 3/
63 - 60.000      DATA RUB221/ 5, 1, 1, 5,-1,-1, -5,-1, 1, -5, 1,-1, 1, 5, 1/
64 - 61.000      1      -1, 5,-1, -1,-5, 1, 1,-5,-1, 1, 1, 5, -1,-1, 5/
65 - 62.000      2      -1, 1,-5, 1,-1,-5, -3,-3,-3, 3, 3,-3, 3,-3, 3/
66 - 63.000      3      -3, 3, 3/
67 - 64.000      DATA RUB311/-5,-3,-1, 5, 3,-1, -5, 3, 1, 5,-3, 1, -1,-3,-5/
68 - 65.000      1      1, 3,-5, 1,-3, 5, -1, 3, 5, -1,-5,-3, 1, 5,-3/
69 - 66.000      2      -1, 5, 3, 1,-5, 3, -5,-1,-3, 5, 1,-3, 5,-1, 3/
70 - 67.000      3      -5, 1, 3, -3,-5,-1, 3, 5,-1, -3, 5, 1, 3,-5, 1/
71 - 68.000      4      -3,-1,-5, 3, 1,-5, -3, 1, 5, 3,-1, 5/
72 - 69.000      DATA RUB320/ 5, 3, 3, 5,-3,-3, -5, 3,-3, -5,-3, 3, 3, 5, 3/
73 - 70.000      1      3,-5,-3, -3, 5,-3, -3,-5, 3, 3, 3, 5, 3,-3,-5/
74 - 71.000      2      -3, 3,-5, -3,-3, 5/
75 - 79.000      EQUIVALENCE (RUB100(1),IRUB(1)),(RUB111(1),IRUB(13)),
76 - 80.000      1      (RUB210(1),IRUB(49)),(RUB221(1),IRUB(85)),
77 - 81.000      2      (RUB311(1),IRUB(133)),(RUB320(1),IRUB(205))
78 - 82.000      REAL*8 ZAADIS,ZAXDIS
79 - 83.000      DATA ZAADIS/'AADIST'//,ZAXDIS/'AXDIST'/
80 - 84.000      REAL*8 ZVNCA,ZVNCX,ZLARNX,ZLADNX,ZLXRNX,ZLXDNX
81 - 85.000      DATA ZLARNX/'LARINX'//,ZLADNX/'LADINX'//,ZLXRNX/'LXRINX'//
82 - 86.000      1      ZLXDNX/'LXDINX'//
83 - 87.000      DATA LLFRST/.TRUE./
84 - 88.000 C      DIMENSION CA(MBLCHA,2*LAMAX+1)
85 - 89.000 C      CX(MBLCHX,2*LXMAX+1)
86 - 90.000 C      RANL,DRANL(MXAFN*MXPTS)
87 - 91.000 C      RXNL,DRXNL(MXXFN*MXPTS)
88 - 92.000 C      RAMESH,RXMesh,VNCAFN,VCZAFN,VNCFN,VCZXFN(MXPTS)
89 - 93.000 C      LTOAFN AND LBOCFN(LAMAX+1)
90 - 94.000 C      LTOXFN AND LBOCFN(LXMAX+1)
91 - 95.000 C      LARINX AND LADINX(LAMAX+1)
92 - 96.000 C      LXRINX AND LXDINX(LXMAX+1)
93 - 97.000 C      NA AND LA(MBLCHA)
94 - 98.000 C      NX AND LX(MBLCHX)
95 - 99.000 C      ALPHA(MXPTS,MXPAN+1)
96 - 100.000 C     KLETE(MBLCHA+MBLCHX)
97 - 101.000 C     NAASET,NAXSET,AADIST,AXDIST(NSETS)
98 - 102.000      MXAFN=9
99 - 103.000      MXXFN=9
100 - 104.000     LAMAX=3
101 - 105.000     LXMAX=3
102 - 106.000     MBLCHA=19
103 - 107.000     MBLCHX=15
104 - 108.000     MXPTS=161
105 - 109.000     MXPAN=30
106 - 110.000     NSETS=6
107 - 111.000 C     FOR NAEL LATTICE
108 - 112.000     DATA AMDLNG/1,7+7565/
109 - 113.000     IALPTS=MXPAN*MXPTS
110 - 114.000     EPS=5.0E-05
111 - 115.000 C
112 - 116.000 C     DEFINE FILE 26(151,4830,U,JAAUNM)
113 - 117.000     IAAFL=26
114 - 118.000 C     NUMBER OF RECORDS = NSETS*LADINX(LAMAX+2)+1 = 6*25+1 = 151
115 - 119.000 C     THIS ASSUMES THE LARGEST NUMBER OF FUNCTIONS WITH A

```

```

116 = 120.000 C          GIVEN Lj STORED IN LTOAFN, ARE (6,5,3)
117 = 121.000 C          MAXIMUM RECORD SIZE (WORDS) = MXPTS*MXPAN = 161*30 = 4830
118 = 122.000 C          NOTE ONLY ONE RECORD PER TRACK IF 880 TO 1823 WORDS PER REC.
119 = 123.000 C          U INDICATES UNFORMATTED I/O
120 = 124.000 C          JAAUNM IS THE ASSOCIATED VARIABLE
121 = 125.000 C
122 = 126.000 C          DEFINE FILE 27( 79,4830,U,JXXUNM)
123 = 127.000 C          IXFFLE=27
124 = 128.000 C          NUMBER OF RECORDS = NSETS*LXDINX(LXMAX+2)+1 = 6*13+1 = 79
125 = 129.000 C          THIS ASSUMES THE LARGEST NUMBER OF FUNCTIONS WITH A
126 = 130.000 C          GIVEN Lj STORED IN LTOAFN, ARE (4,3,1)
127 = 131.000 C          MAXIMUM RECORD SIZE (WORDS) = MXPTS*MXPAN = 161*30 = 4830
128 = 132.000 C          NOTE ONLY ONE RECORD PER TRACK IF 880 TO 1823 WORDS PER REC.
129 = 133.000 C          U INDICATES UNFORMATTED I/O
130 = 134.000 C          JXXUNM IS THE ASSOCIATED VARIABLE
131 = 135.000 C
132 = 136.000 C          DEFINE FILE 28(150,4830,U,JAXUNM)
133 = 137.000 C          IAXFLE=28
134 = 138.000 C          NUMBER OF RECORDS = NSETS*LADINX(LAMAX+2) = 6*25 = 150
135 = 139.000 C          THIS ASSUMES THE LARGEST NUMBER OF FUNCTIONS WITH A
136 = 140.000 C          GIVEN Lj STORED IN LTOAFN, ARE (6,5,3)
137 = 141.000 C          MAXIMUM RECORD SIZE (WORDS) = MXPTS*MXPAN = 161*30 = 4830
138 = 142.000 C          NOTE ONLY ONE RECORD PER TRACK IF 880 TO 1823 WORDS PER REC.
139 = 143.000 C          U INDICATES UNFORMATTED I/O
140 = 144.000 C          JAXUNM IS THE ASSOCIATED VARIABLE
141 = 145.000 C
142 = 146.000 C          DEFINE FILE 29( 78,4830,U,JXAUNM)
143 = 147.000 C          IXAFLE=29
144 = 148.000 C          NUMBER OF RECORDS = NSETS*LXDINX(LXHAX+2) = 6*13 = 78
145 = 149.000 C          THIS ASSUMES THE LARGEST NUMBER OF FUNCTIONS WITH A
146 = 150.000 C          GIVEN Lj STORED IN LTOAFN, ARE (4,3,1)
147 = 151.000 C          MAXIMUM RECORD SIZE (WORDS) = MXPTS*MXPAN = 161*30 = 4830
148 = 152.000 C          NOTE ONLY ONE RECORD PER TRACK IF 880 TO 1823 WORDS PER REC.
149 = 153.000 C          U INDICATES UNFORMATTED I/O
150 = 154.000 C          JXAUNM IS THE ASSOCIATED VARIABLE
151 = 155.000 C
152 = 156.000 C          IIFILE=30
153 = 157.000 C          IEVFLE=31
154 = 157.500 C          FILE 13 NEEDED ALSO FOR COMPLEX DIAGONALIZATION ROUTINES (ROUTIN)
155 = 158.000 C
156 = 159.000 C          INPUT ITME
157 = 160.000 C          CALL CHCKPT('HCKPTFILE',ITME)
158 = 161.000 CCCCC CALL ELAPSE(IMINU)
159 = 162.000 CCCCC TMINU=0.0
160 = 163.000 C          WRITE(6,899)
161 = 164.000 C          899 FORMAT(' 1,30X'ELECTRONIC BAND STRUCTURE OF CLOSED SHELL SOLIDS'/
162 = 165.000 C          1      ' 1,30X'  HARTREE-FOCK, LCAO, LOCALIZED ORBITALS'
163 = 166.000 C          2      ' 1,30X'  PROGRAMED BY D. J. MICKISH'/
164 = 167.000 C          3      ' 1,30X'  PHYSICS DEPARTMENT'/
165 = 168.000 C          4      ' 1,30X'  UNIVERSITY OF ILLINOIS'
166 = 169.000 C          5      ' 1,30X'  URBANA, ILLINOIS 61801'//)
167 = 170.000 C          READ(5,900) LLDOTAL,NROUT,LLMTRX
168 = 171.000 C          900 FORMAT(10X,L5,I5,L5)
169 = 172.000 C          WRITE(6,901) LLDOTAL,NROUT,LLMTRX
170 = 173.000 C          901 FORMAT('////' 1,LLDOTAL =',L2,' NROUT =',I2,' LLMTRX =',L2)
171 = 174.000 CCCCC *****
172 = 175.000 C          SITES *****
173 = 176.000 CCCCC *****
174 = 177.000 C          1 READ(5,902) LLABA,NEIGHA,LLAOX,NEIGHX
175 = 178.000 C          902 FORMAT(2(L5,I5))

```

```

176 - 179.000 C   NOTE THAT ALL AVAILABLE SETS OF NEIGHBORS WILL BE USED IN MION
177 - 180.000 C   SUBROUTINE
178 - 181.000     LLNABA=.NOT.LLABA
179 - 182.000     LLNABX=.NOT.LLABX
180 - 183.000     WRITE(6,903) LLABA,NEIGHA,LLABX,NEIGHX
181 - 184.000     903 FORMAT(' ',LLABA='L2', NEIGHA=' ',I3,' ' LLABX='L2,
182 - 185.000     1     NEIGHX=' ',I3)
183 - 186.000     IF (LLNABA.AND.LLNABX) CALL EXIT
184 - 187.000 CCCCC *****
185 - 188.000 C   A0'S
186 - 189.000 CCCCC *****
187 - 190.000     2 ZA=0.0
188 - 191.000     ZX=0.0
189 - 192.000     NAEL=0
190 - 193.000     NXEL=0
191 - 194.000     LAHI1=1
192 - 195.000     LXHI1=1
193 - 196.000     IF (LLNABA) GO TO 4
194 - 197.000     IF (NEIGHA.GE.1.AND.NEIGHA.LE.NSETS) GO TO 3
195 - 198.000     WRITE(6,904) NEIGHA,NSETS
196 - 199.000     904 FORMAT(' ',NEIGHA=' ',I3,' ' NSETS=' ',I3)
197 - 200.000     CALL EXIT
198 - 201.000     3 WRITE(6,905)
199 - 202.000     905 FORMAT(' ', 'INPUT DATA FOR CENTER !!!' FOLLOWS')
200 - 203.000     CALL WAVEFN(ZA,ZAION,ZAR,RHOA,HA,IAPTS,RAMESH,RANL,QRANL,MXAFN,
201 - 204.000     1     MXPTS,LAMAX1,LOCAFN,LT0AFN,LAHI1,NAEL,FBA,FRASH)
202 - 205.000     NTOAFN=0
203 - 206.000     DO 30001 L1=1,LAHI1
204 - 207.000     30001 NTOAFN=NTOAFN+LT0AFN(L1)
205 - 208.000     DO 30002 I=1,IAPTS
206 - 209.000     RR=RAMESH(I)**2
207 - 210.000     INXFN=I
208 - 211.000     DO 30002 NN=1,NTOAFN
209 - 212.000     RRANL(INXFN)=RANL(INXFN)**2+RR
210 - 213.000     30002 INXFN=INXFN+IAPTS
211 - 214.000     DO 30005 NN=1,NTOAFN,8
212 - 215.000     NMAX=NN+7
213 - 216.000     IF (NMAX.GT.NTOAFN) NMAX=NTOAFN
214 - 217.000     WRITE(6,9053) (N,N=N,NMAX)
215 - 218.000     9053 FORMAT('O', ' I',5X'R MESH(I)',2X8(3X'2', ' ORBITAL'))
216 - 219.000     DO 30005 I=1,IAPTS
217 - 220.000     30005 WRITE(6,9057) I,RAMESH(I),(RRANL((N#1)+IAPTS+I),N=NN,NMAX)
218 - 221.000     9057 FORMAT(' ',I3,F12.6,2X8F13.6)
219 - 222.000     4 IF (LLNABX) GO TO 6
220 - 223.000     IF (NEIGHX.GE.1.AND.NEIGHX.LE.NSETS) GO TO 5
221 - 224.000     WRITE(6,906) NEIGHX,NSETS
222 - 225.000     906 FORMAT(' ',NEIGHX=' ',I3,' ' NSETS=' ',I3)
223 - 226.000     CALL EXIT
224 - 227.000     5 WRITE(6,907)
225 - 228.000     907 FORMAT(' ', 'INPUT DATA FOR CENTER !!!' FOLLOWS')
226 - 229.000     CALL WAVEFN(ZX,ZXION,ZXR,RHOX,HX,IXPTS,RXMESH,RXNL,QRXNL,MXRFN,
227 - 230.000     1     MXPTS,LXMAX1,LOCFN,LT0XFN,LXHI1,NXEL,FBX,FRXS)
228 - 231.000     NTOXFN=0
229 - 232.000     DO 50001 L1=1,LXHI1
230 - 233.000     50001 NTOXFN=NTOXFN+LT0XFN(L1)
231 - 234.000     DO 50002 I=1,IXPTS
232 - 235.000     RR=RXMESH(I)**2
233 - 236.000     INXFN=I
234 - 237.000     DO 50002 NN=1,NTOXFN
235 - 238.000     RRXNL(INXFN)=RXNL(INXFN)**2+RR

```

```

236 - 239.000 50002 INXFN=INXFN+IXPTS
237 - 240.000      DO 50005 NN=1,NT0XFN,8
238 - 241.000      NMAX=NN+7
239 - 242.000      IF (NMAX.GT.NT0XFN) NMAX=NT0XFN
240 - 243.000      WRITE(6,9053) (N,N=NN,NMAX)
241 - 244.000      DO 50005 I=1,IXPTS
242 - 245.000 50005 WRITE(6,9057) I,RXMESH(I),(RRXNL((N=1)+IXPTS+I),N=NN,NMAX)
243 - 246.000      6 LMAX1=MAX0(LAHI1,LXHI1)
244 - 247.000      IF (FLOAT(NAEL+NXEL).EQ.ZA+ZX) GO TO 60005
245 - 248.000      WRITE(6,908) NAEL,NXEL,ZA,ZX
246 - 249.000 908  FORMAT(' ','SYSTEM DOES NOT HAVE NEUTRAL CHARGE  NAEL =',I3,
247 - 250.000      1      ' NXEL =',I3,' ZA =',F6.1,' ZX =',F6.1)
248 - 251.000      CALL EXIT
249 - 252.000      CCCCC CALL ELAPSE(IMINU) (LABEL IS 60005)
250 - 252.500 60005 CONTINUE
251 - 253.000 CCCCC FMINU=FLOAT(IMINU)*1.0E-3
252 - 254.000 CCCCC TMINU=TMINU+FMINU
253 - 255.000 CCCCC WRITE(6,9095) FMINU,TMINU
254 - 256.000 9095  FORMAT('0','INTERVAL =',F6.2,' MIN',10X'TIME =',F6.2,' MIN',4)
255 - 257.000 CCCCC *****
256 - 258.000 C      AO NXPAN
257 - 259.000 CCCCC *****
258 - 260.000      READ(5,909) AO,UNIT,NXPAN,JTHPTS,LLGENA,LLCKAL,LLBUF
259 - 261.000 909  FORMAT(2F10.0,2I5,3L5)
260 - 262.000 C      VECTORS RU=R(U) AND RUB=R(U)+B ARE STORED IN UNITS OF UNIT*AO
261 - 263.000      WRITE(6,910) AO,UNIT,NXPAN,JTHPTS,LLGENA,LLCKAL,LLBUF
262 - 264.000 910  FORMAT('1','AO =',F7.4,' UNIT =',F7.4,' NXPAN =',I3,
263 - 265.000      1      ' JTHPTS =',I4,' LLGENA =',L2,' LLCKAL =',L2,
264 - 266.000      2      ' LLBUF =',L2)
265 - 267.000 C      CALCULATE NEIGHBOR DISTANCES
266 - 268.000      IF (LLN00A) GO TO 12
267 - 269.000      WRITE(6,9105)
268 - 270.000 9105  FORMAT('0','AA NEIGHBORS IN UNITS OF UNIT*AO')
269 - 271.000      MUINX=0
270 - 272.000      DO 10 MUSET=1,NSETS
271 - 273.000      MULIM=NAASET(MUSET)
272 - 274.000      WRITE(6,911)
273 - 275.000 911  FORMAT(' ')
274 - 276.000      DO 9 MU=1,MULIM
275 - 277.000      WRITE(6,912) IRU(MUINX+1),IRU(MUINX+2),IRU(MUINX+3)
276 - 278.000 912  FORMAT(' ',3I4)
277 - 279.000      RU(MUINX+1)=FLOAT(IRU(MUINX+1))
278 - 280.000      RU(MUINX+2)=FLOAT(IRU(MUINX+2))
279 - 281.000      RU(MUINX+3)=FLOAT(IRU(MUINX+3))
280 - 282.000      9  MUINX=MUINX+3
281 - 283.000 10  AADIST(MUSET)=SQRT(RU(MUINX-2)**2+RU(MUINX-1)**2+RU(MUINX)**2)
282 - 284.000      1      *UNIT*AO
283 - 285.000      CALL SUB1(AADIST,NSETS,ZAADIS)
284 - 286.000 12  IF (LLN00X) GO TO 16
285 - 287.000      WRITE(6,913)
286 - 288.000 913  FORMAT('0','AX NEIGHBORS IN UNITS OF UNIT*AO')
287 - 289.000      MUINX=0
288 - 290.000      DO 14 MUSET=1,NSETS
289 - 291.000      MULIM=NAXSET(MUSET)
290 - 292.000      WRITE(6,911)
291 - 293.000      DO 13 MU=1,MULIM
292 - 294.000      WRITE(6,912) IRUB(MUINX+1),IRUB(MUINX+2),IRUB(MUINX+3)
293 - 295.000      RUB(MUINX+1)=FLOAT(IRUB(MUINX+1))
294 - 296.000      RUB(MUINX+2)=FLOAT(IRUB(MUINX+2))
295 - 297.000      RUB(MUINX+3)=FLOAT(IRUB(MUINX+3))

```

```

296 = 298.000 13 MUINX=MUINX+3
297 = 299.000 14 AXDIST(MUSET)=SQRT(RUB(MUINX-2)**2+RUB(MUINX-1)**2+RUB(MUINX)**2)
298 = 300.000 1
299 = 301.000 CALL IOUT1(AXDIST,NSETS,ZAXDIS)
300 = 302.000 C CALCULATE PARTIAL INDICES FOR DIRECT ACCESS OF ALPHA FUNCTIONS
301 = 303.000 16 IF (LLNADA) GO TO 18
302 = 304.000 LADINX(1)=0
303 = 305.000 LARINX(1)=0
304 = 306.000 DO 17 L1=1,LAMAX1
305 = 307.000 LARINX(L1+1)=LARINX(L1)+LT0AFN(L1)
306 = 308.000 17 LADINX(L1+1)=LADINX(L1)+L1*LT0AFN(L1)
307 = 309.000 IF (LLDTAL) CALL IOUT1(LARINX,LAMAX1+1,ZLARNX)
308 = 310.000 IF (LLDTAL) CALL IOUT1(LADINX,LAMAX1+1,ZLADNX)
309 = 311.000 NREC=NEIGHA*LADINX(LAMAX1+1)+1
310 = 312.000 IF (LLGENA.OR.LLCKAL)
311 = 313.000 1 CALL CKALPH(IAAFLE,NREC,ALPHA,MXPTS,MXPAN,LLGENA)
312 = 314.000 18 IF (LLNADX) GO TO 20
313 = 315.000 LXDINX(1)=0
314 = 316.000 LXRINX(1)=0
315 = 317.000 DO 19 L1=1,LXMAX1
316 = 318.000 LXRINX(L1+1)=LXRINX(L1)+LT0XFN(L1)
317 = 319.000 19 LXDINX(L1+1)=LXDINX(L1)+L1*LT0XFN(L1)
318 = 320.000 IF (LLDTAL) CALL IOUT1(LXRINX,LXMAX1+1,ZLXRNX)
319 = 321.000 IF (LLDTAL) CALL IOUT1(LXDINX,LXMAX1+1,ZLXDNX)
320 = 322.000 NREC=NEIGHX*LXDINX(LXMAX1+1)+1
321 = 323.000 IF (LLGENA.OR.LLCKAL)
322 = 324.000 1 CALL CKALPH(IXFLE,NREC,ALPHA,MXPTS,MXPAN,LLGENA)
323 = 325.000 20 IF (LLNADA.OR.LLNADX) GO TO 2005
324 = 326.000 NREC=NEIGHX*LADINX(LAMAX1+1)
325 = 327.000 IF (LLGENA.OR.LLCKAL)
326 = 328.000 1 CALL CKALPH(IXAFLE,NREC,ALPHA,MXPTS,MXPAN,LLGENA)
327 = 329.000 NREC=NEIGHX*LXDINX(LXMAX1+1)
328 = 330.000 IF (LLGENA.OR.LLCKAL)
329 = 331.000 1 CALL CKALPH(IXAFLE,NREC,ALPHA,MXPTS,MXPAN,LLGENA)
330 = 332.000 2005 DO 2008 I=1,MXPTS
331 = 333.000 VCZAFN(I)=0.0
332 = 334.000 VCZXFN(I)=0.0
333 = 335.000 VNCAFN(I)=0.0
334 = 336.000 2008 VNCXFN(I)=0.0
335 = 337.000 IF (.NOT.LLGENA) GO TO 36
336 = 338.000 IF (LLABA) CALL VNC(RANL,LAMAX1,L0CAFN,LTOAFN,RAMESH,IAPTS,
337 = 339.000 1 VNCAFN,FRA)
338 = 340.000 IF (LLAOX) CALL VNC(RXNL,LXMAX1,L0CXFN,LTOXFN,RXMEBH,IXRPTS,
339 = 341.000 1 VNCXFN,FRX)
340 = 342.000 WRITE(6,9132) (I,RAMESH(I),VNCAFN(I),
341 = 343.000 1 RXMESH(I),VNCXFN(I),I=1,MXPTS)
342 = 344.000 9132 FORMAT(' ',3X'I',9X'RAMESH',4X'ATOMIC COUL A',
343 = 345.000 1 10X'RXMESH',4X'ATOMIC COUL X',
344 = 346.000 2 (' ',I4,4X'P2E14.5',5X'2E14.5')
345 = 347.000 C EXPAND 'A' ORBITALS ABOUT SITE 'A'
346 = 348.000 C EXPAND 'X' ORBITALS ABOUT SITE 'X'
347 = 349.000 IAAUNM=1
348 = 350.000 IXXUNM=1
349 = 351.000 DO 29 MUSET=1,NEIGHA
350 = 352.000 AA=AADIST(MUSET)
351 = 353.000 AABORS=FL0AT(NAASET(MUSET))
352 = 354.000 IF (LLNADA) GO TO 25
353 = 355.000 CALL EXPAND(0,LMAX1,VNCAFN,0,RAMESH,MXPTS,IAPTS,AA,
354 = 356.000 1 ALPHA,MXPAN,1,RAMESH,ZAR,RHOA,HA,IAPTS,UTPTS,
355 = 357.000 2 ,FALSE,,0,0,LLDTAL,LL0UF)

```

```

356 - 358.000      DB 2105 I=1, IAPTS
357 - 359.000      2105 VCZAFN(I)=VCZAFN(I)+AABORS*ALPHA(I,1)
358 - 360.000      INXFN=0
359 - 361.000      DB 23 L1=1, LAHI1
360 - 362.000      L=L1-1
361 - 363.000      LTDFN=LTDFN(L1)
362 - 364.000      IF (LTDFN.EQ.0) GO TO 23
363 - 365.000      DB 22 J=1, LTDFN
364 - 366.000      CALL EXPAND(L, LMAX1, RANL, INXFN, RAMESH, MXPTS, IAPTS, AA,
365 - 367.000      1          ALPHA, MXPAN, NXPAN, RAMESH, ZAR, RH0A, HA, IAPTS, JTHPTS,
366 - 368.000      2          .TRUE., IAAFL, IAAUNM, LLDAL, LLBUF)
367 - 369.000      22 INXFN=INXFN+IAPTS
368 - 370.000      23 CONTINUE
369 - 371.000      25 IF (LLNABX) GO TO 29
370 - 372.000      CALL EXPAND(0, LMAX1, VNCXFN, 0, RXMESH, MXPTS, IXPTS, AA,
371 - 373.000      1          ALPHA, MXPAN, 1, RXMESH, ZXR, RH0X, HX, IXPTS, JTHPTS,
372 - 374.000      2          .FALSE., 0, 0, LLDAL, LLBUF)
373 - 375.000      DB 2505 I=1, IXPTS
374 - 376.000      2505 VCZXFN(I)=VCZXFN(I)+AABORS*ALPHA(I,1)
375 - 377.000      INXFN=0
376 - 378.000      DB 27 L1=1, LXHI1
377 - 379.000      L=L1-1
378 - 380.000      LTDFN=LTDFN(L1)
379 - 381.000      IF (LTDFN.EQ.0) GO TO 27
380 - 382.000      DB 26 J=1, LTDFN
381 - 383.000      CALL EXPAND(L, LMAX1, RXNL, INXFN, RXMESH, MXPTS, IXPTS, AA,
382 - 384.000      1          ALPHA, MXPAN, NXPAN, RXMESH, ZXR, RH0X, HX, IXPTS, JTHPTS,
383 - 385.000      2          .TRUE., IXXFL, IXXUNM, LLDAL, LLBUF)
384 - 386.000      26 INXFN=INXFN+IXPTS
385 - 387.000      27 CONTINUE
386 - 388.000      29 CONTINUE
387 - 389.000      C   EXPAND 'A' ORBITALS ABOUT SITE 'X'
388 - 390.000      C   EXPAND 'X' ORBITALS ABOUT SITE 'A'
389 - 391.000      IF (LLNABA.OR.LLNABX) GO TO 35
390 - 392.000      IAXUNM=1
391 - 393.000      IAXAUNM=1
392 - 394.000      DB 34 MUSET=1, NEIGHX
393 - 395.000      AX=AXDIST(MUSET)
394 - 396.000      AXBORS=FL0AT(NAXSET(MUSET))
395 - 397.000      CALL EXPAND(0, LMAX1, VNCAFN, 0, RXMESH, MXPTS, IXPTS, AX,
396 - 398.000      1          ALPHA, MXPAN, 1, RAMESH, ZAR, RH0A, HA, IAPTS, JTHPTS,
397 - 399.000      2          .FALSE., 0, 0, LLDAL, LLBUF)
398 - 400.000      DB 2901 I=1, IXPTS
399 - 401.000      2901 VCZXFN(I)=VCZXFN(I)+AXBORS*ALPHA(I,1)
400 - 402.000      INXFN=0
401 - 403.000      DB 2904 L1=1, LAHI1
402 - 404.000      L=L1-1
403 - 405.000      LTDFN=LTDFN(L1)
404 - 406.000      IF (LTDFN.EQ.0) GO TO 2904
405 - 407.000      DB 2902 J=1, LTDFN
406 - 408.000      CALL EXPAND(L, LMAX1, RANL, INXFN, RXMESH, MXPTS, IXPTS, AX,
407 - 409.000      1          ALPHA, MXPAN, NXPAN, RAMESH, ZAR, RH0A, HA, IAPTS, JTHPTS,
408 - 410.000      2          .TRUE., IAXFL, IAXUNM, LLDAL, LLBUF)
409 - 411.000      2902 INXFN=INXFN+IAPTS
410 - 412.000      2904 CONTINUE
411 - 413.000      CALL EXPAND(0, LMAX1, VNCXFN, 0, RAMESH, MXPTS, IAPTS, AX,
412 - 414.000      1          ALPHA, MXPAN, 1, RXMESH, ZXR, RH0X, HX, IXPTS, JTHPTS,
413 - 415.000      2          .FALSE., 0, 0, LLDAL, LLBUF)
414 - 416.000      DB 2906 I=1, IAPTS
415 - 417.000      2906 VCZAFN(I)=VCZAFN(I)+AXBORS*ALPHA(I,1)

```

```

416 - 418.000      INXFN=0
417 - 419.000      DB 31 L1=1,LXHI1
418 - 420.000      LT0FN=LT0XFN(L1)
419 - 421.000      IF (LT0FN.EQ.0) GO TO 31
420 - 422.000      L=L1-1
421 - 423.000      DB 30 J=1,LT0FN
422 - 424.000      CALL EXPAND(L,LMAX1,RXNL,INXFN,RAMESH,MXPTS,IAPTS,AX,
423 - 425.000      1      ALPHA,MXPAN,NXPAN,RXMESH,ZXR,RHOX,HX,IXPTS,UTHRIS,
424 - 426.000      2      ,TRUE,,IXAFLE,IXAUNM,LLDTAL,LLBUF)
425 - 427.000      30 INXFN=INXFN+IXPTS
426 - 428.000      31 CONTINUE
427 - 429.000      34 CONTINUE
428 - 430.000      35 IF (LLDTAL) WRITE(6,9134) (I,RAMESH(I),VCZAFN(I),RXMESH(I),
429 - 431.000      1      VCZXFN(I),I=1,MXPTS)
430 - 432.000      9134 FORMAT('0',3X'I',9X'RAMESH',5X'COUL ENVIOR A',
431 - 433.000      1      9X'RXMESH',5X'COUL ENVIOR X'/
432 - 434.000      2      (' ',I4,4X1P2E14.5,5X2E14.5)
433 - 435.000      DB 3504 I=1,MXPTS
434 - 436.000      VNCAFN(I)=0.0
435 - 437.000      3504 VNCXFN(I)=0.0
436 - 438.000      IF (ZAI0N.EQ.0.0) GO TO 37
437 - 439.000      CALL PTION(ZAI0N,AMDNG,AXDIST,NAXSET,AADIST,NAASET,NSETS,
438 - 440.000      1      VNCAFN,RAMESH,IAPTS)
439 - 441.000      CALL PTION(ZAI0N,AMDNG,AXDIST,NAXSET,AADIST,NAASET,NSETS,
440 - 442.000      1      VNCXFN,RXMESH,IXPTS)
441 - 443.000      DB 3506 I=1,MXPTS
442 - 444.000      VCZAFN(I)=VCZAFN(I)+VNCAFN(I)
443 - 445.000      3506 VCZXFN(I)=VCZXFN(I)+VNCXFN(I)
444 - 446.000      GO TO 37
445 - 447.000      36 IF (.NOT.LLGENA.AND..NOT.LLCKAL) GO TO 38
446 - 448.000      IF (LLA0A) IAAUNM=NEIGHA*LADINX(LAMAX1+1)+1
447 - 449.000      IF (LLA0X) IXXUNM=NEIGHA*LXDINX(LXMAX1+1)+1
448 - 450.000      37 CALL I0VCZ(LLGENA,LLA0A,LLA0X,VCZAFN,VCZXFN,IAPTS,IXPTS,
449 - 451.000      1      ALPHA,MXPTS,MXPAN,IAAFLE,IAAUNM,IXXFLE,IXXUNM)
450 - 452.000      WRITE(6,9136) (I,RAMESH(I),VNCAFN(I),VCZAFN(I),
451 - 453.000      1      RXMESH(I),VNCXFN(I),VCZXFN(I),I=1,MXPTS)
452 - 454.000      9136 FORMAT('0',2X'I',9X'RAMESH(I)',3X'I0N ENVIOR A',2X'I0T ENVIOR A',
453 - 455.000      1      9X'RXMESH(I)',3X'I0N ENVIOR X',2X'I0T ENVIOR X'/
454 - 456.000      2      (' ',I4,4X1P3E14.5,5X3E14.5)
455 - 457.000      IF (LLN0A) GO TO 3705
456 - 458.000      K=0
457 - 459.000      JMAX=LARINX(LAHI1+1)
458 - 460.000      DB 3703 J=1,JMAX
459 - 461.000      DB 3703 I=1,IAPTS
460 - 462.000      K=K+1
461 - 463.000      3703 DRANL(K)=DRANL(K)+VCZAFN(I)*RANL(K)
462 - 464.000      3705 IF (LLN0X) GO TO 3709
463 - 465.000      K=0
464 - 466.000      JMAX=LXRINX(LXHI1+1)
465 - 467.000      DB 3707 J=1,JMAX
466 - 468.000      DB 3707 I=1,IXPTS
467 - 469.000      K=K+1
468 - 470.000      3707 DRXNL(K)=DRXNL(K)+VCZXFN(I)*RXNL(K)
469 - 471.000      C3709 CALL ELAPSE(IMINU)
470 - 471.500      3709 CONTINUE
471 - 472.000      CCCCC FMINU=FL0AT(IMINU)*1.0E-3
472 - 473.000      CCCCC THINU=THINU+FMINU
473 - 474.000      CCCCC WRITE(6,9095) FMINU,THINU
474 - 475.000      CCCCC *****
475 - 476.000      C      BL0CH'S

```

```

476 - 477.000 CCCCC *****
477 - 477.005   38 IF (LLNABA) GO TO 620
478 - 477.010     DO 601 I=1,LAMAX1
479 - 477.015   601 READ(5,649) NAF(I),NAG(I)
480 - 477.016     WRITE(6,651)
481 - 477.017   651 FORMAT(/2X,'I J FA(I,J) XAF(I,J)'/)
482 - 477.020     DO 612 I=1,LAMAX1
483 - 477.025     N=NAF(I)
484 - 477.026     IF (N.EQ.0) GO TO 612
485 - 477.030     DO 602 J=1,N
486 - 477.033     READ(5,648) FA(I,J),XAF(I,J)
487 - 477.034   602 WRITE(6,650) I,J,FA(I,J),XAF(I,J)
488 - 477.035   650 FORMAT(1X,I2,1X,I2,1X,I4,F12.8)
489 - 477.036   612 CONTINUE
490 - 477.037     WRITE(6,653)
491 - 477.038   653 FORMAT(/' I J GA(I,J,1) GA(I,J,2) XAG(I,J)'/)
492 - 477.040     DO 613 I=1,LAMAX1
493 - 477.045     N=NAG(I)
494 - 477.046     IF (N.EQ.0) GO TO 613
495 - 477.050     DO 603 J=1,N
496 - 477.053     READ(5,647) GA(I,J,1),GA(I,J,2),XAG(I,J)
497 - 477.054   603 WRITE(6,652) I,J,GA(I,J,1),GA(I,J,2),XAG(I,J)
498 - 477.055   652 FORMAT(1X,I2,1X,I2,1X,I5,5X,I5,1X,F12.8)
499 - 477.056   613 CONTINUE
500 - 477.060   649 FORMAT(3I4)
501 - 477.061   648 FORMAT(I4,F12.9)
502 - 477.065   647 FORMAT(2I4,F12.9)
503 - 477.070   620 IF (LLNABX) GO TO 624
504 - 477.075     DO 621 I=1,LXMAX1
505 - 477.080   621 READ(5,649) NXF(I),NXG(I)
506 - 477.081     WRITE(6,654)
507 - 477.082   654 FORMAT(/2X,'I J FX(I,J) XXF(I,J)'/)
508 - 477.085     DO 632 I=1,LXMAX1
509 - 477.090     N=NXF(I)
510 - 477.091     IF (N.EQ.0) GO TO 632
511 - 477.095     DO 622 J=1,N
512 - 477.099     READ(5,648) FX(I,J),XXF(I,J)
513 - 477.100   622 WRITE(6,650) I,J,FX(I,J),XXF(I,J)
514 - 477.101   632 CONTINUE
515 - 477.102     WRITE(6,655)
516 - 477.103   655 FORMAT(/' I J GX(I,J,1) GX(I,J,2) XXG(I,J)'/)
517 - 477.105     DO 633 I=1,LXMAX1
518 - 477.110     N=NXG(I)
519 - 477.111     IF (N.EQ.0) GO TO 633
520 - 477.115     DO 623 J=1,N
521 - 477.119     READ(5,647) GX(I,J,1),GX(I,J,2),XXG(I,J)
522 - 477.120   623 WRITE(6,652) I,J,GX(I,J,1),GX(I,J,2),XXG(I,J)
523 - 477.121   633 CONTINUE
524 - 477.125   624 IF (LLNABA) GO TO 625
525 - 477.130     LBCFT=0
526 - 477.135     DO 607 I=1,LAMAX1
527 - 477.140   607 LBCFT=LBCFT + LBCAFN(I)
528 - 477.145     DO 608 I=1,LAMAX1
529 - 477.146     IF (NAF(I).EQ.0.AND.NAG(I).EQ.0) GO TO 608
530 - 477.150     IF (FA(I,NAF(I)).EQ.LBCFT.AND.GA(I,NAG(I),2).EQ.LBCFT)
531 - 477.155     1GO TO 608
532 - 477.160     WRITE(6,643)FA(I,NAF(I)),GA(I,NAG(I),2),LBCFT,I
533 - 477.165   643 FORMAT(1X,'FA(I,NAF(I))='I2,'GA(I,NAG(I),2)='I2,
534 - 477.170     1'LBCFT='I2,'I='I2)
535 - 477.175     CALL EXIT

```

```

536 = 477.180 608 CONTINUE
537 = 477.185 625 IF (LLNABX) GO TO 611
538 = 477.190 L0CFT=0
539 = 477.195 DO 609 I=1,LXMAX1
540 = 477.200 L0CFT=L0CFT + L0CXFN(I)
541 = 477.205 DO 610 I=1,LXMAX1
542 = 477.206 IF (NXF(I).EQ.0.AND.NXG(I).EQ.0) GO TO 610
543 = 477.210 IF (FX(I,NXF(I)).EQ.L0CFT.AND.GX(I,NXG(I),2).EQ.L0CFT)
544 = 477.215 1 GO TO 610
545 = 477.220 WRITE(6,642) FX(I,NXF(I)),GX(I,NXG(I),2),L0CFT,I
546 = 477.225 642 FORMAT(1X,'FX(I,NXF(I))=',I2,'GX(I,NXG(I),2)=',I2,
547 = 477.230 1'L0CFT=',I2,'I=',I2)
548 = 477.235 CALL EXIT
549 = 477.240 610 CONTINUE
550 = 478.000 611 READ(5,914) NBLCHA,NBLCHX
551 = 479.000 914 FORMAT(2I5)
552 = 480.000 WRITE(6,915) NBLCHA,NBLCHX
553 = 481.000 915 FORMAT('0','NBLCHA =',I3,' NBLCHX =',I3/)
554 = 482.000 IF (LLNABA) GO TO 44
555 = 483.000 IF (NBLCHA.GT.0) GO TO 40
556 = 484.000 39 WRITE(6,916) LLA0A,NBLCHA,MBLCHA
557 = 485.000 916 FORMAT(' ','LLA0A =',I2,' NBLCHA =',I3,' MBLCHA =',I3)
558 = 486.000 CALL EXIT
559 = 487.000 40 IF (NBLCHA.GT.MBLCHA) GO TO 39
560 = 488.000 DO 42 NBLCH=1,NBLCHA
561 = 489.000 READ(5,9165) NAA,LAA
562 = 490.000 9165 FORMAT(2I5)
563 = 491.000 IF (NAA-LAA.LE.LT0AFN(LAA+1)) GO TO 41
564 = 492.000 WRITE(6,917) NAA,LAA,LT0AFN(LAA+1)
565 = 493.000 917 FORMAT(' ','ORBITAL TO BE USED FOR BUILDING BLOCK '
566 = 494.000 1 'FUNCTION DOES NOT EXIST'/
567 = 495.000 2 ' ','NAA =',I2,' LLA =',I2,' LT0AFN(LAA+1) =',I2)
568 = 496.000 CALL EXIT
569 = 497.000 41 NA(NBLCH)=NAA
570 = 498.000 LA(NBLCH)=LAA
571 = 499.000 MT0T=2*LAA+1
572 = 500.000 42 READ(5,918) (CA(NBLCH,I),I=1,MT0T)
573 = 501.000 918 FORMAT(8F10.0)
574 = 502.000 DO 43 NBLCH=1,NBLCHA
575 = 503.000 NAA=NA(NBLCH)
576 = 504.000 LAA=LA(NBLCH)
577 = 505.000 WRITE(6,919) NBLCH,NAA,LAA
578 = 506.000 919 FORMAT(' ','NBLCH =',I3,' NAA =',I2,' LAA =',I2)
579 = 507.000 MT0T=2*LAA+1
580 = 508.000 L1=LAA+1
581 = 509.000 DO 43 J=1,MT0T
582 = 510.000 M=J-L1
583 = 511.000 INX=2*IABS(M)
584 = 512.000 IF (M.LE.0) INX=INX+1
585 = 513.000 43 WRITE(6,920) LAA,M,CA(NBLCH,INX)
586 = 514.000 920 FORMAT(' ','Y(',I2,',',I2,') C = (',I1PE11.4,',',E11.4,')')
587 = 515.000 GO TO 45
588 = 516.000 44 IF (NBLCHA.NE.0) GO TO 39
589 = 517.000 45 IF (LLNABX) GO TO 51
590 = 518.000 IF (NBLCHX.GT.0) GO TO 47
591 = 519.000 46 WRITE(6,921) LLA0X,NBLCHX,MBLCHX
592 = 520.000 921 FORMAT(' ','LLA0X =',I2,' NBLCHX =',I3,' MBLCHX =',I3)
593 = 521.000 CALL EXIT
594 = 522.000 47 IF (NBLCHX.GT.MBLCHX) GO TO 46
595 = 523.000 DO 49 NBLCH=1,NBLCHX

```

```

596 - 524.000      READ(5,9165) NXX,LXX
597 - 525.000      IF (NXX=LXX.LE.LT0XFN(LXX+1)) GO TO 48
598 - 526.000      WRITE(6,922) NXX,LXX,LT0XFN(LXX+1)
599 - 527.000      922 FORMAT(' ', 'ORBITAL TO BE USED FOR BUILDING BLOCK !'
600 - 528.000      1      'FUNCTIONS DOES NOT EXIST'/
601 - 529.000      2      ' ', 'NXX =', I2, ' LXX =', I2, ' LT0XFN(LXX+1) =', I2)
602 - 530.000      CALL EXIT
603 - 531.000      48 NX(NBLCH)=NXX
604 - 532.000      LX(NBLCH)=LXX
605 - 533.000      MTOT=2*LXX+1
606 - 534.000      49 READ(5,918) (CX(NBLCH,I), I=1, MTOT)
607 - 535.000      DO 50 NBLCH=1, NBLCHX
608 - 536.000      NXX=NX(NBLCH)
609 - 537.000      LXX=LX(NBLCH)
610 - 538.000      WRITE(6,923) NBLCH,NXX,LXX
611 - 539.000      923 FORMAT(' ', 'NBLCH =', I3, ' NXX =', I2, ' LXX =', I2)
612 - 540.000      MTOT=2*LXX+1
613 - 541.000      L1=LXX+1
614 - 542.000      DO 50 J=1, MTOT
615 - 543.000      M=J-L1
616 - 544.000      INX=2*IABS(M)
617 - 545.000      IF (M.LE.0) INX=INX+1
618 - 546.000      50 WRITE(6,920) LXX,M,CX(NBLCH,INX)
619 - 547.000      GO TO 52
620 - 548.000      51 IF (NBLCHX.NE.0) GO TO 46
621 - 549.000      52 READ(5,9232) LLETE,NLETE
622 - 550.000      9232 FORMAT(L5,I5)
623 - 551.000      IF (.NOT.LLETE) GO TO 5250
624 - 552.000      READ(5,9234) (KLETE(I), I=1,NLETE)
625 - 553.000      9234 FORMAT(16I5)
626 - 554.000      WRITE(6,9236) (KLETE(I), I=1,NLETE)
627 - 555.000      9236 FORMAT('0', 'BASIS FUNCTIONS TO BE DELETED'/( ' ', 16I5))
628 - 556.000      IF (NLETE.LE.1) GO TO 5250
629 - 557.000      DO 5210 ILETE=2,NLETE
630 - 558.000      5210 KLETE(ILETE)*KLETE(ILETE)=ILETE+1
631 - 559.000      WRITE(6,9238) (KLETE(I), I=1,NLETE)
632 - 560.000      9238 FORMAT(' ', 'BASIS FUNCTION POSITION'/( ' ', 16I5))
633 - 561.000      CCCCC *****
634 - 562.000      C      K VECTOR
635 - 563.000      CCCCC *****
636 - 564.000      5250 NVEC=0
637 - 565.000      5290 READ(5,924,END=54) XKX,XKY,XKZ,IWT,LLGENI,NEIGAC,NEIGXC,NVECSV
638 - 566.000      924  FORMAT(3F10.0,I5,L5,3I5)
639 - 567.000      C      VECTOR K ASSUMED TO HAVE UNITS OF 2*PI/AO
640 - 568.000      C      K VECTOR USED FOR GENERATING INTEGRALS SHOULD BE WELL WITHIN
641 - 569.000      C      BOUNDRY OF SYMMETRY ZONE
642 - 570.000      IF (NEIGAC.LE.NEIGHA.AND.NEIGXC.LE.NEIGHX) GO TO 52
643 - 571.000      WRITE(6,9242) NEIGAC,NEIGXC
644 - 572.000      9242 FORMAT(' ', 'NEIGAC =', I3, ' AND/OR NEIGXC =', I3, ' TOO LARGE'!
645 - 573.000      CALL EXIT
646 - 574.000      53 IF (NEIGAC.EQ.0) NEIGAC=NEIGHA
647 - 575.000      IF (NEIGXC.EQ.0) NEIGXC=NEIGHX
648 - 576.000      IF (.NOT.LLFRST) LLGENI=.FALSE.
649 - 577.000      LLFRST=.FALSE.
650 - 578.000      IF (NVECSV.EQ.0) NVECSV=NVEC
651 - 579.000      NVEC=NVECSV
652 - 580.000      WRITE(6,9245) XKX,XKY,XKZ,IWT,LLGENI,NEIGAC,NEIGXC,NVEC
653 - 581.000      9245 FORMAT('0', 'XKX =', E11.5, ' XKY =', E11.5, ' XKZ =', E11.5,
654 - 582.000      2      ' IWT =', I3,
655 - 583.000      3      ' LLGENI =', L2, ' NEIGAC =', I3, ' NEIGXC =', I3,

```

```

656 - 584.000      4      NVEC = 1, I3/)
657 - 585.000      CALL MTRX(NA,NX,LA,LX,FRA,FRX,LOCAFN,LOCXFN,LARINX,LXRINX,
658 - 586.000      1      LADINX,LXDINX,LAMAX1,LXMAX1,CA,CX,MBLCHA,MBLCHX,
659 - 587.000      2      NBLCHA,NBLCHX,RANL,RXNL,DRANL,DRXNL,ALPHA,XPPT,
660 - 588.000      3      MXPAN,NXPAN,RAMESH,RXMESH,IAPTS,IXPTS,
661 - 589.000      4      RU,RUB,UNIT,NAASET,NAASET,NEIGHA,NEIGHX,
662 - 590.000      5      NEIGAC,NEIGXC,KLETE,NLETE,XXY,XXY,XXZ,INT,NVEC,
663 - 590.500      6      NAG,NXG,XAF,XXF,GA,GX,XAG,XXG,FRASP,FRXBP,
664 - 591.000      7      LLGENI,IFILE,IAAFLE,IXFLE,IXFLE,IXFLE,IEVFL,
665 - 592.000      8      LLNAB,LLNABX,LLDTAL,LLBUF,NROUT,LLMTRX)
666 - 593.000      CCCCC CALL ELAPSE(IMINU)
667 - 594.000      CCCCC FMINU=FLOAT(IMINU)*1.0E-3
668 - 595.000      CCCCC TMINU=TMINU+FMINU
669 - 596.000      CCCCC WRITE(6,9095) FMINU,TMINU
670 - 597.000      GO TO 5290
671 - 598.000      54 CALL EXIT
672 - 599.000      STOP
673 - 600.000      END
674 - 601.000      FUNCTION FACT(I)
675 - 602.000      CCCCC FUNCTION FACT(I) DETERMINES FACTORIAL OF INTEGER I USING
676 - 603.000      CCCCC FLOATING POINT ARITHMETIC
677 - 604.000      CCCCC WRITTEN BY D. J. MICKISH, FEBRUARY, 1972.
678 - 605.000      CCCCC LIMERR = NUMBER OF ERRORS ALLOWED BEFORE JOB IS TERMINATED
679 - 606.000      DATA LIMERR/10/,NERR/0/
680 - 607.000      FACT=1.0
681 - 608.000      IF (I.GE.0) GO TO 3
682 - 609.000      WRITE(6,1) I
683 - 610.000      1 FORMAT(' ', 'NEGATIVE ARGUMENT IN FUNCTION FACT, RESULT = 1.0, ')
684 - 611.000      1      ' ARGUMENT = ',I12)
685 - 612.000      2 NERR=NERR+1
686 - 613.000      IF (NERR.GE.LIMERR) CALL EXIT
687 - 614.000      RETURN
688 - 615.000      3 IF (I.LE.1) RETURN
689 - 616.000      IF (I.LE.56) GO TO 5
690 - 617.000      WRITE(6,4) I
691 - 618.000      4 FORMAT(' ', 'ARGUMENT = ',I4, ' IN FUNCTION FACT .GT. 56, ')
692 - 619.000      1      ' RESULT = 7.237E+75')
693 - 620.000      GO TO 2
694 - 621.000      5 FACT=FLOAT(I)
695 - 622.000      XI=FACT-1.0
696 - 623.000      CCCCC I GREATER THAN 2, COMPUTE FACTORIAL
697 - 624.000      6 IF (XI.LE.1.0) RETURN
698 - 625.000      FACT=XI*FACT
699 - 626.000      XI=XI-1.0
700 - 627.000      GO TO 6
701 - 628.000      END
702 - 629.000      SUBROUTINE IOUT1(I11111,N,ZZZZ)
703 - 630.000      DIMENSION I11111(N)
704 - 631.000      REAL*8 ZZZZ
705 - 632.000      WRITE(6,1)
706 - 633.000      1 FORMAT(1H )
707 - 634.000      WRITE(6,2) (ZZZZ,J,I11111(J),J=1,N)
708 - 635.000      2 FORMAT(1P4(1X46,1H(,I3,3H) =,I12))
709 - 636.000      RETURN
710 - 637.000      END
711 - 638.000      SUBROUTINE OUT1(XXXXXX,N,ZZZZ)
712 - 639.000      DIMENSION XXXXXX(N)
713 - 640.000      REAL*8 ZZZZ
714 - 641.000      WRITE(6,1)
715 - 642.000      1 FORMAT(1H )

```

```

716 = 643.000      WRITE(6,2) (ZZZ,J,XXXXXX(J),J=1,N)
717 = 644.000      2 FORMAT(1P4(1XA6,1H(,I3,3H) =,E13.6)
718 = 645.000      RETURN
719 = 646.000      END
720 = 647.000      SUBROUTINE IR12K(P3P4,K,SS,RMESH,IPTS,XN)
721 = 648.000      DIMENSION P3P4(1),SS(1),RMESH(1)
722 = 649.000      CCCCC      COMPUTE AND STORE 'R**K' INTEGRAND
723 = 650.000      SAVES=0.0
724 = 651.000      SAVESN=0.0
725 = 652.000      RINT=RMESH(IPTS/2)
726 = 653.000      DO 13 J=3,IPTS,2
727 = 654.000      IF (J.NE.3) GO TO 11
728 = 655.000      R1=RMESH(J=2)/RINT
729 = 656.000      F1=P3P4(J=2)
730 = 657.000      R1K=R1**K
731 = 658.000      R1K1I=1.0/(R1+R1K)
732 = 659.000      GO TO 12
733 = 660.000      11 R1=R3
734 = 661.000      F1=F3
735 = 662.000      R1K=R3K
736 = 663.000      R1K1I=R3K1I
737 = 664.000      12 R2=RMESH(J=1)/RINT
738 = 665.000      F2=P3P4(J=1)
739 = 666.000      R2K=R2**K
740 = 667.000      R2K1I=1.0/(R2+R2K)
741 = 668.000      R3=RMESH(J)/RINT
742 = 669.000      F3=P3P4(J)
743 = 670.000      R3K=R3**K
744 = 671.000      R3K1I=1.0/(R3+R3K)
745 = 672.000      H3=(R3=R2)/3.0*RINT
746 = 673.000      SAVESN=SAVESN+H3*(F1*R1K1I+4.0*F2*R2K1I+F3*R3K1I)
747 = 674.000      SAVES=SAVES+H3*(F1*R1K+4.0*F2*R2K+F3*R3K)
748 = 675.000      13 SS(J)=(R3K1I*SAVES=R3K+SAVESN)/RINT
749 = 676.000      DO 14 J=3,IPTS,2
750 = 677.000      R=RMESH(J)
751 = 678.000      14 SS(J)=SS(J)+SAVESN*(R/RINT)**K/RINT+XN/R
752 = 679.000      SS(1)=XN/RMESH(1)
753 = 680.000      SS(2)=0.5*(SS(1)+SS(3))
754 = 681.000      SS(IPTS-1)=0.5*(SS(IPTS-2)+SS(IPTS))
755 = 682.000      IPTS3=IPTS-3
756 = 683.000      DO 3 I=4,IPTS3,2
757 = 684.000      RI=RMESH(I)
758 = 685.000      RIM3=RI-RMESH(I=3)
759 = 686.000      RIM1=RI-RMESH(I=1)
760 = 687.000      RI1=RI-RMESH(I+1)
761 = 688.000      RI3=RI-RMESH(I+3)
762 = 689.000      IF (I.NE.4) GO TO 1
763 = 690.000      RIM3M1=RIM1-RIM3
764 = 691.000      RIM31=RI1-RIM3
765 = 692.000      RIM11=RI1-RIM1
766 = 693.000      SM3=SS(1)
767 = 694.000      SM1=SS(3)
768 = 695.000      S1=SS(5)
769 = 696.000      GO TO 2
770 = 697.000      1 RIM3M1=RIM11
771 = 698.000      RIM31=RIM13
772 = 699.000      RIM11=RI13
773 = 700.000      SM3=SM1
774 = 701.000      SM1=S1
775 = 702.000      S1=S3

```

```

776 = 703.000      2 RIM33=RI3=RIM3
777 = 704.000      RIM13=RI3=RIM1
778 = 705.000      RI13=RI3=RI1
779 = 706.000      S3=SS(I+3)
780 = 707.000      3 SS(I)=SM3*RIM1*RI1*RI3/(RIM3M1*RIM31+RIM33)
781 = 708.000      1   =SM1*RIM3*RI1*RI3/(RIM3M1*RIM11+RIM13)
782 = 709.000      2   +S1*RIM3*RIM1*RI3/(RIM31+RIM11*RI13)
783 = 710.000      3   =S3*RIM3*RIM1*RI1/(RIM33+RIM13*RI13)
784 = 711.000      RETURN
785 = 712.000      END
786 = 713.000      SUBROUTINE RW(LOC,IFILE,BUFFER,IWORDS,LLWRT,LLBUF)
787 = 714.000      DIMENSION BUFFER(IWORDS)
788 = 715.000      LOGICAL LLWRT,LLBUF
789 = 716.000      IF (LLBUF) GO TO 2
790 = 717.000      IF (LLWRT) GO TO 1
791 = 718.000      READ(IFILE) BUFFER
792 = 719.000      RETURN
793 = 720.000      1 WRITE(IFILE) BUFFER
794 = 721.000      RETURN
795 = 722.000      C 2 IWORDS=IWORDS*2
796 = 722.500      2 CONTINUE
797 = 723.000      IF (LLWRT) GO TO 3
798 = 724.000      CALL BUFFER IN(IFILE,1,BUFFER,IWORDS,IND,N)
799 = 725.000      GO TO 4
800 = 726.000      3 CALL BUFFER OUT(IFILE,1,BUFFER,IWORDS,IND,N)
801 = 727.000      4 IF (IND.NE.2) GO TO 5
802 = 729.000      C 43 IWORDS=IWORDS/2
803 = 729.500      43 CONTINUE
804 = 730.000      RETURN
805 = 731.000      45 IF (LLWRT) GO TO 5
806 = 732.000      IF (N.EQ.IWORDS/(IWORDS/8192)*8192) GO TO 43
807 = 733.000      5 WRITE(6,900) LOC,IFILE,IWORDS,IND,N,LLWRT
808 = 734.000      900 FORMAT(' I, I0 ERROR USING BUFFER IN/OUT SUBROUTINE01/
809 = 735.000      1   ' I, 'LOCATION I, I3, ' FILE I, I3, ' NO. WORDS ATTEMPTED 01
810 = 736.000      2   I5, ' IND = I, I2, ' NO. WORDS TRANSFERRED 01, I5,
811 = 737.000      3   ' LLWRT = I, L2)
812 = 738.000      CALL EXIT
813 = 739.000      END
814 = 740.000      C I ASSIGN M:BS, (IFILE, MEXPAND), (SAVE)
815 = 741.000      C I FORTRAN LS, BS, ADP
816 = 742.000      SUBROUTINE EXPAND(L,LMAX1,RNL,INXFN,RONE,MXPTS1,IPT01,A,
817 = 743.000      1   ALPHA,MXPAN,NXPAN,RTWO,ZZR,RH02,N0,IRIS2,
818 = 744.000      2   JTHPTS,LLOUT,IFILE6,IUNLM,LLDTAL,LLBUF)
819 = 745.000      DIMENSION RNL(1),RONE(1),RTWO(1),ALPHA(MXPTS1,MXPAN)
820 = 746.000      DIMENSION CTH(101),STH(101),CCTH( 3030),RRSTR( 3030),PLM(301),
821 = 747.000      1   PKMTHS(3030),RCOEF( 3030),RNLRTH( 3030),RLMCTH( 3030),
822 = 748.000      2   BUFF1( 3030),BUFF2( 3030)
823 = 749.000      COMMON WORK(161,31)
824 = 750.000      EQUIVALENCE (PKMTHS(3030),WORK(161,31)),
825 = 751.000      1   (BUFF1(1),CCTH(1),RNLRTH(1),RCOEF(1)),
826 = 752.000      2   (BUFF2(1),RRSTR(1),PLMCTH(1))
827 = 753.000      LOGICAL LLTH,LLR1A,LLOUT,LLDTAL,LLBUF
828 = 754.000      DATA PI/3.1415926535897932//,ATEMP/0.0//,
829 = 755.000      1   IPTS1T/0//,R1LBW/0.0//,R1HI/0.0//,
830 = 756.000      2   IPTS2T/0//,R2LBW/0.0//,R2HI/0.0//,
831 = 757.000      3   JTHPTT/0//,LMAX1T/0//,LT/-1//,MT/1//
832 = 758.000      DATA IFILE0/20//,IFILE1/21//,IFILE2/22//,IFILE3/23//,IFILE4/24//,
833 = 759.000      1   IFILE5/25//
834 = 760.000      DATA MXTHT/101//,MXPANT/30//,MXPTST/161//,NRWT/ 30//
835 = 761.000      IPTS11=IPTS1+1

```

```

836 = 761.500 C CHANGE HERE FOR REAL*8 ALPHAS
837 = 762.000 IALPTS=MXPTS1*MXPAN
838 = 763.000 JTHPAN=JTHPTS*MXPAN
839 = 764.000 JTHPT3=JTHPTS*3
840 = 765.000 J5THPT=5*JTHPTS
841 = 766.000 NRW5T=NRWT/5
842 = 767.000 C DIMENSION WORK(MXPTST,MPANT+1)
843 = 768.000 C CTH,STH(MXTHPT)
844 = 769.000 C CCTH,RRSTR,RCOEF,BUFF1,RNLRTH,PLMCTH,BUFF2(NRWT*MXTHPT)
845 = 770.000 C PLM(MXPANT)
846 = 771.000 C PKMTHS(MXPANT*MXTHPT)
847 = 772.000 C PROGRAM REQUIRES MXPTST.GE.MXTHPT AND NRWT BE A MULTIPLE OF 5
848 = 773.000 L1=L+1
849 = 774.000 LMAX=LMAX1=1
850 = 775.000 IF (MXPTS1.LE.MXPTST) GO TO 1000
851 = 776.000 WRITE(6,899) MXPTS1,MXPTST
852 = 777.000 899 FORMAT(' ', 'MXPTS1 =', I4, ' .GT. MXPTST =', I4,
853 = 778.000 1 ' ' IN SUBROUTINE EXPAND!')
854 = 779.000 CALL EXIT
855 = 780.000 1000 IF (MXPAN.GT.LMAX.AND.MXPAN.LE.MXPANT) GO TO 1
856 = 781.000 WRITE(6,900) LMAX,MXPAN,MXPANT
857 = 782.000 900 FORMAT(' ', 'LMAX =', I2, ' MXPAN =', I3, ' MXPANT =', I2)
858 = 783.000 CALL EXIT
859 = 784.000 1 IF (L.LE.LMAX) GO TO 2
860 = 785.000 WRITE(6,901) L,LMAX
861 = 786.000 901 FORMAT(' ', 'L =', I2, ' SENT TO SUBROUTINE EXPAND TOO LARGE ',
862 = 787.000 1 '(LMAX =', I2, ')')
863 = 788.000 CALL EXIT
864 = 789.000 2 IF (JTHPTS.LE.MXTHPT) GO TO 3
865 = 790.000 WRITE(6,902) JTHPTS,MXTHPT
866 = 791.000 902 FORMAT(' ', 'JTHPTS =', I4, ' SENT TO SUBROUTINE EXPAND TOO LARGE ',
867 = 792.000 1 '(MXTHPT =', I4, ')')
868 = 793.000 CALL EXIT
869 = 794.000 3 IF (NXPAN.LE.MXPAN) GO TO 4
870 = 795.000 WRITE(6,903) NXPAN,MXPAN
871 = 796.000 903 FORMAT(' ', 'NXPAN =', I3, ' SENT TO SUBROUTINE EXPAND TOO LARGE ',
872 = 797.000 1 '(MXPAN =', I3, ')')
873 = 798.000 CALL EXIT
874 = 799.000 4 IF (IPTSI.LE.MXPTS1) GO TO 5
875 = 800.000 WRITE(6,904) IPTSI,MXPTS1
876 = 801.000 904 FORMAT(' ', 'IPTSI =', I4, ' SENT TO SUBROUTINE EXPAND TOO LARGE ',
877 = 802.000 1 '(MXPTS1 =', I4, ')')
878 = 803.000 CALL EXIT
879 = 804.000 5 LLTH=.FALSE.
880 = 805.000 C COMPUTE CONSTANTS IF JTHPTS HAS BEEN CHANGED OR LMAX1 HAS BEEN
881 = 806.000 C INCREASED
882 = 807.000 C SIN(THETA)
883 = 808.000 C COS(THETA)
884 = 809.000 C P(K,M)(COS(THETA)) * SIN(THETA) (FILEO)
885 = 810.000 IF (JTHPTS.EQ.JTHPT1.AND.LMAX1.LE.LMAX1T) GO TO 9
886 = 811.000 LLTH=.TRUE.
887 = 812.000 JTHPT1=JTHPTS
888 = 813.000 LMAX1T=LMAX1
889 = 814.000 DTH=PI/FL0AT(JTHPTS=1)
890 = 815.000 DTH3=DTH/3.0
891 = 816.000 XJ=0.0
892 = 817.000 DO 6 J=1,JTHPTS
893 = 818.000 TH=XJ*DTH
894 = 819.000 CTH(J)=COS(TH)
895 = 820.000 STH(J)=SIN(TH)

```

```

896 = 821.000      6 XJ=XJ+1.0
897 = 822.000      DO 8 M1=1, LMAX1
898 = 823.000      M=M1-1
899 = 824.000      LIM=MXPAN=M
900 = 825.000      DO 7 J=1, JTHPTS
901 = 826.000      CALL LEGEN(MXPAN=1, M, CTH(J), PLM)
902 = 827.000      SINTH=STH(J)
903 = 828.000      INDEX=J+M*JTHPTS
904 = 829.000      DO 7 LL=1, LIM
905 = 830.000      PKMTHS(INDEX)=PLM(LL)*SINTH
906 = 831.000      7 INDEX=INDEX+JTHPTS
907 = 832.000      IF (M.EQ.0) GO TO 702
908 = 833.000      ILIM=M*JTHPTS
909 = 834.000      DO 701 I=1, ILIM
910 = 835.000      701 PKMTHS(I)=0.0
911 = 836.000      702 IF (LLDTAL) WRITE(6,905) M
912 = 837.000      905 FORMAT(' ', 'PKMTHS BLOCK CALCULATED FOR M =', I2, ' FILE0')
913 = 838.000      CALL RW(1, IFILE0, PKMTHS, JTHPAN, .TRUE., LLBUF)
914 = 839.000      8 CONTINUE
915 = 840.000      REWIND IFILE0
916 = 841.000      C CALCULATE FOLLOWING CONSTANTS IF A OR RONE MESH HAS CHANGED
917 = 842.000      C RR(R, THETA) (FILE1)
918 = 843.000      C COS(CAP*THETA(R, THETA)) (FILE2)
919 = 844.000      9 LLR1A=.FALSE.
920 = 845.000      IF (LLTH.OR. A.NE.ATEMP.OR. IPTS1.NE.IPTS1T.OR.
921 = 846.000      1 RONE(1).NE.R1LOW.OR.RONE(IPTS1).NE.R1H) GO TO 10
922 = 847.000      GO TO 13
923 = 848.000      10 LLR1A=.TRUE.
924 = 849.000      ATEMP=A
925 = 850.000      IPTS1T=IPTS1
926 = 851.000      R1LOW=RONE(1)
927 = 852.000      R1H=RONE(IPTS1)
928 = 853.000      ASQ=A*A
929 = 854.000      NRW=0
930 = 855.000      INXRW=0
931 = 856.000      DO 12 I=1, IPTS1
932 = 857.000      R=RONE(I)
933 = 858.000      RSQ=R*R
934 = 859.000      AR=A*R
935 = 860.000      DO 11 J=1, JTHPTS
936 = 861.000      COSTH=CTH(J)
937 = 862.000      RR=SQRT(RSQ+ASQ-2.0*AR*COSTH)
938 = 863.000      RRSTR(INXRW+J)=RR
939 = 864.000      IF (ABS(RR).GT.1.0E-8) GO TO 1005
940 = 865.000      COSCTH=(-1.0)
941 = 866.000      GO TO 11
942 = 867.000      1005 COSCTH=(R*COSTH-A)/RR
943 = 868.000      IF (COSCTH.GT.1.0) COSCTH=1.0
944 = 869.000      IF (COSCTH.LT.(-1.0)) COSCTH=(-1.0)
945 = 870.000      11 CCTH(INXRW+J)=COSCTH
946 = 871.000      NRW=NRW+1
947 = 872.000      INXRW=INXRW+JTHPTS
948 = 873.000      IF (NRW.LT.NRWT.AND.I.LT.IPTS1) GO TO 12
949 = 874.000      CALL RW(2, IFILE1, RRSTR, INXRW, .TRUE., LLBUF)
950 = 875.000      CALL RW(3, IFILE2, CCTH, INXRW, .TRUE., LLBUF)
951 = 876.000      NRW=0
952 = 877.000      INXRW=0
953 = 878.000      12 CONTINUE
954 = 879.000      IF (LLDTAL) WRITE(6,906)
955 = 880.000      906 FORMAT(' ', 'RRSTR BLOCK CALCULATED FILE1')

```

```

956 = 881.000      1      ' ' 'CCTH BLOCK CALCULATED FILE2')
957 = 882.000      REWIND IFILE1
958 = 883.000      REWIND IFILE2
959 = 884.000 C     CALCULATE LARGEST INDEX II SUCH THAT RTW0(II),LT,RR IF RTW0 MESH
960 = 885.000 C     HAS CHANGED (FILE3)
961 = 886.000      13 IF (LLR1A.GR,IPTS2.NE,IPTS2T.GR.
962 = 887.000      1      RTW0(1).NE.R2LOW.GR.RTW0(IPTS2).NE.R2N) GO TO 14
963 = 888.000      GO TO 23
964 = 889.000      14 IPTS2T=IPTS2
965 = 890.000      R2LOW=RTW0(1)
966 = 891.000      R2HI=RTW0(IPTS2)
967 = 892.000      INXLIM=5*JTHPTS
968 = 893.000      NRW=NRWT
969 = 894.000      INXRW=NRW*JTHPTS
970 = 895.000      NRW5=0
971 = 896.000      INXRW5=0
972 = 897.000      DO 22 I=1,IPTS1
973 = 898.000      IF (NRW.LT.NRWT) GO TO 1406
974 = 899.000      IF (IPTS1=I+1,GE.NRWT) GO TO 1402
975 = 900.000      INXRW=(IPTS1=I+1)*JTHPTS
976 = 901.000      1402 CALL RW(4,IFILE1,RRSTR,INXRW,.FALSE.,LLBUF)
977 = 902.000      NRW=0
978 = 903.000      INXRW=0
979 = 904.000      1406 INDEX=INXRW5
980 = 905.000      DO 21 J=1,JTHPTS
981 = 906.000      RR=RRSTR(INXRW+J)
982 = 907.000      IF (RR.GT.RTW0(1)) GO TO 15
983 = 908.000      II=2
984 = 909.000      GO TO 20
985 = 910.000      15 II=1+IFIX((ALOG(Z2R*RR)-RH02)/H2)
986 = 911.000      II=4*((II=1)/4)+5
987 = 912.000      IF (II.LE.IPTS2) GO TO 1505
988 = 913.000      RC0EF(INDEX+1)=FLOAT(IPTS2)
989 = 914.000      RC0EF(INDEX+2)=0.0
990 = 915.000      RC0EF(INDEX+3)=0.0
991 = 916.000      RC0EF(INDEX+4)=0.0
992 = 917.000      RC0EF(INDEX+5)=0.0
993 = 918.000      GO TO 21
994 = 919.000      1505 DO 16 K=1,4
995 = 920.000      KK=K
996 = 921.000      IF (K.EQ.4) GO TO 17
997 = 922.000      16 IF (RR.GT.RTW0(II=K)) GO TO 17
998 = 923.000      17 II=II-KK
999 = 924.000      IF (II.GE.2) GO TO 19
1000 = 925.000      II=2
1001 = 926.000      GO TO 20
1002 = 927.000      19 IF (II.GT.IPTS2-2) II=IPTS2-2
1003 = 928.000 C     CALCULATE LAGRANGIAN INTERPOLATION COEFFICIENTS (FILE3)
1004 = 929.000      20 RRIM1=RR-RTW0(II=1)
1005 = 930.000      RRI0=RR-RTW0(II)
1006 = 931.000      RRI1=RR-RTW0(II+1)
1007 = 932.000      RRI2=RR-RTW0(II+2)
1008 = 933.000      RRIM10=RRIM1-RRI0
1009 = 934.000      RRIM11=RRI1-RRI0
1010 = 935.000      RRIM12=RRI2-RRI0
1011 = 936.000      RRI01=RRI1-RRI0
1012 = 937.000      RRI02=RRI2-RRI0
1013 = 938.000      RRI12=RRI2-RRI1
1014 = 939.000      RC0EF(INDEX+1)=FLOAT(II)
1015 = 940.000      RC0EF(INDEX+2)=RRI0*RRI1+RRI2/(RRI10+RRI11+RRI12)

```

```

1016 = 941.000      RCDEF(INDEX+3)=0.0-RRIM1*RR11*RR12/(RRIM10*RR101*RR102)
1017 = 942.000      RCDEF(INDEX+4)=RRIM1*RR10*RR12/(RRIM11*RR101*RR112)
1018 = 943.000      RCDEF(INDEX+5)=0.0-RRIM1*RR10*RR11/(RRIM12*RR102*RR112)
1019 = 944.000      21 INDEX=INDEX+5
1020 = 945.000      NRW=NRW+1
1021 = 946.000      INXRW=INXRW+JTHPTS
1022 = 947.000      NRW5=NRW5+1
1023 = 948.000      INXRW5=INXRW5+J5THPT
1024 = 949.000      IF (NRW5.LT.NRW5T.AND.I.LT.IPTS1) GO TO 22
1025 = 950.000      CALL RW(5,IFILE3,RCDEF,INXRW5,.TRUE.,LLBUF)
1026 = 951.000      NRW5=0
1027 = 952.000      INXRW5=0
1028 = 953.000      22 CONTINUE
1029 = 954.000      IF (LLDTAL) WRITE(6,907)
1030 = 955.000      907 FORMAT(' ', 'RCDEF BLOCK CALCULATED FILE3')
1031 = 956.000      REWIND IFILE1
1032 = 957.000      REWIND IFILE3
1033 = 958.000      C   CALCULATE RNL(RR(R,THETA)) (FILE4)
1034 = 959.000      C   NOTE BUFF2=RCDEF INSTEAD OF USUAL BUFF1 SINCE RNLRTH=BUFF1
1035 = 960.000      23 NRW5=NRW5T
1036 = 961.000      INXRW5=NRW5*J5THPT
1037 = 962.000      NRW=0
1038 = 963.000      INXRW=0
1039 = 964.000      DO 26 I=1,IPTS1
1040 = 965.000      IF (NRW5.LT.NRW5T) GO TO 2304
1041 = 966.000      IF (IPTS1=I+1.GE.NRW5T) GO TO 2302
1042 = 967.000      INXRW5=(IPTS1=I+1)*J5THPT
1043 = 968.000      2302 CALL RW(6,IFILE3,BUFF2,INXRW5,.FALSE.,LLBUF)
1044 = 969.000      NRW5=0
1045 = 970.000      INXRW5=0
1046 = 971.000      2304 INDEX=INXRW5
1047 = 972.000      DO 25 J=1,JTHPTS
1048 = 973.000      II=IFIX(BUFF2(INDEX+1))
1049 = 974.000      IF (II.NE.IPTS2) GO TO 24
1050 = 975.000      RNLRTH(INXRW+J)=0.0
1051 = 976.000      GO TO 25
1052 = 977.000      C   INTERPOLATE USING CALCULATED COEFFICIENTS
1053 = 978.000      24 II=II+INXFN
1054 = 979.000      RNLRTH(INXRW+J)=BUFF2(INDEX+2)*RNL(II-1)
1055 = 980.000      1      +BUFF2(INDEX+3)*RNL(II)
1056 = 981.000      2      +BUFF2(INDEX+4)*RNL(II+1)
1057 = 982.000      3      +BUFF2(INDEX+5)*RNL(II+2)
1058 = 983.000      25 INDEX=INDEX+5
1059 = 984.000      NRW5=NRW5+1
1060 = 985.000      INXRW5=INXRW5+J5THPT
1061 = 986.000      NRW=NRW+1
1062 = 987.000      INXRW=INXRW+JTHPTS
1063 = 988.000      IF (NRW.LT.NRW.T.AND.I.LT.IPTS1) GO TO 26
1064 = 989.000      CALL RW(7,IFILE4,RNLRTH,INXRW,.TRUE.,LLBUF)
1065 = 990.000      NRW=0
1066 = 991.000      INXRW=0
1067 = 992.000      26 CONTINUE
1068 = 993.000      IF (LLDTAL) WRITE(6,908)
1069 = 994.000      908 FORMAT(' ', 'RNLRTH BLOCK CALCULATED FILE4')
1070 = 995.000      REWIND IFILE3
1071 = 996.000      REWIND IFILE4
1072 = 997.000      DO 35 M1=1,L1
1073 = 998.000      M=M1-1
1074 = 999.000      C   COMPUTE P(L,M)(COS(CAP=THETA(R,THETA))) (FILE5)
1075 = 1000.000     IF (LLR1A.OR.L.NE.LT.OR.M.NE.MT) GO TO 29

```

```

1076 - 1001.000      GO TO 3201
1077 - 1002.000      29 LT=L
1078 - 1003.000      MT=M
1079 - 1004.000      NRW=NRWT
1080 - 1005.000      INXRW=NRW*JTHPTS
1081 - 1006.000      DO 32 I=1,IPTS1
1082 - 1007.000      IF (NRW*LT*NRWT) GO TO 3006
1083 - 1008.000      IF (IPTS1=I+1,GE*NRWT) GO TO 3002
1084 - 1009.000      INXRW=(IPTS1=I+1)*JTHPTS
1085 - 1010.000      3002 CALL RW(8,IFILE2,CCTH,INXRW,.FALSE,,LLBUF)
1086 - 1011.000      NRW=0
1087 - 1012.000      INXRW=0
1088 - 1013.000      3006 DO 31 J=1,JTHPTS
1089 - 1014.000      CALL LEGEN(L,M,CCTH(INXRW+J),PLM)
1090 - 1015.000      31 PLMCTH(INXRW+J)=PLM(L=M+1)
1091 - 1016.000      NRW=NRW+1
1092 - 1017.000      INXRW=INXRW+JTHPTS
1093 - 1018.000      IF (NRW*LT*NRWT*AND*I,LT,IPTS1) GO TO 32
1094 - 1019.000      CALL RW(9,IFILE5,PLMCTH,INXRW,.TRUE,,LLBUF)
1095 - 1020.000      32 CONTINUE
1096 - 1021.000      IF (LLDTAL) WRITE(6,909) L,M
1097 - 1022.000      909 FORMAT(' ',PLMCTH BLOCK CALCULATED WITH L = ',I2,' AND M = ',I2,
1098 - 1023.000      1      ' FILE5')
1099 - 1024.000      REWIND IFILE2
1100 - 1025.000      REWIND IFILE5
1101 - 1026.000      3201 CALL RW(10,IFILE0,PKMTHS,JTHPAN,.FALSE,,LLBUF)
1102 - 1027.000      INDEX=M*JTHPTS
1103 - 1028.000      DO 3405 K1=M1,NXPAN
1104 - 1029.000      K=K1-1
1105 - 1030.000      XNORM=FLOAT(2*K+1)*FACT(K=M)/(2.0*FACT(K+M))
1106 - 1031.000      NRW=NRWT
1107 - 1032.000      INXRW=NRW*JTHPTS
1108 - 1033.000      DO 34 I=1,IPTS1
1109 - 1034.000      IF (NRW*LT*NRWT) GO TO 3208
1110 - 1035.000      IF (IPTS1=I+1,GE*NRWT) GO TO 3202
1111 - 1036.000      INXRW=(IPTS1=I+1)*JTHPTS
1112 - 1037.000      3202 CALL RW(11,IFILE4,RNLRTH,INXRW,.FALSE,,LLBUF)
1113 - 1038.000      CALL RW(12,IFILE5,PLMCTH,INXRW,.FALSE,,LLBUF)
1114 - 1039.000      NRW=0
1115 - 1040.000      INXRW=0
1116 - 1041.000      3208 SUM=RNLRTH(INXRW+1)*PKMTHS(INDEX+1)+PLMCTH(INXRW+1)
1117 - 1042.000      1      +4.0*RNLRTH(INXRW+JTHPTS-1)*PKMTHS(INDEX+JTHPTS-1)
1118 - 1043.000      2      *PLMCTH(INXRW+JTHPTS-1)
1119 - 1044.000      3      +RNLRTH(INXRW+JTHPTS)*PKMTHS(INDEX+JTHPTS)+PLMCTH(INXRW+JTHPTS)
1120 - 1045.000      DO 33 J=2,JTHPTS,2
1121 - 1046.000      33 SUM=SUM+4.0*RNLRTH(INXRW+J)*PKMTHS(INDEX+J)+PLMCTH(INXRW+J)
1122 - 1047.000      1      +2.0*RNLRTH(INXRW+J+1)*PKMTHS(INDEX+J+1)+PLMCTH(INXRW+J+1)
1123 - 1048.000      NRW=NRW+1
1124 - 1049.000      INXRW=INXRW+JTHPTS
1125 - 1050.000      34 ALPHA(I,K1)=XNORM*DTH3*SUM
1126 - 1051.000      INDEX=INDEX+JTHPTS
1127 - 1052.000      REWIND IFILE4
1128 - 1053.000      REWIND IFILE5
1129 - 1054.000      3405 CONTINUE
1130 - 1055.000      IF (IPTS1.EQ.MXPPTS1) GO TO 3407
1131 - 1056.000      DO 3406 K1=1,MXPAN
1132 - 1057.000      DO 3406 I=1,IPTS1,MXPPTS1
1133 - 1058.000      3406 ALPHA(I,K1)=0.0
1134 - 1059.000      3407 IF (M.EQ.0) GO TO 3409
1135 - 1060.000      DO 3408 K1=1,M

```

```

1136 - 1061.000      DO 3408 I=1,IPTS1
1137 - 1062.000      3408 ALPHA(I,K1)=0.0
1138 - 1063.000      3409 WRITE(6,910) A,INXFN,L,M,IUNLM,LLBUT
1139 - 1064.000      910 FORMAT(' ',ALPHA BLOCK CALCULATED FOR A =',F9.5,' INXFN =',I4,
1140 - 1065.000      1      ' L =',I2,' M =',I2,' IUNLM =',I3,' LLBUT =',L2/)
1141 - 1066.000      C      IF LLBUT IS TRUE, ALPHA FUNCTIONS ARE WRITTEN ON DISK. UPON
1142 - 1067.000      C      RETURN TO CALLING PROGRAM THE ARRAY ALPHA CONTAINS LAST SET
1143 - 1068.000      C      OF ALPHA FUNCTIONS GENERATED (M=L).
1144 - 1069.000      IF (.NOT.LLBUT) GO TO 35
1145 - 1070.000      CCCCC *****
1146 - 1071.000      IUNLM=IALPTS*(IUNLM+1)
1147 - 1072.000      WRITE DISK IFILE6,IUNLM,ALPHA
1148 - 1073.000      IUNLM=(IUNLM+1)/IALPTS+1
1149 - 1074.000      CCCCC *****
1150 - 1075.000      CCCCC *****
1151 - 1076.000      C      WRITE(IFILE6,IUNLM) ALPHA
1152 - 1077.000      CCCCC *****
1153 - 1078.000      IUNLM=IUNLM+1
1154 - 1079.000      35 CONTINUE
1155 - 1080.000      REWIND IFILE0
1156 - 1081.000      RETURN
1157 - 1082.000      END
1158 - 1083.000      SUBROUTINE LEGEN(LL,M,X,PLM)
1159 - 1084.000      CCCCC      SUBROUTINE LEGEN(LL,M,X,PLM) COMPUTES THE ASSOCIATED
1160 - 1085.000      CCCCC      LEGENDRE POLYNOMIALS P(M,M) TO P(LL,M) OF ARGUMENT X AND
1161 - 1086.000      CCCCC      STORES THEM IN VECTOR PLM.
1162 - 1087.000      CCCCC      WRITTEN BY D. J. MICKISH, FEBRUARY 1972
1163 - 1088.000      CCCCC      EVALUATION IS BASED ON THE RECURSION RELATION FOR ASSOCIATED
1164 - 1089.000      CCCCC      LEGENDRE POLYNOMIALS P(L,M,X)
1165 - 1090.000      CCCCC      P(L,M,X)=2*X*P(L-1,M,X)-P(L-2,M,X)
1166 - 1091.000      CCCCC      + (2*M-1)*(X*P(L-1,M,X)-P(L-2,M,X))/L(M)
1167 - 1092.000      CCCCC      THIS IS THE FORM ANALOGOUS TO THAT USED IN IBM'S SGP SUB=
1168 - 1093.000      CCCCC      ROUTINES FOR LEGENDRE POLYNOMIALS. THE ADVANTAGES ARE
1169 - 1094.000      CCCCC      'ECONOMY AND NUMERICAL STABILITY'
1170 - 1095.000      CCCCC      STARTING VALUES ARE P(M-1,M,X)=0.0
1171 - 1096.000      CCCCC      P(M,M,X)=(2*M-1)!!*(1.0-X**2)**(M/2)
1172 - 1097.000      DIMENSION PLM(1)
1173 - 1098.000      CCCCC      COMPUTE P(M,M)
1174 - 1099.000      L=M
1175 - 1100.000      IF (L.NE.0) GO TO 1
1176 - 1101.000      PL=1.0
1177 - 1102.000      GO TO 5
1178 - 1103.000      1 XINT=FLOAT(2*L-1)
1179 - 1104.000      CCCCC      DETERMINE IF SQUARE ROOT NECESSARY
1180 - 1105.000      IF (L-2*(L/2).EQ.0) GO TO 2
1181 - 1106.000      PL=XINT*SQRT(1.0-X*X)**L
1182 - 1107.000      GO TO 3
1183 - 1108.000      2 PL=XINT*(1.0-X*X)**(L/2)
1184 - 1109.000      3 IMAX=L-2
1185 - 1110.000      IF (IMAX.LE.0) GO TO 5
1186 - 1111.000      DO 4 I=1,IMAX
1187 - 1112.000      XINT=XINT-2.0
1188 - 1113.000      4 PL=XINT*PL
1189 - 1114.000      5 PLM(1)=PL
1190 - 1115.000      CCCCC      RETURN IF ONLY ONE VALUE REQUIRED (LL=M)
1191 - 1116.000      IF (LL.EQ.L) RETURN
1192 - 1117.000      CCCCC      USE RECURSION RELATIONS TO COMPUTE REMAINING POLYNOMIALS
1193 - 1118.000      PL2=0.0
1194 - 1119.000      PL1=PL
1195 - 1120.000      LM=LM+1

```

```

1196 - 1121.000      X2M1=FLB*AT(M+M-1)
1197 - 1122.000      XLM=FLB*AT(LM*IN=M)
1198 - 1123.000      INDEX=2
1199 - 1124.000      DO 6 L=LMIN,LL
1200 - 1125.000      XPL1=X*PL1
1201 - 1126.000      PL=XPL1-PL2+XPL1+X2M1*(XPL1-PL2)/XLM
1202 - 1127.000      PLM(INDEX)=PL
1203 - 1128.000      PL2=PL1
1204 - 1129.000      PL1=PL
1205 - 1130.000      XLM=XLM+1.0
1206 - 1131.000      6 INDEX=INDEX+1
1207 - 1132.000      RETURN
1208 - 1133.000      END
1209 - 1134.000      C! ASSIGN M:BB, (FILE, MWVEP0TD), (SAVE)
1210 - 1135.000      C! FORTTRAN LS, BB, ADP
1211 - 1136.000      SUBROUTINE WAVEFN(Z, ZION, ZR, RH0, H, IPTS, RMESH, RNL, DRNL,
1212 - 1137.000      1      MXFN, MXPTS, LMAX1, LOCCFN, LTOTFN, LHI1, NEL, FR, FRSP)
1213 - 1138.000      DIMENSION RMESH(1), RNL(1), DRNL(1), LOCCFN(1), LTOTFN(1)
1214 - 1139.000      DIMENSION NPRINC(15), ZETA(15), XNORM(15), C(15), FR(3,15)
1215 - 1139.500      DIMENSION FRSP(3,15)
1216 - 1140.000      REAL *8 ZZETA, ZNPRIN, ZC, ZNORMC, ZLOCC, ZLTOT
1217 - 1141.000      DATA ZZETA/' ZETA'//, ZNPRIN/'NPRINC'//, ZC/' C3'//
1218 - 1142.000      1      ZNORMC/' NORMC'//, ZLOCC/'LOCCFN'//, ZLTOT/'LTOTFN'//
1219 - 1143.000      FNORM(ZZ, NN)=SQRT((2.0*ZZ)**(2*NN+1)/FACT(2*NN))
1220 - 1144.000      MAXBAS=15
1221 - 1145.000      DO 1 I=1, LMAX1
1222 - 1146.000      LOCCFN(I)=0
1223 - 1147.000      1      LTOTFN(I)=0
1224 - 1148.000      READ(5,900) Z, ZION, ZR, RH0, H, IPTS, NOCC, NTOT
1225 - 1149.000      900 FORMAT(5F10.0, 3I5)
1226 - 1150.000      C      NOCC AND NTOT REFER TO CLOSED SUBSHELLS
1227 - 1151.000      WRITE(6,901) Z, ZION, ZR, RH0, H, IPTS, NOCC, NTOT
1228 - 1152.000      901 FORMAT(' ', 'Z =', F5.1, ' ZION =', F5.1, ' ZR =', F5.1,
1229 - 1153.000      1      ' RH0 =', F7.3, ' H =', F7.4, ' IPTS =', I4,
1230 - 1154.000      2      ' NOCC =', I3, ' NTOT =', I3)
1231 - 1155.000      IF (IPTS*LE*MXPTS) GO TO 2
1232 - 1156.000      WRITE(6,902) IPTS, MXPTS
1233 - 1157.000      902 FORMAT(' ', 'NUMBER OF POINTS USED TO REPRESENT A0 =', I6,
1234 - 1158.000      1      ' EXCEEDS DIMENSION =', I4)
1235 - 1159.000      CALL EXIT
1236 - 1160.000      2 IF (NTOT*GE*NOCC) GO TO 3
1237 - 1161.000      WRITE(6,903) NTOT, NOCC
1238 - 1162.000      903 FORMAT(' ', 'TOTAL NUMBER OF A0'S =', I3,
1239 - 1163.000      1      ' BUT NUMBER OF OCCUPIED A0'S =', I3)
1240 - 1164.000      CALL EXIT
1241 - 1165.000      3 IF (NTOT*LE*MXFN) GO TO 4
1242 - 1166.000      WRITE(6,904) NTOT, MXFN
1243 - 1167.000      904 FORMAT(' ', 'TOTAL NUMBER OF A0'S =', I3,
1244 - 1168.000      1      ' EXCEEDS DIMENSION =', I3)
1245 - 1169.000      CALL EXIT
1246 - 1170.000      4 RH0T=RH0
1247 - 1171.000      DO 5 I=1, IPTS, 4
1248 - 1172.000      R=EXP(RH0T)/ZR
1249 - 1173.000      RMESH(I)=R
1250 - 1174.000      RH0T=RH0T+4.0*H
1251 - 1175.000      IF (I.EQ.1) GO TO 5
1252 - 1176.000      DEL=(R-R4)/4.0
1253 - 1177.000      RMESH(I-3)=R4+DEL
1254 - 1178.000      RMESH(I-2)=R4+DEL+DEL
1255 - 1179.000      RMESH(I-1)=R-DEL

```

```

1256 = 1180.000      5 R4=R
1257 = 1182.000      LSAVE=(-1)
1258 = 1183.000      NLSUM=0
1259 = 1184.000      NLPTS=0
1260 = 1185.000      6 READ(5,905) L,NBASES,L0CFN,LT0FN
1261 = 1186.000      905 FORMAT(4I5)
1262 = 1187.000      WRITE(6,906) L,NBASES,L0CFN,LT0FN
1263 = 1188.000      906 FORMAT(/'0','L =',I2,' NBASES =',I3,' L0CFN =',I3,
1264 = 1189.000      1      ' LT0FN =',I3)
1265 = 1190.000      L1=L+1
1266 = 1191.000      IF (L1.LE.LMAX1) GO TO 7
1267 = 1192.000      WRITE(6,907) L1,LMAX1
1268 = 1193.000      907 FORMAT(' ',L+1 =',I2,' EXCEEDS DIMENSION =',I2)
1269 = 1194.000      CALL EXIT
1270 = 1195.000      7 IF (NBASES.LE.MAXBAS) GO TO 8
1271 = 1196.000      WRITE(6,908) NBASES,MAXBAS
1272 = 1197.000      908 FORMAT(' ',NUMBER OF BASIS FUNCTIONS =',I3,' EXCEEDS'
1273 = 1198.000      1      ' DIMENSION =',I3)
1274 = 1199.000      CALL EXIT
1275 = 1200.000      8 IF (L0CFN.LE.LT0FN.AND.LT0FN.GT.0) GO TO 9
1276 = 1201.000      WRITE(6,909) L0CFN,LT0FN,L
1277 = 1202.000      909 FORMAT(' ',NUMBER OF OCCUPIED FUNCTIONS =',I2,
1278 = 1203.000      1      ' BUT TOTAL NUMBER OF FUNCTIONS =',I2,' WITH L =',I2)
1279 = 1204.000      CALL EXIT
1280 = 1205.000      9 IF (L.GT.LSAVE) GO TO 10
1281 = 1206.000      WRITE(6,910) L,LSAVE
1282 = 1207.000      910 FORMAT(' ',L =',I2,' IS OUT OF SEQUENCE WITH PREVIOUS ',
1283 = 1208.000      1      'L =',I2)
1284 = 1209.000      CALL EXIT
1285 = 1210.000      10 LSAVE=L
1286 = 1212.000      L0CCFN(L1)=L0CFN
1287 = 1213.000      LT0TFN(L1)=LT0FN
1288 = 1214.000      READ(5,911) (ZETA(I),NPRINC(I),I=1,NBASES)
1289 = 1215.000      911 FORMAT(F8.0,I2,F8.0,I2,F8.0,I2,F8.0,I2,F8.0,I2,F8.0,I2,
1290 = 1216.000      1      F8.0,I2)
1291 = 1217.000      CALL OUT1(ZETA,NBASES,ZZETA)
1292 = 1218.000      CALL IOUT1(NPRINC,NBASES,ZNPRIN)
1293 = 1219.000      DO 11 I=1,NBASES
1294 = 1220.000      11 XNORM(I)=FNORM(ZETA(I),NPRINC(I))
1295 = 1221.000      DO 19 J=1,LT0FN
1296 = 1222.000      NLSUM=NLSUM+1
1297 = 1223.000      READ(5,912) (C(I),I=1,NBASES),FR(L1,J),FRSP(L1,J),RCUT
1298 = 1223.100      IF (FR(L1,J).LT+.00001) FR(L1,J)=1.
1299 = 1224.000      912 FORMAT(8F10.0)
1300 = 1225.000      WRITE(6,913) NLSUM
1301 = 1226.000      913 FORMAT(/'0',I2,' ORBITAL')
1302 = 1227.000      CALL OUT1(C,NBASES,ZC)
1303 = 1227.100      WRITE(6,920) FR(L1,J),FRSP(L1,J),RCUT
1304 = 1227.200      920 FORMAT('0','FRACTION OF WHOLE SHELL FILLED = ',E14.8/
1305 = 1227.300      11X,'FRACTION OF CURRENT SPIN IN SHELL FILLED = ',E14.8/
1306 = 1227.400      21X,'ORBITAL SET EQUAL ZERO AFTER R = ',E14.8/
1307 = 1227.500      31X,'NOTE IF R ABOVE EQUAL ZERO--NO CUTOFF')
1308 = 1228.000      DO 13 K=1,IPTS
1309 = 1229.000      SUM=0.0
1310 = 1230.000      DSUM=0.0
1311 = 1231.000      R=RMESH(K)
1312 = 1231.100      IF (RCUT.LT+.00001) GO TO 25
1313 = 1231.200      IF (R.GE.RCUT) GO TO 24
1314 = 1232.000      25 DO 12 I=1,NBASES
1315 = 1233.000      NPRINT=NPRINC(I)

```

```

1316 = 1234.000      TNPRIN=FLOAT(NPRINT)
1317 = 1235.000      ZETAT=ZETA(I)
1318 = 1236.000      TEMP=C(I)*XNORM(I)*R**(NPRINT-1)*EXP(0.0-ZETAT*R)
1319 = 1237.000      SUM=SUM+TEMP
1320 = 1238.000      12 DSUM=DSUM+(ZETAT**2+(TNPRIN*(TNPRIN=1.0)/R-2.0*ZETAT*TNPRIN)/R)
1321 = 1239.000      1
1322 = 1240.000      24 NLPTS=NLPTS+1
1323 = 1241.000      RNL(NLPTS)=SUM
1324 = 1242.000      13 DRNL(NLPTS)=DSUM+(FLOAT(L*L1)/R-2.0*Z)/R*SUM
1325 = 1243.000      K=1
1326 = 1244.000      S=0.0
1327 = 1245.000      KMIN=NLPTS-IPTS+3
1328 = 1246.000      DO 16 KK=KMIN,NLPTS,2
1329 = 1247.000      K=K+2
1330 = 1248.000      IF (K.NE.3) GO TO 14
1331 = 1249.000      R1=RMESH(1)
1332 = 1250.000      F1=(RNL(KK=2)*R1)**2
1333 = 1251.000      GO TO 15
1334 = 1252.000      14 R1=R3
1335 = 1253.000      F1=F3
1336 = 1254.000      15 R3=RMESH(K)
1337 = 1255.000      F3=(RNL(KK)*R3)**2
1338 = 1256.000      16 S=S+(R3-R1)/6.0*(F1+4.0*(RNL(KK=1)*RMESH(K=1))**2+F3)
1339 = 1257.000      SNORM=1.0/SQRT(S)
1340 = 1258.000      DO 17 I=1,NBASES
1341 = 1259.000      17 C(I)=SNORM*C(I)
1342 = 1260.000      KMIN=NLPTS-IPTS+1
1343 = 1261.000      DO 18 KK=KMIN,NLPTS
1344 = 1262.000      DRNL(KK)=SNORM*DRNL(KK)
1345 = 1263.000      18 RNL(KK)=SNORM*RNL(KK)
1346 = 1264.000      WRITE(6,914) SNORM
1347 = 1265.000      914 FORMAT('0','SNORM =',1PE12.5)
1348 = 1266.000      CALL OUT1(C,NBASES,ZNORMC)
1349 = 1267.000      19 CONTINUE
1350 = 1268.000      IF (NTOT=NLSUM) 20,21,6
1351 = 1269.000      20 WRITE(6,915) NTOT,NLSUM,L
1352 = 1270.000      915 FORMAT(' ',NUMBER OF AB'S ANTICIPATED =',I2,
1353 = 1271.000      1          ' NUMBER OF AB'S ENTERED =',I2,
1354 = 1272.000      2          ' L =',I2)
1355 = 1273.000      CALL EXIT
1356 = 1274.000      21 CALL IOUT1(L0CCFN,LMAX1,ZL0CC)
1357 = 1275.000      CALL IOUT1(LYT0TFN,LMAX1,ZLT0T)
1358 = 1276.000      LHI1=LSAVE+1
1359 = 1276.004      ANELSP=0.0
1360 = 1276.005      ANEL=0.0
1361 = 1276.010      DO 810 L1=1,LHI1
1362 = 1276.015      DO 800 J=1,L0CCFN(L1)
1363 = 1276.020      IF (L0CCFN(L1).EQ.0) GO TO 810
1364 = 1276.025      L=L1+1
1365 = 1276.026      ANELSP=ANELSP+(2.*FLOAT(L)+1.)*FRSP(L1,J)
1366 = 1276.030      800 ANEL=ANEL+2.*(2.*FLOAT(L)+1.)*FR(L1,J)
1367 = 1276.035      810 CONTINUE
1368 = 1276.036      WRITE(6,820) ANELSP
1369 = 1276.037      820 FORMAT(1X,'NUMBER OF INTERESTING SPIN ELECTRONS = ',F10.7)
1370 = 1276.040      NEL=ANEL
1371 = 1276.045      WRITE(6,813) ANEL
1372 = 1276.050      IF (ANEL=FLOAT(NEL)) 801,812,801
1373 = 1276.055      801 IF ((ANEL=FLOAT(NEL)).LT.0.001) GO TO 812
1374 = 1276.060      IF ((ANEL=FLOAT(NEL)).GT.999) GO TO 802
1375 = 1276.065      CALL EXIT

```

```

1376 - 1276.070 813 FORMAT('O','ANEL=',F10.7,'IN WAVEFN')
1377 - 1276.075 802 NEL=NEL+1
1378 - 1277.000 812 WRITE(6,916) NEL
1379 - 1278.000 916 FORMAT('O','ORBITALS ACCOMMODATE',I3,'ELECTRONS ABOUT THIS SITE')
1380 - 1279.000 ACZION=Z=FLOAT(NEL)
1381 - 1280.000 IF (ACZION.EQ.ZION) GO TO 22
1382 - 1281.000 WRITE(6,917) ZION,ACZION
1383 - 1282.000 917 FORMAT(' ',ZION ANTICIPATED =',F5.1,' ACTUAL ZION =',F5.1)
1384 - 1283.000 CALL EXIT
1385 - 1284.000 22 I0CC=0
1386 - 1285.000 DO 23 I=1,LMAX1
1387 - 1286.000 23 I0CC=I0CC+L0CCFN(I)
1388 - 1287.000 IF (I0CC.EQ.N0CC) RETURN
1389 - 1288.000 WRITE(6,918) N0CC,I0CC
1390 - 1289.000 918 FORMAT(' ',N0CC ANTICIPATED =',I2,' ACTUAL N0CC =',I2)
1391 - 1290.000 CALL EXIT
1392 - 1291.000 END
1393 - 1292.000 SUBROUTINE CKALPH(IFILE,NREC,ALPHA,MXPPTS,MXPAN,LLGENA)
1394 - 1293.000 DIMENSION ALPHA(MXPPTS,MXPAN)
1395 - 1294.000 LOGICAL LLGENA
1396 - 1295.000 IF (.NOT.LLGENA) GO TO 2
1397 - 1296.000 DO 1 I=1,MXPAN
1398 - 1297.000 DO 1 J=1,MXPPTS
1399 - 1298.000 1 ALPHA(J,I)=0.0
1400 - 1298.500 C CHANGE HERE FOR REAL*8 ALPHAS
1401 - 1299.000 2 IALPTS=MXPPTS*MXPAN
1402 - 1300.000 IUNLM=NREC
1403 - 1301.000 IF (LLGENA) GO TO 3
1404 - 1302.000 CCCCC *****
1405 - 1303.000 IUNLM=IALPTS*(IUNLM-1)+1
1406 - 1304.000 READ DISK IFILE,IUNLM,ALPHA
1407 - 1305.000 IUNLM=(IUNLM-1)/IALPTS+1
1408 - 1306.000 CCCCC *****
1409 - 1307.000 CCCCC *****
1410 - 1308.000 C READ(IFILE,IUNLM) ALPHA
1411 - 1309.000 CCCCC *****
1412 - 1310.000 GO TO 4
1413 - 1311.000 3 CONTINUE
1414 - 1312.000 CCCCC *****
1415 - 1313.000 IUNLM=IALPTS*(IUNLM-1)+1
1416 - 1314.000 WRITE DISK IFILE,IUNLM,ALPHA
1417 - 1315.000 IUNLM=(IUNLM-1)/IALPTS+1
1418 - 1316.000 CCCCC *****
1419 - 1317.000 CCCCC *****
1420 - 1318.000 C WRITE (IFILE,IUNLM) ALPHA
1421 - 1319.000 CCCCC *****
1422 - 1320.000 4 WRITE(6,900) IFILE,NREC,MXPPTS,MXPAN,LLGENA
1423 - 1321.000 900 FORMAT('O','FILE',I3,' CHECKED FOR',I4,' ALPHA(',I3,' ',I2,')')
1424 - 1322.000 1 ' BLOCKS LLGENA =',I2/)
1425 - 1323.000 RETURN
1426 - 1324.000 END
1427 - 1325.000 SUBROUTINE VNC(RNL,LMAX1,L0CCFN,L0TFN,RMESH,IPTS,VNCFN,FR)
1428 - 1326.000 DIMENSION RNL(1),L0CCFN(1),L0TFN(1),RMESH(1),VNCFN(1)
1429 - 1327.000 DIMENSION SRNLSQ(161),FR(3,15)
1430 - 1328.000 MXPPTS=161
1431 - 1329.000 IF (IPTS.LE.MXPPTS) GO TO 100
1432 - 1330.000 WRITE(6,900) IPTS,MXPPTS
1433 - 1331.000 900 FORMAT(' ',IPTS =',I4,' EXCEEDS MXPPTS =',I4,' IN SUBROUTINE VNC')
1434 - 1332.000 100 INXFN=0
1435 - 1333.000 DO 1 I=1,IPTS

```

AD-A037 797

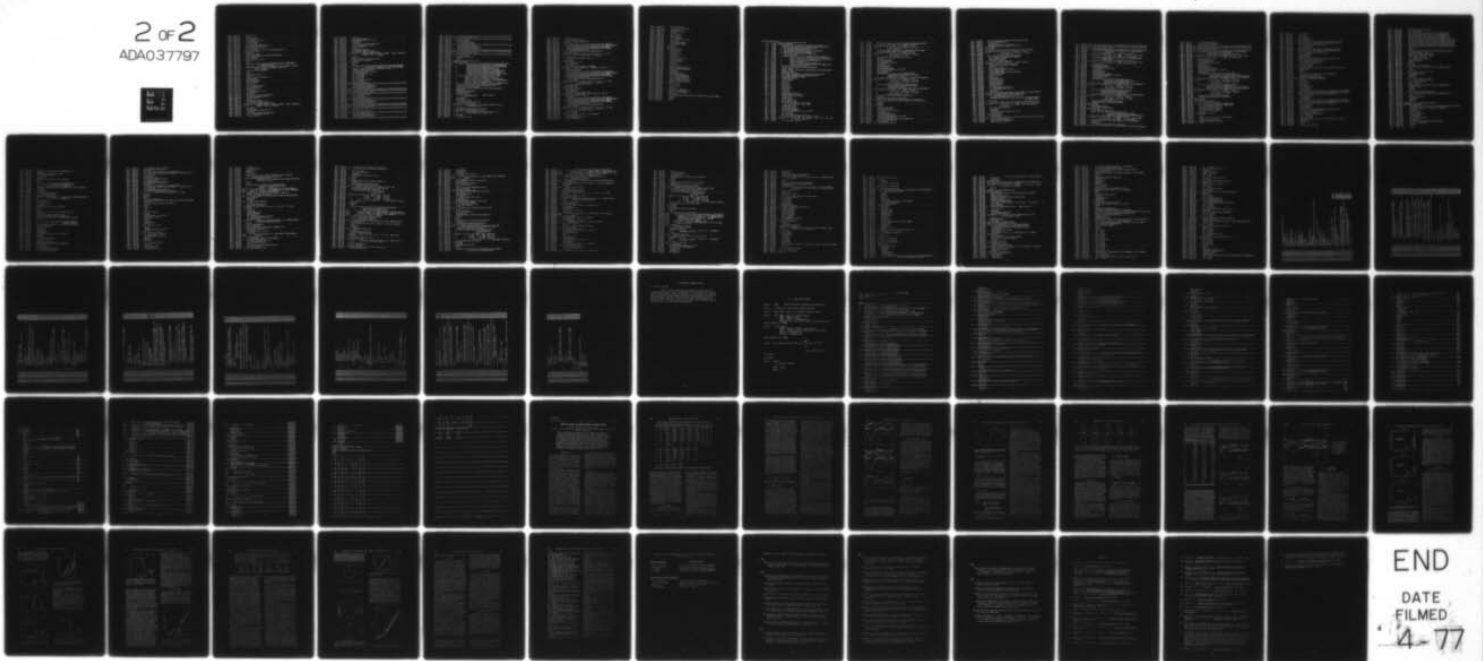
ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF PHYSICS  
THE ELECTRONIC STRUCTURE OF CRYSTALLINE SOLIDS: THEIR SURFACES --ETC(U)  
FEB 77 A KUNZ

F/G 20/11  
F33615-72-C-1506  
NL

UNCLASSIFIED

AFAL-TR-76-140

2 of 2  
ADA037797



END  
DATE  
FILMED  
4-77



```

1436 = 1334.000      1 SRNLSQ(I)=0.0
1437 = 1335.000      ANEL=0.
1438 = 1336.000      DO 4 L1=1,LMAX1
1439 = 1337.000      LTOFN=LTOFN(L1)
1440 = 1338.000      IF (LTOFN.EQ.0) GO TO 4
1441 = 1339.000      LCCFN=LCCFN(L1)
1442 = 1340.000      DO 3 J=1,LTOFN
1443 = 1341.000      IF (J.GT.LCCFN) GO TO 3
1444 = 1342.000      ANEL=ANEL+FLOAT(*L1=2)*FR(L1,J)
1445 = 1343.000      WT=FLOAT(*L1=4)*FR(L1,J)
1446 = 1344.000      DO 2 I=1,IPTS
1447 = 1345.000      2 SRNLSQ(I)=SRNLSQ(I)+WT*(RNL(INXFN+I)*RMESH(I))*2
1448 = 1346.000      3 INXFN=INXFN+IPTS
1449 = 1347.000      4 CONTINUE
1450 = 1348.000      TWONEL=0.**2.*ANEL
1451 = 1349.000      CALL IR12K(SRNLSQ,0,VNCFN,RMESH,IPTS,TWONEL)
1452 = 1350.000      TAIL=VNCFN(IPTS)
1453 = 1351.000      DO 5 I=1,IPTS
1454 = 1352.000      5 VNCFN(I)=VNCFN(I)-TAIL
1455 = 1353.000      RETURN
1456 = 1354.000      END
1457 = 1355.000      SUBROUTINE PTION(Z,AMD LNG,AXDIST,NAXSET,AADIST,NAASET,NBETS,
1458 = 1356.000      1 VCZFN,RMESH,IPTS)
1459 = 1357.000      C SUBROUTINE PTION STORES LONG-RANGE POINT-ION POTENTIAL IN VCZFN
1460 = 1358.000      DIMENSION AXDIST(1),NAXSET(1),AADIST(1),NAASET(1),
1461 = 1359.000      1 VCZFN(1),RMESH(1)
1462 = 1360.000      IF (ABS(Z).LT.0.5) RETURN
1463 = 1361.000      MUA=0
1464 = 1362.000      MUX=1
1465 = 1363.000      VO=Z*AMD LNG/AXDIST(1)
1466 = 1364.000      ZSUM=0.0
1467 = 1365.000      ZSHELL=0.0=Z*FLOAT(NAXSET(1))
1468 = 1366.000      BOUND=AXDIST(1)
1469 = 1367.000      BOUNDA=AADIST(1)
1470 = 1368.000      BOUNDX=AXDIST(2)
1471 = 1369.000      DO 10 I=1,IPTS
1472 = 1370.000      R=RMESH(I)
1473 = 1371.000      IF (R.LT.BOUND) GO TO 9
1474 = 1372.000      VO=VO+ZSHELL/BOUND
1475 = 1373.000      ZSUM=ZSUM+ZSHELL
1476 = 1374.000      ZSHELL=0.0
1477 = 1375.000      IF (BOUNDX=BOUND) 1,4,6
1478 = 1376.000      1 BOUND=BOUNDX
1479 = 1377.000      MUX=MUX+1
1480 = 1378.000      ZSHELL=ZSHELL+Z*FLOAT(NAXSET(MUX))
1481 = 1379.000      IF (MUX+1.LE.NAXSET) GO TO 2
1482 = 1380.000      BOUNDX=1.0E+08
1483 = 1381.000      GO TO 9
1484 = 1382.000      2 BOUNDX=AXDIST(MUX+1)
1485 = 1383.000      IF (BOUNDX.GT.BOUND) GO TO 9
1486 = 1384.000      3 WRITE(6,900) BOUND,MUX,BOUNDX,MUX
1487 = 1385.000      900 FORMAT(' ',SEQUENCE ERROR IN SHELLS: BOUNDX ('I8') @',BBE14.7)
1488 = 1386.000      1 ' BOUNDX ('I2,'+1) =',E14.7)
1489 = 1387.000      CALL EXIT
1490 = 1388.000      4 BOUND=BOUNDX
1491 = 1389.000      IF (BOUND.EQ.1.0E+08) GO TO 8
1492 = 1390.000      MUX=MUX+1
1493 = 1391.000      ZSHELL=ZSHELL+Z*FLOAT(NAXSET(MUX))
1494 = 1392.000      IF (MUX+1.LE.NAXSET) GO TO 5
1495 = 1393.000      BOUNDX=1.0E+08

```

```

1496 - 1394.000      GO TO 6
1497 - 1395.000      5 BOUNDX=AXDIST(MUX+1)
1498 - 1396.000      IF (BOUNDX.LE.BOUND) GO TO 3
1499 - 1397.000      6 BOUND=BOUNDA
1500 - 1398.000      MUA=MUA+1
1501 - 1399.000      ZSHELL=ZSHELL+Z*FLOAT(NAASET(MUA))
1502 - 1400.000      IF (MUA+1.LE.NSETS) GO TO 7
1503 - 1401.000      BOUND=1.0E+08
1504 - 1402.000      GO TO 9
1505 - 1403.000      7 BOUNDA=AADIST(MUA+1)
1506 - 1404.000      IF (BOUNDA.GT.BOUND) GO TO 9
1507 - 1405.000      WRITE(6,901) BOUND,MUA,BOUNDA,MUA
1508 - 1406.000      901 FORMAT(' ','SEQUENCE ERROR IN SHELLS. BOUND ('',I8,'') =',BBE14.7,
1509 - 1407.000      1      ' BOUNDA ('',I2,'') =',E14.7)
1510 - 1408.000      CALL EXIT
1511 - 1409.000      8 VO=0.0
1512 - 1410.000      ZSUM=0.0+RMESH(I=1)+VCZFN(I=1)/2.0
1513 - 1411.000      9 VCZFN(I)=2.0*(VO-ZSUM/R)
1514 - 1412.000      10 CONTINUE
1515 - 1413.000      RETURN
1516 - 1414.000      END
1517 - 1415.000      SUBROUTINE IOVCZ(LLGENA,LLAGA,LLAOX,VCZAFN,VCZXFN,IAPTS,IXPTS,
1518 - 1416.000      1      ALPHA,MXPTS,MXPAN,IAAFLE,IAAUNM,IAXFLE,IXXUNM)
1519 - 1417.000      DIMENSION VCZAFN(1),VCZXFN(1),ALPHA(MXPTS,MXPAN)
1520 - 1418.000      LOGICAL LLGENA,LLAGA,LLAOX
1521 - 1418.500      C CHANGE HERE FOR REAL*8 ALPHAS
1522 - 1419.000      IALPTS=MXPTS*MXPAN
1523 - 1420.000      IF (.NOT.LLGENA) GO TO 5
1524 - 1421.000      DO 1 K1=1,MXPAN
1525 - 1422.000      DO 1 I=1,MXPTS
1526 - 1423.000      1 ALPHA(I,K1)=0.0
1527 - 1424.000      IF (.NOT.LLAGA) GO TO 3
1528 - 1425.000      DO 2 I=1,IAPTS
1529 - 1426.000      2 ALPHA(I,1)=VCZAFN(I)
1530 - 1427.000      CCCCC *****
1531 - 1428.000      IAAUNM=IALPTS*(IAAUNM=1)+1
1532 - 1429.000      WRITE DISK IAAFLE,IAAUNM,ALPHA
1533 - 1430.000      IAAUNM=(IAAUNM=1)/IALPTS+1
1534 - 1431.000      CCCCC *****
1535 - 1432.000      CCCCC *****
1536 - 1433.000      C WRITE(IAAFLE,IAAUNM) ALPHA
1537 - 1434.000      CCCCC *****
1538 - 1435.000      3 IF (.NOT.LLAOX) RETURN
1539 - 1436.000      DO 4 I=1,IXPTS
1540 - 1437.000      4 ALPHA(I,1)=VCZXFN(I)
1541 - 1438.000      CCCCC *****
1542 - 1439.000      IXXUNM=IALPTS*(IXXUNM=1)+1
1543 - 1440.000      WRITE DISK IXXFLE,IXXUNM,ALPHA
1544 - 1441.000      IXXUNM=(IXXUNM=1)/IALPTS+1
1545 - 1442.000      CCCCC *****
1546 - 1443.000      CCCCC *****
1547 - 1444.000      C WRITE(IXXFLE,IXXUNM) ALPHA
1548 - 1445.000      CCCCC *****
1549 - 1446.000      RETURN
1550 - 1447.000      5 IF (.NOT.LLAGA) GO TO 7
1551 - 1448.000      CCCCC *****
1552 - 1449.000      IAAUNM=IALPTS*(IAAUNM=1)+1
1553 - 1450.000      READ DISK IAAFLE,IAAUNM,ALPHA
1554 - 1451.000      IAAUNM=(IAAUNM=1)/IALPTS+1
1555 - 1452.000      CCCCC *****

```

```

1556 = 1453.000 CCCCC *****
1557 = 1454.000 C READ (IAAFLE,IAAUNM) ALPHA
1558 = 1455.000 CCCCC *****
1559 = 1456.000 DO 6 I=1,IAPTS
1560 = 1457.000 6 VCZAFN(I)=ALPHA(I,1)
1561 = 1458.000 7 IF (.NOT.LLQX) RETURN
1562 = 1459.000 CCCCC *****
1563 = 1460.000 IXXUNM=IALPTS*(IXXUNM=1)+1
1564 = 1461.000 READ DISK IXXFLE,IXXUNM,ALPHA
1565 = 1462.000 IXXUNM=(IXXUNM=1)/IALPTS+1
1566 = 1463.000 CCCCC *****
1567 = 1464.000 CCCCC *****
1568 = 1465.000 C READ (IXXFLE,IXXUNM) ALPHA
1569 = 1466.000 CCCCC *****
1570 = 1467.000 DO 8 I=1,IXPTS
1571 = 1468.000 8 VCZXFN(I)=ALPHA(I,1)
1572 = 1469.000 RETURN
1573 = 1470.000 END
1574 = 1471.000 C:ASSIGN M:BO,(FILE,MMTRXD),(SAVE)
1575 = 1472.000 C:FORTRAN LS,BO,ADP
1576 = 1473.000 SUBROUTINE MTRX(NA,NX,LA,LX,FRA,FRX,LOCAFN,LOCXFN,LARINX,LXRINX,
1577 = 1474.000 1 LADINX,LXDINX,LAMAX1,LXMAX1,CA,CX,MBLCHA,MBLCHX,
1578 = 1475.000 2 NBLCHA,NBLCHX,RANL,RXNL,DRANL,DRXNL,ALPHA,MXPT,
1579 = 1476.000 3 MXPAN,NXPAN,RAMESH,RXMESH,IAPTS,IXPTS,
1580 = 1477.000 4 RU,RUB,UNIT,NAASET,MAXSET,NEIGHA,NEIGHX,
1581 = 1478.000 5 NEIGAC,NEIGXC,KLETE,NLETE,XXY,XY,XXZ,SWT,NVEC,
1582 = 1478.500 6 NAG,NXG,XAF,XXF,GA,GX,XAG,XXG,FRASP,FRXOP,
1583 = 1479.000 7 LLGENI,IIFILE,IAAFLE,IXXFLE,IAFLE,IXAFLE,IVFLE,
1584 = 1480.000 8 LLNAGA,LLNAGX,LLDTAL,LLBUF,NROUT,LLMTRX)
1585 = 1481.000 DIMENSION NA(1),NX(1),LA(1),LX(1),ERA(3,15),FRX(3,15),
1586 = 1482.000 1 LOCAFN(1),LOCXFN(1),LARINX(1),LXRINX(1),
1587 = 1483.000 2 LADINX(1),LXDINX(1),CA(MBLCHA,1),CX(MBLCHX,1),
1588 = 1484.000 3 RANL(1),RXNL(1),DRANL(1),DRXNL(1),ALPHA(MXPT,1),
1589 = 1485.000 4 RAMESH(1),RXMESH(1),RU(1),RUB(1),
1590 = 1486.000 5 NAASET(1),MAXSET(1),KLETE(1),FRASP(3,15),FRXOP(3,15)
1591 = 1486.500 6,NAG(3),NXG(3),XAF(3,25),XAG(3,25),XXF(3,25),XXG(3,25)
1592 = 1487.000 LOGICAL LLGENI,LLNAGA,LLNAGX,LLDTAL,LLBUF,LLMTRX
1593 = 1487.500 INTEGER GA(3,25,2),GX(3,25,2)
1594 = 1488.000 COMPLEX CA,CX,H,S,HH(595),SS(595),DUMH(1156),DUMS(1156),
1595 = 1488.500 1DUMS2(1156)
1596 = 1488.600 COMMON DUMS,DUMS2,DUMH
1597 = 1489.000 DIMENSION ALPHA2(1)
1598 = 1490.000 DATA DUMMY/0.0/
1599 = 1491.000 C DIMENSION SS,HH(N*(N+1)/2) WHERE N=MBLHAX
1600 = 1492.000 C ALPHA2(IALPTT)
1601 = 1493.000 IALPTT=4830
1602 = 1494.000 MBLHAX=34
1603 = 1495.000 IALPTS=MXPTS*MXPAN
1604 = 1496.000 IF (IALPTS.LE.IALPTT) GO TO 51
1605 = 1497.000 WRITE(6,899) IALPTS,IALPTT
1606 = 1498.000 899 FORMAT(' ',MXPTS*MXPAN=' ',I5,' EXCEEDS DIMENSION ',I5,
1607 = 1499.000 1 ' IN SUBPROGRAM MTRX')
1608 = 1500.000 CALL EXIT
1609 = 1501.000 51 IF (NBLCHA+NBLCHX.LE.MBLHAX) GO TO 52
1610 = 1502.000 WRITE(6,900) NBLCHA,NBLCHX,MBLHAX
1611 = 1503.000 900 FORMAT(' ',SUM OF NBLCHA=' ',I3,' AND NBLCHX=' ',I3,
1612 = 1504.000 1 ' EXCEEDS DIMENSION MBLHAX=' ',I3)
1613 = 1505.000 CALL EXIT
1614 = 1506.000 52 CONTINUE
1615 = 1507.000 IF (LLNAGA) GO TO 57

```

```

1616 - 1508.000      INXHS=0
1617 - 1509.000      DO 54 J=1,NBLCHA
1618 - 1510.000      IF (LLGENI) WRITE(6,910) J
1619 - 1511.000      910 FORMAT(' ', 'BEGIN COLUMN', I3)
1620 - 1512.000      DO 54 I=1, J
1621 - 1513.000      CALL AXELEM(.TRUE., I, J, NA, NA, LA, LA, FRA, FRA, LOCAFN, LOCAFN,
1622 - 1514.000      1  LARINX, LARINX, LADINX, LADINX, LAMAX1, LAMAX1, CA, CA, NBLCHA, NBLCHA,
1623 - 1515.000      2  RANL, RANL, DRANL, DRANL, ALPHA, ALPHA2, MXPPTS, MXPAN, NXBAN,
1624 - 1516.000      3  RAMESH, RAMESH, IAPTS, IAPTS, RU, UNIT, NAASET, NEIGHA, NEIGAC,
1625 - 1516.500      4  NAG, XAF, GA, XAG, FRASP, FRASP,
1626 - 1517.000      5  XKX, XKY, XKZ, LLGENI, IIFILE, IAAFLE, IAAFLE, H, S,
1627 - 1518.000      6  LLDTAL, LLBUF)
1628 - 1522.000      925 FORMAT('O', 'FOR AA MATRIX ELEMENT  I =', I3, '  J =', I3,
1629 - 1523.000      1  '  H =', 1P2E14.6, '  S =', 2E14.6//)
1630 - 1525.000      INXHS=INXHS+1
1631 - 1526.000      IF (LLDTAL) WRITE(6, 925) I, J, H, S
1632 - 1527.000      SS(INXHS)=S
1633 - 1528.000      54 HH(INXHS)=H
1634 - 1529.000      IF (LLNABX) GO TO 60
1635 - 1530.000      DO 56 J=1,NBLCHX
1636 - 1531.000      JAX=J+NBLCHA
1637 - 1532.000      IF (LLGENI) WRITE(6, 9255) JAX, NBLCHA
1638 - 1533.000      9255 FORMAT(' ', 'BEGIN COLUMN', I3, ' (FIRST', I3, ' ROWS)')
1639 - 1534.000      INXHSJ=JAX*(JAX-1)/2
1640 - 1535.000      DO 56 I=1,NBLCHA
1641 - 1536.000      CALL AXELEM(.FALSE., I, J, NA, NX, LA, LX, FRA, FRX, LOCAFN, LOCFXN,
1642 - 1537.000      1  LARINX, LXRINX, LADINX, LXDINX, LAMAX1, LXMAY1, CA, CX, NBLCHA, NBLCHX,
1643 - 1538.000      2  RANL, RXNL, DRANL, DRXNL, ALPHA, ALPHA2, MXPPTS, MXPAN, NXBAN,
1644 - 1539.000      3  RAMESH, RXMESH, IAPTS, IXPTS, RUB, UNIT, NAXSET, NEIGHX, NEIGXC,
1645 - 1539.500      4  NAG, XAF, GA, XAG, FRASP, FRXSP,
1646 - 1540.000      5  XKX, XKY, XKZ, LLGENI, IIFILE, IAXFLE, IAXFLE, H, S,
1647 - 1541.000      6  LLDTAL, LLBUF)
1648 - 1545.000      926 FORMAT('O', 'FOR AX MATRIX ELEMENT  I =', I3, '  J =', I3,
1649 - 1546.000      1  '  H =', 1P2E14.6, '  S =', 2E14.6//)
1650 - 1548.000      IF (LLDTAL) WRITE(6, 926) I, J, H, S
1651 - 1549.000      SS(INXHSJ+I)=S
1652 - 1550.000      56 HH(INXHSJ+I)=H
1653 - 1551.000      57 IF (LLNABX) GO TO 60
1654 - 1552.000      DO 59 J=1,NBLCHX
1655 - 1553.000      JAX=J+NBLCHA
1656 - 1554.000      IF (LLGENI) WRITE(6, 9265) JAX, J
1657 - 1555.000      9265 FORMAT(' ', 'FINISH COLUMN', I3, ' (LAST', I3, ' ROWS)')
1658 - 1556.000      INXHSJ=JAX*(JAX-1)/2+NBLCHA
1659 - 1557.000      DO 59 I=1, J
1660 - 1558.000      CALL AXELEM(.TRUE., I, J, NX, NX, LX, LX, FRA, FRX, LOCAFN, LOCFXN,
1661 - 1559.000      1  LXRINX, LXRINX, LXDINX, LXDINX, LXMAY1, LXMAY1, CX, CX, NBLCHA, NBLCHX,
1662 - 1560.000      2  RXNL, RXNL, DRXNL, DRXNL, ALPHA, ALPHA2, MXPPTS, MXPAN, NXBAN,
1663 - 1561.000      3  RXMESH, RXMESH, IXPTS, IXPTS, RU, UNIT, NAASET, NEIGHA, NEIGAC,
1664 - 1561.500      4  NXG, XXF, GX, XXG, FRXSP, FRXSP,
1665 - 1562.000      5  XKX, XKY, XKZ, LLGENI, IIFILE, IXXFLE, IXXFLE, H, S,
1666 - 1563.000      6  LLDTAL, LLBUF)
1667 - 1567.000      927 FORMAT('O', 'FOR XX MATRIX ELEMENT  I =', I3, '  J =', I3,
1668 - 1568.000      1  '  H =', 1P2E14.6, '  S =', 2E14.6//)
1669 - 1570.000      IF (LLDTAL) WRITE(6, 927) I, J, H, S
1670 - 1571.000      SS(INXHSJ+I)=S
1671 - 1572.000      59 HH(INXHSJ+I)=H
1672 - 1573.000      60 CALL IO(LLGENI, IIFILE, DUMMY, .TRUE., LLBUF)
1673 - 1574.000      N=NBLCHA+NBLCHX
1674 - 1575.000      IF (NLETE.EQ.0) GO TO 63
1675 - 1576.000      DO 62 ILETE=1, NLETE

```

```

1676 - 1577.000      K=KLETE(ILETE)
1677 - 1578.000      IF (K.EQ.N) GO TO 62
1678 - 1579.000      KT0=K*(K+1)/2
1679 - 1580.000      KFR0M=K*(K+1)/2
1680 - 1581.000      K1=K+1
1681 - 1582.000      DO 61 J=K1,N
1682 - 1583.000      DO 61 I=1,J
1683 - 1584.000      KFR0M=KFR0M+1
1684 - 1585.000      IF (I.EQ.K) GO TO 61
1685 - 1586.000      KT0=KT0+1
1686 - 1587.000      SS(KT0)=SS(KFR0M)
1687 - 1588.000      HH(KT0)=HH(KFR0M)
1688 - 1589.000      61 CONTINUE
1689 - 1590.000      62 N=N-1
1690 - 1590.010      63 K1=N*(N+1)/2
1691 - 1590.020      DO 10 JP=1,N
1692 - 1590.030      J=N-JP+1
1693 - 1590.040      DO 10 IP=1,N
1694 - 1590.050      I=N-IP+1
1695 - 1590.060      K=N*(J-1)+I
1696 - 1590.070      IF (I.GT.J) GO TO 9
1697 - 1590.080      DUMH(K)=HH(K1)
1698 - 1590.090      DUMS(K)=SS(K1)
1699 - 1590.100      K1=K1-1
1700 - 1590.110      GO TO 10
1701 - 1590.120      9 K2=N*(I-1)+J
1702 - 1590.130      DUMH(K)=CONJG(DUMH(K2))
1703 - 1590.140      DUMS(K)=CONJG(DUMS(K2))
1704 - 1590.150      10 CONTINUE
1705 - 1590.160      LINDIM=N*N
1706 - 1590.170      ISTART=LINDIM+1
1707 - 1590.180      MATMAX=MBLHAX*MBLHAX
1708 - 1590.190      DO 11 I=ISTART,MATMAX
1709 - 1590.200      DUMH(I)=CMPLX(0.0,0.0)
1710 - 1590.210      DUMS(I)=CMPLX(0.0,0.0)
1711 - 1590.220      11 DUMS2/I)=CMPLX(0.0,0.0)
1712 - 1590.230      DO 12 I=1,N
1713 - 1590.240      DO 12 J=1,N
1714 - 1590.250      K=N*(J-1)+I
1715 - 1590.260      REL=0.0
1716 - 1590.270      IF (I.EQ.J) REL=1.0
1717 - 1590.280      12 DUMS2(K)=CMPLX(REL,0.0)
1718 - 1590.290      WRITE(23) DUMH,DUMS
1719 - 1590.300      REWIND 23
1720 - 1591.000      CALL EVDRVE(N,XKX,XKY,XKZ,IWT,NVEC,IEVFLE,LLDTAL,NR0UT,
1721 - 1591.500      1LLMTRX)
1722 - 1592.000      RETURN
1723 - 1593.000      END

```

COPY HCMPLX2 TO LP(K)

```

1 - 1.000 C IASSIGN M:BO, (FILE, MAXELEM), (SAVE)
2 - 2.000 C I FORTRAN LS,BO,ADP
3 - 3.000 SUBROUTINE AXELEM(LLSAME,IA,JX,NA,NX,LA,LX,FRA,FRX,
4 - 4.000 1 LOCAFN,LOCXFN,LARINX,LXRINX,LADINX,LXDINX,LAMAX1,LMAX1,
5 - 5.000 2 CA,CX,MBLCHA,MBLCHX,RANL,RXNL,DRANL,DRXNL,
6 - 6.000 3 ALPHA,ALPHA2,MXPTS,MXPAN,NXPAN,
7 - 7.000 4 RAMESH,RXMESH,IAPTS,IXPTS,RUB,UNIT,NAXSET,NSETS,
8 - 8.000 5 NSETSC,NAG,XAF,GA,XAG,FRASP,FRXSP,
9 - 8.500 6 XKX,XKY,XKZ,LLGENI,IIFILE,IAXFLE,IXAFLE,
10 - 9.000 7 H,S,LLDTAL,LLBUF)
11 - 10.000 DIMENSION NA(1),NX(1),LA(1),LX(1),FRA(3,15),FRX(3,15),
12 - 11.000 1 LOCAFN(1),LOCXFN(1),LARINX(1),LXRINX(1),LADINX(1),
13 - 12.000 2 LXDINX(1),CA(MBLCHA,1),CX(MBLCHX,1),RANL(1),RXNL(1),
14 - 13.000 3 DRANL(1),DRXNL(1),ALPHA(MXPTS,MXPAN),
15 - 14.000 4 ALPHA2(MXPTS,MXPAN),
16 - 15.000 5 RAMESH(1),RXMESH(1),RUB(1),NAXSET(1),
17 - 15.100 6 NAG(3),XAF(3,25),XAG(3,25),FRASP(3,15),FRXSP(3,15)
18 - 15.200 INTEGER GA(3,25,2),GATEM1,GATEM2
19 - 16.000 COMPLEX CA,CX,H,S,ZSUM,Z,ZCROT,M,ZCRM,ZKCRM
20 - 17.000 REAL*4 IMZCRM
21 - 18.000 LOGICAL LLSAME,LLGENI,LLAOC,LLXOC,LLANOC,LLXNOC,LLDTAL,LLBUF
22 - 18.500 LOGICAL LLCNFG
23 - 19.000 DIMENSION DUMMY(2)
24 - 20.000 EQUIVALENCE (ZCRM,DUMMY(1),REZCRM),(DUMMY(2),IMZCRM)
25 - 21.000 DATA TWOPI/6.283185307179586/,EPSLON/1.0E-05/
26 - 21.500 C CHANGE HERE FOR REAL*8 ALPHAS
27 - 22.000 IALPTS=MXPTS*MXPAN
28 - 22.500 LLCNFG=.FALSE.
29 - 23.000 H=(0.0,0.0)
30 - 24.000 S=(0.0,0.0)
31 - 24.000 NAI=NA(IA)
32 - 24.000 LAI=LA(IA)
33 - 25.000 LAI1=LAI+1
34 - 26.000 MATOT=2*LAI+1
35 - 27.000 NXJ=NX(JX)
36 - 28.000 LXJ=LX(JX)
37 - 29.000 LXJ1=LXJ+1
38 - 30.000 MMAX1=MINO(LAI,LXJ)+1
39 - 31.000 IASEQ=LARINX(LAI1)+NAI-LAI
40 - 32.000 JXSEQ=LXRINX(LXJ1)+NXJ-LXJ
41 - 33.000 LLAOC=.FALSE.
42 - 34.000 LLXOC=.FALSE.
43 - 35.000 LOCCAI=LOCAFN(LAI1)
44 - 36.000 LOCCXJ=LOCXFN(LXJ1)
45 - 37.000 IF (NAI-LAI.LE.LOCCAI) LLAOC=.TRUE.
46 - 38.000 IF (NXJ-LXJ.LE.LOCCXJ) LLXOC=.TRUE.
47 - 39.000 LLANOC=.NOT.LLAOC
48 - 40.000 LLXNOC=.NOT.LLXOC
49 - 41.000 INXAI=IALPTS*(IASEQ-1)
50 - 42.000 INXJ=IXPTS*(JXSEQ-1)
51 - 43.000 IF (.NOT.LLSAME.OR.LAI.NE.LXJ) GO TO 6
52 - 44.000 ZSUM=(0.0,0.0)
53 - 45.000 DO 1 I=1,MATOT
54 - 46.000 1 ZSUM=ZSUM+CONJG(CA(JX,I))*CA(IA,I)
55 - 47.000 IF (LLDTAL) WRITE(6,888) NXJ,LXJ,NAI,LAI,ZSUM
56 - 48.000 888 FORMAT(' ',NXJ='I2',' LXJ='I2',' NAI='I2',' LAI='I2,
57 - 49.000 1 ZSUM='1P2E14.7)
58 - 50.000 IF (.NOT.LLGENI) GO TO 4

```

```

59 - 71.000      M=0
60 - 72.000      OCT1=EL1CT1(RXNL,INXXJ,XNULL,RANL,INXAI,RAMESH,IAPTS,.FALSE.)
61 - 73.000      IF (LLDTAL) WRITE(6,890) NXJ,LXJ,M,NAI,LAI,M,OCT1
62 - 74.000      890 FORMAT('0','NXJ=',I2,' LXJ=',I2,' M=',I2,
63 - 75.000      1      ' NAI=',I2,' LAI=',I2,' M=',I2,
64 - 76.000      2      ' OCT1=',IPE14.6)
65 - 77.000      F=EL1CT1(RXNL,INXXJ,XNULL,DRANL,INXAI,RAMESH,IAPTS,.FALSE.)
66 - 78.000      IF (LLDTAL) WRITE(6,892) NXJ,LXJ,M,NAI,LAI,M,F
67 - 79.000      892 FORMAT('0','NXJ=',I2,' LXJ=',I2,' M=',I2,
68 - 80.000      1      ' NAI=',I2,' LAI=',I2,' M=',I2,
69 - 81.000      2      ' F1U=',IPE14.6)
70 - 81.100      IF (.NOT.LLDTAL.AND..NOT.LLCNFG) GO TO 104
71 - 81.200      FSAVE=F
72 - 82.000      DO 3 LAK1=1,LAMAX1
73 - 83.000      LCCAK=LCCAFN(LAK1)
74 - 84.000      IF (LCCAK.EQ.0) GO TO 3
75 - 85.000      LAK=LAK1+1
76 - 86.000      NAKMAX=LCCAK+LAK
77 - 87.000      MAKTOT=2*LAK+1
78 - 88.000      DO 2 NAK=LAK1,NAKMAX
79 - 89.000      INXAK=IAPTS*(LARINX(LAK1)+NAK-LAK1)
80 - 90.000      DO 2 MMAK=1,MAKTOT
81 - 91.000      MAK=MMAK-LAK1
82 - 92.000      VCCT1=EL2CT1(LXJ,M,RXNL,INXXJ,LAI,M,RANL,INXAI,
83 - 93.000      1      LAK,MAK,RANL,INXAK,LAK,MAK,RANL,INXAK,
84 - 94.000      2      RAMESH,IAPTS,.FALSE.)
85 - 95.000      IF (LLDTAL) WRITE(6,894) NXJ,LXJ,M,NAI,LAI,M,NAK,LAK,MAK,VCCT1
86 - 96.000      894 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,
87 - 97.000      1      ' NAI=',I2,' LAI=',I2,' M=',I2,
88 - 98.000      2      ' NAK=',I2,' LAK=',I2,' MAK=',I2,
89 - 99.000      3      ' VCCT1=',IPE14.6)
90 - 100.000     VECT1=EL2CT1(LXJ,M,RXNL,INXXJ,LAK,MAK,RANL,INXAK,
91 - 101.000     1      LAK,MAK,RANL,INXAK,LAI,M,RANL,INXAI,
92 - 102.000     2      RAMESH,IAPTS,.FALSE.)
93 - 103.000     IF (LLDTAL) WRITE(6,896) NXJ,LXJ,M,NAK,LAK,MAK,NAI,LAI,M,VECT1
94 - 104.000     896 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,
95 - 105.000     1      ' NAK=',I2,' LAK=',I2,' MAK=',I2,
96 - 106.000     2      ' NAI=',I2,' LAI=',I2,' M=',I2,
97 - 107.000     3      ' VECT1=',IPE14.6)
98 - 108.000     2 F=F+(4.0*VCCT1-2.0*VECT1)*FRA(LAK1,NAK-LAK)
99 - 109.000     3 CONTINUE
100 - 109.005     WRITE(6,199) NXJ,LXJ,M,NAI,LAI,M,F
101 - 109.010     199 FORMAT('0','CLOSED SHELL VERSION GIVES','0','NXJ=',I2,
102 - 109.015     1 ' LXJ=',I2,' M=',I2,' NAI=',I2,' LAI=',I2,' M=',I2,' FAV=',
103 - 109.020     2 IPE14.6)
104 - 109.025     F=FSAVE
105 - 109.030     104 KOUNG=1
106 - 109.035     KOUNF=1
107 - 109.040     DO 103 LAK1=1,LAMAX1
108 - 109.045     LCCAK=LCCAFN(LAK1)
109 - 109.050     IF (LCCAK.EQ.0) GO TO 103
110 - 109.055     LAK=LAK1+1
111 - 109.060     NAKMAX=LCCAK+LAK
112 - 109.061     DO 102 NAK=LAK1,NAKMAX
113 - 109.065     INXAK=IAPTS*(LARINX(LAK1)+NAK-LAK1)
114 - 109.070     GATEM2=GA(LAI1,KOUNG,2)
115 - 109.075     CALL FK(RAMESH,RANL,0,INXAK,INXAI,INXXJ,ANS)
116 - 109.080     F=F+XAF(LAI1,KOUNF)*ANS
117 - 109.085     KOUNF=KOUNF+1
118 - 109.090     100 GATEM1=GA(LAI1,KOUNG,1)

```

```

119 - 109.095      CALL GK(RAMESH,RANL,GATEM1,INXAK,INXAI,INXXJ,ANS)
120 - 109.100      F=F+XAG(LAI1,KOUNG)*ANS
121 - 109.105      KOUNG=KOUNG+1
122 - 109.110      IF(KOUNG.GT.NAG(LAI1)) GO TO 102
123 - 109.115      IF(GATEM2.EQ.GA(LAI1,KOUNG,2)) GO TO 100
124 - 109.120      102 CONTINUE
125 - 109.125      103 CONTINUE
126 - 110.000      IF(LLDTAL.OR.LLCNFG) WRITE(6,898) NXJ,LXJ,M,NAI,LAI,M,F
127 - 111.000      898 FORMAT('0','NXJ =',I2,' LXJ =',I2,' M =',I2,
128 - 112.000      1      ' NAI =',I2,' LAI =',I2,' M =',I2,
129 - 113.000      2      ' FAU =',IPE14.6)
130 - 114.000      4 CALL I0(LLGENI,IIFILE,OC1,.FALSE.,LLBUF)
131 - 115.000      CALL I0(LLGENI,IIFILE,F,.FALSE.,LLBUF)
132 - 116.000      S=CMPLX(OC1,0.0)*ZSUM
133 - 117.000      H=CMPLX(F,0.0)*ZSUM
134 - 118.000      IF (LLDTAL) WRITE(6,899) M,S
135 - 119.000      899 FORMAT(' ',M,SCUM =',IPE14.7,' SCUM =',IPE14.7)
136 - 120.000      6 IXJDNL=LXDINX(LXJ1)+LXJ1*(NXJ-LXJ1)
137 - 121.000      IAIDNL=LADINX(LAI1)+LAI1*(NAI-LAI1)
138 - 122.000      INDEX=0
139 - 123.000      DO 17 MUSET=1,NSETS
140 - 124.000      IDISP=(MUSET-1)*LXDINX(LXMAX1+1)
141 - 125.000      IXJDU=IXJDNL+IDISP
142 - 126.000      IAIDU=IAIDNL+(MUSET-1)*LADINX(LAMAX1+1)
143 - 127.000      NAXULM=NAXSET(MUSET)
144 - 128.000      DO 16 M1=1,MMAX1
145 - 129.000      M=M1-1
146 - 130.000      MUINX=INDEX
147 - 131.000      ZKCRM=(0.0,0.0)
148 - 132.000      DO 607 MU=1,NAXULM
149 - 133.000      RUBX=RUB(MUINX+1)
150 - 134.000      RUBY=RUB(MUINX+2)
151 - 135.000      RUBZ=RUB(MUINX+3)
152 - 136.000      MUINX=MUINX+3
153 - 137.000      IF (.NOT.LLGENI) GO TO 603
154 - 138.000      IF (MUSET.LE.NSETSC) GO TO 601
155 - 139.000      ZCRM=(0.0,0.0)
156 - 140.000      GO TO 603
157 - 141.000      601 ZCRM=ZCRM+RUBX,RUBY,RUBZ,JX,LXJ,CX,MBLCHX,IA,LAI,CA,MBLGHA)
158 - 142.000      603 CALL I0(LLGENI,IIFILE,REZCRM,.FALSE.,LLBUF)
159 - 143.000      CALL I0(LLGENI,IIFILE,IMZCRM,.FALSE.,LLBUF)
160 - 144.000      IF (MUSET.GT.NSETSC) GO TO 607
161 - 145.000      ZKCRM=ZKCRM+CEXP(CMPLX(0.0,0.0-TWOPI*UNIT*(XKX+RUBX+XKY+RUBY
162 - 146.000      1      +XKZ+RUBZ)))*ZCRM
163 - 147.000      607 CONTINUE
164 - 148.000      IF (LLDTAL) WRITE(6,900) MUSET,JX,LXJ,IA,LAI,M,ZKCRM,NAXULM
165 - 149.000      900 FORMAT('0','MUSET =',I2,' JX =',I2,' LXJ =',I2,' IA =',I2,
166 - 150.000      1      ' LAI =',I2,' M =',I2,' ZKCRM =',IPE14.6,
167 - 151.000      2      ' NAXULM =',I3)
168 - 152.000      IF (.NOT.LLGENI) GO TO 11
169 - 153.000      SUSETH=0.0
170 - 154.000      HUSETH=0.0
171 - 155.000      IF (CABS(ZKCRM).GE.EPSLON) GO TO 608
172 - 156.000      ZKCRM=(0.0,0.0)
173 - 158.000      608 IXJDM=IXJDU+M1
174 - 159.000      IAIDM=IAIDU+M1
175 - 160.000      CCCCC *****
176 - 161.000      IXJDM=IALPTS*(IXJDM-1)+1
177 - 162.000      READ DISK IXAFLE,IXJDM,ALPHA
178 - 163.000      IXJDM=(IXJDM-1)/IALPTS+1

```

```

179 - 164.000 CCCCC *****
180 - 165.000 CCCCC *****
181 - 166.000 C      READ (IXAFLE,IXJDM) ALPHA
182 - 167.000 CCCCC *****
183 - 168.000      SUSETM=OVRLAP(LXJ,M,ALPHA,MXPPTS,MXPAN,NXPAN,LAI,M,RANL,INXAI,
184 - 169.000      1      RAMESH,IAPTS)
185 - 170.000      IF (LLDTAL) WRITE(6,901) NXJ,LXJ,M,NAI,LAI,M,SUSETM
186 - 171.000      901 FORMAT('0','NXJ=',I2,' LXJ=',I2,' M=',I2,' NAI=',I2,
187 - 172.000      1      ' LAI=',I2,' M=',I2,' O=',I2,'PE14.6/')
188 - 173.000      EL1A=OVRLAP(LXJ,M,ALPHA,MXPPTS,MXPAN,NXPAN,LAI,M,DRANL,INXAI,
189 - 174.000      1      RAMESH,IAPTS)
190 - 175.000      IF (LLDTAL) WRITE(6,90012) NXJ,LXJ,M,NAI,LAI,M,EL1A
191 - 176.000      90012 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,' NAI=',I2,
192 - 177.000      1      ' LAI=',I2,' M=',I2,' EL1A=',I2,'PE14.6/')
193 - 178.000      HTEMP=0.5*EL1A
194 - 179.000      DO 7003 LAK1=1,LAMAX1
195 - 180.000      LCCAK=LCCAFN(LAK1)
196 - 181.000      IF (LCCAK.EQ.0) GO TO 7003
197 - 182.000      LAK=LAK1-1
198 - 183.000      NAKMAX=LCCAK+LAK
199 - 184.000      MAKTOT=2*LAK+1
200 - 185.000      DO 7002 NAK=LAK1,NAKMAX
201 - 186.000      INXAK=IAPTS*(LARINX(LAK1)+NAK=LAK1)
202 - 187.000      DO 7002 MMAK=1,MAKTOT
203 - 188.000      MAK=MMAK-LAK1
204 - 189.000      HYVC=HYBAAA(LXJ,M,ALPHA,NXPAN,LAI,M,RANL,INXAI,
205 - 190.000      1      LAK,MAK,RANL,INXAK,LAK,MAK,RANL,INXAK,
206 - 191.000      2      RAMESH,IAPTS,MXPPTS,MXPAN, .FALSE.)
207 - 192.000      IF (LLDTAL) WRITE(6,90014) NXJ,LXJ,M,NAI,LAI,M,NAK,LAK,MAK,MYVC
208 - 193.000      90014 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,
209 - 194.000      1      ' NAI=',I2,' LAI=',I2,' M=',I2,
210 - 195.000      2      ' NAK=',I2,' LAK=',I2,' MAK=',I2,
211 - 196.000      3      ' HYVC=',I2,'PE14.6)
212 - 197.000      HYVE=HYBAAA(LXJ,M,ALPHA,NXPAN,LAK,MAK,RANL,INXAK,
213 - 198.000      1      LAK,MAK,RANL,INXAK,LAI,M,RANL,INXAI,
214 - 199.000      2      RAMESH,IAPTS,MXPPTS,MXPAN, .FALSE.)
215 - 200.000      IF (LLDTAL) WRITE(6,90015) NXJ,LXJ,M,NAK,LAK,MAK,NAI,LAI,M,MYVE
216 - 201.000      90015 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,
217 - 202.000      1      ' NAK=',I2,' LAK=',I2,' MAK=',I2,
218 - 203.000      2      ' NAI=',I2,' LAI=',I2,' M=',I2,
219 - 204.000      3      ' HYVE=',I2,'PE14.6)
220 - 205.000      HTEMP=HTEMP+2.0*HYVC*FRA(LAK1,NAK=LAK)
221 - 205.500      HTEMP=HTEMP+2.0*HYVE*FRASP(LAK1,NAK=LAK)
222 - 206.000      7002 CONTINUE
223 - 207.000      7003 CONTINUE
224 - 208.000      HUSETM=HUSETM+HTEMP
225 - 209.000      IF (LLDTAL) WRITE(6,90016)
226 - 210.000      90016 FORMAT('0')
227 - 211.000      8 IF (LLANOC.OR.LLXNOC) GO TO 9
228 - 212.000      C=COUL(LXJ,M,ALPHA,NXPAN,LXJ,M,ALPHA,NXPAN,
229 - 213.000      1      LAI,M,RANL,INXAI,LAI,M,RANL,INXAI,
230 - 214.000      2      RAMESH,IAPTS,MXPPTS,MXPAN,MXPPTS,MXPAN, .FALSE.)
231 - 215.000      IF (LLDTAL) WRITE(6,904) NXJ,LXJ,M,NAI,LAI,M,C
232 - 216.000      904 FORMAT(' ','NXJ=',I2,' LXJ=',I2,' M=',I2,' NAI=',I2,
233 - 217.000      1      ' LAI=',I2,' M=',I2,' C=',I2,'PE14.6/')
234 - 218.000      HUSETM=HUSETM+2.0*C*SUSETM*SQRT(FRXSP(LXJ1,NXJ=LXJ1)*FRASP(
235 - 218.005      1LAI1,NAI=LAI))
236 - 219.000      9 CONTINUE
237 - 220.000      IF (LLSAME.AND.IXJDM.EQ.IAIDM) GO TO 9009
238 - 221.000 CCCCC *****

```

```

239 = 222.000      IAIDM=IALPTS*(IAIDM=1)+1
240 = 223.000      READ DISK IAXFLE,IAIDM,ALPHA
241 = 224.000      IAIDM=(IAIDM=1)/IALPTS+1
242 = 225.000 CCCCC *****
243 = 226.000 CCCCC *****
244 = 227.000 C      READ(IAXFLE,IAIDM) ALPHA
245 = 228.000 CCCCC *****
246 = 229.000      EL1X=0VRLAP(LAI,M,ALPHA,MXPPTS,MXPAN,NXPAN,LXJ,M,DRXNL,INXXJ,
247 = 230.000      1      RXMESH,IXPTS)*(-1.0)**(LAI+LXJ)
248 = 231.000      IF (LLDTAL) WRITE(6,90043) NXJ,LXJ,M,NAI,LAI,M,EL1X
249 = 232.000 90043 FORMAT(' ',NXJ='I2', LXJ='I2', M='I2', NAI='I2',
250 = 233.000      1      ' LAI='I2', M='I2', EL1X='1PE14.6)
251 = 234.000      HTEMP=0.5*EL1X
252 = 235.000      DO 9008 LKK1=1,LXMAX1
253 = 236.000      LCCCK=LCCXFN(LKK1)
254 = 237.000      IF (LCCCK.EQ.0) GO TO 9008
255 = 238.000      LXX=LKK1-1
256 = 239.000      NXKMAX=LCCCK+LXX
257 = 240.000      MXTOT=2*LXX+1
258 = 241.000      DO 9007 NXK=LKK1,NXKMAX
259 = 242.000      INXX=IXPTS+(LXRINX(LKK1)+NXK=LKK1)
260 = 243.000      DO 9007 MMXK=1,MXTOT
261 = 244.000      MXK=MMXK+LXX
262 = 245.000      HYVC=HYBAAA(LAI,M,ALPHA,NXPAN,LXJ,M,RXNL,INXXJ,
263 = 246.000      1      LXX,MXK,RXNL,INXX,LXX,MXK,RXNL,INXX)
264 = 247.000      2      RXMESH,IXPTS,MXPPTS,MXPAN, .FALSE.)*(-1.0)**(LAI+LXJ)
265 = 248.000      IF (LLDTAL) WRITE(6,90046) NXJ,LXJ,M,NAI,LAI,M,NXK,LXX,MXK,MYVC
266 = 249.000 90046 FORMAT(' ',NXJ='I2', LXJ='I2', M='I2',
267 = 250.000      1      ' NAI='I2', LAI='I2', M='I2',
268 = 251.000      2      ' NXK='I2', LXX='I2', MXK='I2',
269 = 252.000      3      ' HYVC='1PE14.6)
270 = 253.000      HYVE=HYBAAA(LAI,M,ALPHA,NXPAN,LXX,MXK,RXNL,INXX,
271 = 254.000      1      LXX,MXK,RXNL,INXX,LXJ,M,RXNL,INXXJ,
272 = 255.000      3      RXMESH,IXPTS,MXPPTS,MXPAN, .FALSE.)*(-1.0)**(LAI+LXJ)
273 = 256.000      IF (LLDTAL) WRITE(6,90049) NXJ,LXJ,M,NXK,LXX,MXK,NAI,LAI,M,MYVE
274 = 257.000 90049 FORMAT(' ',NXJ='I2', LXJ='I2', M='I2',
275 = 258.000      1      ' NXK='I2', LXX='I2', MXK='I2',
276 = 259.000      2      ' NAI='I2', LAI='I2', M='I2',
277 = 260.000      3      ' HYVE='1PE14.6)
278 = 261.000      HTEMP=HTEMP+2.0*HYVC*FRX(LKK1,NXK=LXX)
279 = 261.500      HTEMP=HTEMP-2.0*HYVE*FRXSP(LKK1,NXK=LXX)
280 = 262.000      9007 CONTINUE
281 = 263.000      9008 CONTINUE
282 = 264.000      9009 HUSETM=HUSETM+HTEMP
283 = 265.000      IF (LLDTAL) WRITE(6,90016)
284 = 266.000      11 CALL I0(LLGENI,IIFILE,SUSETM, .FALSE.,LLBUF)
285 = 267.000      CALL I0(LLGENI,IIFILE,HUSETM, .FALSE.,LLBUF)
286 = 268.000      IF (MUSET.GT.NSETSC) GO TO 16
287 = 269.000      S=S+CMPLX(SUSETM,0.0)*ZKCRM
288 = 270.000      H=H+CMPLX(HUSETM,0.0)*ZKCRM
289 = 271.000      IF (LLDTAL) WRITE(6,899) H,S
290 = 272.000      16 CONTINUE
291 = 273.000      17 INDEX=INDEX+3*NAXULM
292 = 274.000      RETURN
293 = 275.000      END
294 = 275.005      SUBROUTINE FK(R,PSI,K,INXL,INXM,INXN,ANS)
295 = 275.010      DIMENSION R(161),PSI(1)
296 = 275.015      COMMON /SKRAP/S(161),SN(161)
297 = 275.020 CCCCC COMPUTES FK RADIAL INTEGRALS
298 = 275.025      ANS=0.0

```

```

299 - 275.030      DO 1 I=1,161
300 - 275.035      S(I)=0.0
301 - 275.040      1 SN(I)=0.0
302 - 275.045      DO 3 I=3,161,2
303 - 275.050      INXLI=INXL+I
304 - 275.055      3 SN(I)=SN(I-2)+((R(I)-R(I-1))/3.)*(PSI(INXLI)**2/
305 - 275.060      1R(I)**(K-1)+4.*PSI(INXLI-1)**2/R(I-1)**(K-1)
306 - 275.065      2+PSI(INXLI-2)**2/R(I-2)**(K-1))
307 - 275.070      DO 4 I=3,161,2
308 - 275.075      INXLI=INXL+I
309 - 275.080      4 S(I)=S(I-2)+((R(I)-R(I-1))/3.)*(PSI(INXLI)**2*
310 - 275.085      1R(I)**(K+2)+4.*PSI(INXLI-1)**2*R(I-1)**(K+2)
311 - 275.090      2+PSI(INXLI-2)**2*R(I-2)**(K+2))
312 - 275.095      DO 5 I=1,161,2
313 - 275.100      INXMI=INXM+I
314 - 275.105      INXNI=INXN+I
315 - 275.110      5 S(I)=PSI(INXMI)*PSI(INXNI)*(S(I)/R(I))**K-1)
316 - 275.115      1+R(I)**(K+2)*(SN(161)=SN(I))
317 - 275.120      S(2)=0.5*(S(1)+S(3))
318 - 275.125      S(160)=0.5*(S(159)+S(161))
319 - 275.130      CALL INTP(S,R)
320 - 275.135      DO 6 I=1,159,2
321 - 275.140      6 ANS=ANS+((R(I+1)-R(I))/3.)*(S(I)+4.*S(I+1)+S(I+2))
322 - 275.145      ANS=2.0*ANS
323 - 275.150      RETURN
324 - 275.155      END
325 - 275.160      SUBROUTINE GK(R,PSI,K,INXL,INXM,INXN,ANS)
326 - 275.165      DIMENSION R(161),PSI(1)
327 - 275.170      COMMON /SKRAP/S(161),SN(161)
328 - 275.175      CCCCC COMPUTES GK RADIAL INTEGRALS
329 - 275.180      ANS=0.0
330 - 275.185      DO 1 I=1,161
331 - 275.190      S(I)=0.0
332 - 275.195      1 SN(I)=0.0
333 - 275.200      DO 3 I=3,161,2
334 - 275.205      INXLI=INXL+I
335 - 275.210      INXMI=INXM+I
336 - 275.215      3 SN(I)=SN(I-2)+((R(I)-R(I-1))/3.)*(PSI(INXLI)*PSI(INXMI)/
337 - 275.220      1R(I)**(K-1)+PSI(INXLI-1)**4.*PSI(INXMI-1)/R(I-1)**(K-1)
338 - 275.225      2+PSI(INXLI-2)*PSI(INXMI-2)/R(I-2)**(K-1))
339 - 275.230      DO 4 I=3,161,2
340 - 275.235      INXLI=INXL+I
341 - 275.240      INXMI=INXM+I
342 - 275.245      4 S(I)=S(I-2)+((R(I)-R(I-1))/3.)*(PSI(INXLI)*PSI(INXMI)*
343 - 275.250      1R(I)**(K+2)+4.*PSI(INXLI-1)*PSI(INXMI-1)*R(I-1)**(K+2)
344 - 275.255      2+PSI(INXLI-2)*PSI(INXMI-2)*R(I-2)**(K+2))
345 - 275.260      DO 5 I=1,161,2
346 - 275.265      INXLI=INXL+I
347 - 275.270      INXNI=INXN+I
348 - 275.275      5 S(I)=PSI(INXNI)*PSI(INXLI)*(S(I)/R(I))**K-1)
349 - 275.280      1+R(I)**(K+2)*(SN(161)=SN(I))
350 - 275.285      S(2)=0.5*(S(1)+S(3))
351 - 275.290      S(160)=0.5*(S(159)+S(161))
352 - 275.295      CALL INTP(S,R)
353 - 275.300      DO 6 I=1,159,2
354 - 275.305      6 ANS=ANS+((R(I+1)-R(I))/3.)*(S(I)+4.*S(I+1)+S(I+2))
355 - 275.310      ANS=2.0*ANS
356 - 275.315      RETURN
357 - 275.320      END
358 - 275.325      SUBROUTINE INTP(S,R)

```

```

359 - 275.330 DIMENSION S(161),R(161)
360 - 275.335 DO 1 I=4,158,2
361 - 275.340 A=(R(I)-R(I-1))*(R(I)-R(I+1))*(R(I)-R(I+3))
362 - 275.345 B=(R(I)-R(I-3))*(R(I)-R(I+1))*(R(I)-R(I+3))
363 - 275.350 C=(R(I)-R(I-3))*(R(I)-R(I-1))*(R(I)-R(I+3))
364 - 275.355 D=(R(I)-R(I-3))*(R(I)-R(I-1))*(R(I)-R(I+1))
365 - 275.360 E=(R(I-3)-R(I-1))*(R(I-3)-R(I+1))*(R(I-3)-R(I+3))
366 - 275.365 F=(R(I-1)-R(I-3))*(R(I-1)-R(I+1))*(R(I-1)-R(I+3))
367 - 275.370 G=(R(I+1)-R(I-3))*(R(I+1)-R(I-1))*(R(I+1)-R(I+3))
368 - 275.375 H=(R(I+3)-R(I-3))*(R(I+3)-R(I-1))*(R(I+3)-R(I+1))
369 - 275.380 1 S(I)=S(I-3)*A/E+S(I-1)*B/F+S(I+1)*C/G+S(I+3)*D/H
370 - 275.385 RETURN
371 - 275.390 END
372 - 276.000 SUBROUTINE I0(LLGENI,IIFILE,DATUM,LEND,LLBUF)
373 - 277.000 DIMENSION BUFF( 512)
374 - 278.000 LOGICAL LLGENI,LEND,LLBUF
375 - 279.000 DATA INX/1/,INXLIM/ 512/
376 - 280.000 IF (LLGENI) GO TO 5
377 - 281.000 IF (.NOT.LEND) GO TO 1
378 - 282.000 REWIND IIFILE
379 - 283.000 INX=1
380 - 284.000 RETURN
381 - 285.000 1 IF (INX.NE.1) GO TO 3
382 - 286.000 2 CALL RW(13,IIFILE,BUFF,INXLIM,.FALSE.,LLBUF)
383 - 287.000 INX=1
384 - 288.000 GO TO 4
385 - 289.000 3 IF (INX.GT.INXLIM) GO TO 2
386 - 290.000 4 DATUM=BUFF(INX)
387 - 291.000 INX=INX+1
388 - 292.000 RETURN
389 - 293.000 5 IF (.NOT.LEND) GO TO 8
390 - 294.000 IF (INX.GT.INXLIM) GO TO 7
391 - 295.000 DO 6 I=INX,INXLIM
392 - 296.000 6 BUFF(I)=0.0
393 - 297.000 7 CALL RW(14,IIFILE,BUFF,INXLIM,.TRUE.,LLBUF)
394 - 298.000 REWIND IIFILE
395 - 299.000 INX=1
396 - 300.000 RETURN
397 - 301.000 8 IF (INX.LE.INXLIM) GO TO 9
398 - 302.000 CALL RW(15,IIFILE,BUFF,INXLIM,.TRUE.,LLBUF)
399 - 303.000 INX=1
400 - 304.000 9 BUFF(INX)=DATUM
401 - 305.000 INX=INX+1
402 - 306.000 RETURN
403 - 307.000 END
404 - 308.000 C!ASSIGN M:80,(FILE,MINTGD),(SAVE)
405 - 309.000 C!FORTRAN LS,80,ADP
406 - 310.000 FUNCTION EL1CT1(RNL1,INX1,V,RNL2,INX2,RMESH,IPTS,LLV)
407 - 311.000 DIMENSION RNL1(1),V(1),RNL2(1),RMESH(1)
408 - 312.000 LOGICAL LLV
409 - 313.000 EL1CT1=0.0
410 - 314.000 DO 3 K=3,IPTS,2
411 - 315.000 INX1K=INX1+K
412 - 316.000 INX2K=INX2+K
413 - 317.000 IF (K.NE.3) GO TO 1
414 - 318.000 R1=RMESH(1)
415 - 319.000 F1=RNL1(INX1+1)*RNL2(INX2+1)*R1*R1
416 - 320.000 IF (LLV) F1=F1*V(1)
417 - 321.000 GO TO 2
418 - 322.000 1 R1=R3

```

```

419 = 323.000      F1=F3
420 = 324.000      2 F2=RNL1(INX1K-1)*RNL2(INX2K-1)*RMESH(K-1)**2
421 = 325.000      R3=RMESH(K)
422 = 326.000      F3=RNL1(INX1K)*RNL2(INX2K)*R3*R3
423 = 327.000      IF (.NOT.LLV) GO TO 3
424 = 328.000      F2=F2+V(K-1)
425 = 329.000      F3=F3+V(K)
426 = 330.000      3 ELICT1=ELICT1+(R3=R1)/6.0*(F1+4.0*F2+F3)
427 = 331.000      RETURN
428 = 332.000      END
429 = 333.000      FUNCTION BVRLAP(LXJ,MXJ,ALPHA,MXPTS,MXPAN,NXPAN,
430 = 334.000      1      LAI,MAI,RNL,INXI,RMESH,IPTS)
431 = 335.000      DIMENSION ALPHA(MXPTS,MXPAN),RNL(1,1,1,1,1,1,1,1)
432 = 336.000      XNORM2(L,M)=FBAT(2*L+1)*FACT(L-1)*ABS(M)/(2.0*FACT(L+1)*ABS(M))
433 = 337.000      BVRLAP=0.0
434 = 338.000      LXJ1=LXJ+1
435 = 339.000      LAI1=LAI+1
436 = 340.000      IF (MXJ.NE.MAI) RETURN
437 = 341.000      IF (NXPAN.GE.LAI1) GO TO 1
438 = 342.000      WRITE(6,900) LAI1,NXPAN
439 = 343.000      900 FORMAT(' ', 'INSUFFICIENT ALPHA FUNCTIONS TO DETERMINE OVERLAP')
440 = 344.000      1      LAI1 = ',I3,'      NXPAN = ',I3)
441 = 345.000      CALL EXIT
442 = 346.000      1 DO 4 K=3,IPTS,2
443 = 347.000      INXIK=INXI+K
444 = 348.000      IF (K.NE.3) GO TO 2
445 = 349.000      R1=RMESH(1)
446 = 350.000      F1=ALPHA(1,LAI1)*RNL(INXI+1)*R1**2
447 = 351.000      GO TO 3
448 = 352.000      2 R1=R3
449 = 353.000      F1=F3
450 = 354.000      3 F2=ALPHA(K,LAI1)*RNL(INXIK-1)*RMESH(K-1)**2
451 = 355.000      R3=RMESH(K)
452 = 356.000      F3=ALPHA(K,LAI1)*RNL(INXIK)*R3**2
453 = 357.000      4 BVRLAP=BVRLAP+(R3=R1)/6.0*(F1+4.0*F2+F3)
454 = 358.000      BVRLAP=SQRT(XNORM2(LXJ,MXJ)/XNORM2(LAI,MAI))*BVRLAP
455 = 359.000      RETURN
456 = 360.000      END
457 = 361.000      FUNCTION ZCRBTM(M,RUBX,RUBY,RUBZ,JX,LXJ,CX,MXBLHX,
458 = 362.000      1      IA,LA,CA,MXBLHA)
459 = 363.000      COMPLEX ZCRBTM,CX(MXBLHX,1),CA(MXBLHA,1),ZCCXJ,ZCAI
460 = 364.000      DATA EPSLN/1.0E-05/
461 = 365.000      ZCRBTM=(0.0,0.0)
462 = 366.000      LXJ1=LXJ+1
463 = 367.000      LAI1=LAI+1
464 = 368.000      MXTBT=2*LXJ+1
465 = 369.000      MATBT=2*LA+1
466 = 370.000      IF (RUBX.NE.0.0.OR.RUBY.NE.0.0) GO TO 2
467 = 371.000      ALPHA=0.0
468 = 372.000      CBETA=1.0
469 = 373.000      IF (RUBZ.EQ.0.0) GO TO 3
470 = 374.000      CBETA=SIGN(1.0,RUBZ)
471 = 375.000      GO TO 3
472 = 376.000      2 ALPHA=ATAN2(RUBY,RUBX)
473 = 377.000      CBETA=RUBZ/SQRT(RUBX**2+RUBY**2+RUBZ**2)
474 = 378.000      3 DO 5 J=1,MXTBT
475 = 379.000      MXJ=J-LXJ1
476 = 380.000      INXMXJ=2*1ABS(MXJ)
477 = 381.000      IF (MXJ.LE.0) INXMXJ=INXMXJ+1
478 = 382.000      ZCCXJ=CONJG(CX(JX,INXMXJ))

```

```

479 - 383.000      IF (CABS(ZCCXJ).EQ.0.0) GO TO 5
480 - 384.000      ROTXJ=ROTCOE(M,LXJ,MXJ,CBETA)
481 - 385.000      ROTXJM=0.0
482 - 386.000      IF (M.NE.0) ROTXJM=ROTCOE(O=M,LXJ,MXJ,CBETA)
483 - 387.000      IF (ABS(ROTXJ).LT.EPSLON.AND.ABS(ROTXJM).LT.EPSLON) GO TO 6
484 - 388.000      DO 4 I=1,MATOT
485 - 389.000      MAI=I-LAI1
486 - 390.000      INXMAI=2*IABS(MAI)
487 - 391.000      IF (MAI.LE.0) INXMAI=INXMAI+1
488 - 392.000      ZCAI=CA(I,INXMAI)
489 - 393.000      IF (CABS(ZCAI).EQ.0.0) GO TO 4
490 - 394.000      ROTAI=ROTCOE(M,LAJ,MAI,CBETA)
491 - 395.000      ROTAIM=0.0
492 - 396.000      IF (M.NE.0) ROTAIM=ROTCOE(O=M,LAJ,MAI,CBETA)
493 - 397.000      IF (ABS(ROTAI).LT.EPSLON.AND.ABS(ROTAIM).LT.EPSLON) GO TO 4
494 - 398.000      RR=ROTXJ*ROTAI
495 - 399.000      IF (M.NE.0) RR=RR+ROTXJM*ROTAIM
496 - 400.000      ZCROTH=ZCROTH+ZCCXJ*ZCAI*CMPLX(RR,0.0)
497 - 401.000      1      *CEXP(CMPLX(0.0,FLBAY(MAI-MXJ)*ALPHA))
498 - 402.000      4 CONTINUE
499 - 403.000      5 CONTINUE
500 - 404.000      RETURN
501 - 405.000      END
502 - 406.000      FUNCTION ROTCOE(K,L,M,T)
503 - 407.000 C      ROTATIONAL TRANSFORMATION COEFFICIENTS OF SPHERICAL HARMONICS
504 - 408.000      ROTCOE=1.0
505 - 409.000      IF (K.NE.0.OR.L.NE.0.OR.M.NE.0) GO TO 6
506 - 410.000      RETURN
507 - 411.000      6 TT=T
508 - 412.000      SIGN=1.0
509 - 413.000      IM=M
510 - 414.000      IK=K
511 - 415.000      IF (IM) 12,14,15
512 - 416.000      12 IM=0-IM
513 - 417.000      IK=0-IK
514 - 418.000      ROTCOE=ROTCOE*(-1.0)**(IM=IK)
515 - 419.000      IF (IM.GE.IK) GO TO 4
516 - 420.000      13 IKEEP=IM
517 - 421.000      IM=IK
518 - 422.000      IK=IKEEP
519 - 423.000      SIGN=0.0-SIGN
520 - 424.000      GO TO 4
521 - 425.000      14 IF (IM.GE.IK) GO TO 4
522 - 426.000      IM=0-IM
523 - 427.000      IK=0-IK
524 - 428.000      ROTCOE=ROTCOE*(-1.0)**(IM=IK)
525 - 429.000      GO TO 4
526 - 430.000      15 IF (IM.LT.IK) GO TO 13
527 - 431.000      4 IF (TT.GT.(-1.0)) GO TO 10
528 - 432.000      IF (IM.NE.(0=IK)) GO TO 8
529 - 433.000      ROTCOE=ROTCOE*(-1.0)**(L=M)
530 - 434.000      RETURN
531 - 435.000      8 ROTCOE=0.0
532 - 436.000      RETURN
533 - 437.000      10 IF (TT.LT.1.0) GO TO 11
534 - 438.000      IF (IM.NE.IK) GO TO 8
535 - 439.000      RETURN
536 - 440.000      11 FLMIK=FACT(L=IK)
537 - 441.000      FLPIK=FACT(L=IK)
538 - 442.000      TT1=(1.0-TT)/(1.0+TT)

```

```

539 - 443.000      MIN=0-IK
540 - 444.000      MAX=MIN0(L=IM=IK,L)
541 - 445.000      LIM=MAX-MIN+1
542 - 446.000      SUM=0.0
543 - 447.000      DO 5 I=1,LIM
544 - 447.010      J=MIN+I-1
545 - 447.020      5 SUM=SUM+(=1.0)**(IM+J)*FLMIK/(FACT(IM+J)*FACT(L-IM+K-J))
546 - 447.030      1      *FLPIK/(FACT(L-J)*FACT(IK+J+1+TT1+J)
547 - 447.040      ROTCOE=ROTCOE*SUM*SIGN**(IM+IK)
548 - 447.050      1      *SQRT(FACT(L+IM)/FLPK*FACT(L-IM)/FLM+TT1**(IM+IK))
549 - 447.060      2      *((1.0+TT)/2.0)**L
550 - 447.070      RETURN
551 - 447.080      END
552 - 447.090      FUNCTION EL2CT1(L1,M1,RNL1,INX1,L2,M2,RNL2,INX2,
553 - 447.100      1      L3,M3,RNL3,INX3,L4,M4,RNL4,INX4,
554 - 447.110      2      RMESH,IPTS,DETAIL)
555 - 447.120 CCCCC FUNCTION EL2CT1 CALCULATES TWO-ELECTRON ONE-CENTER INTEGRAL
556 - 447.130 CCCCC      GIVEN THE ANGULAR QUANTUM NUMBERS AND THE RADIAL FUNCTIONS
557 - 447.140 CCCCC WRITTEN BY D. J. MICKISH SEPTEMBER 1972
558 - 447.150      DIMENSION RNL1(1),RNL2(1),RNL3(1),RNL4(1),RMESH(1)
559 - 447.160      COMMON /TWDEL/P3P*(161),SS(161)
560 - 447.170 CCCCC      DETAIL IS USED TO CAUSE PRINTING OF TERMS OF RESULTING SUM
561 - 447.180      LOGICAL DETAIL
562 - 448.000      IPTSLM=161
563 - 449.000      EL2CT1=0.0
564 - 450.000 CCCCC      POLAR ANGLE CONDITION
565 - 451.000      M=M2-M1
566 - 452.000      IF (M.EQ.M3-M4) GO TO 97
567 - 453.000      WRITE(6,96) M1,M2,M3,M4
568 - 454.000      96 FORMAT(' ',M1=' ',I2,' ',M2=' ',I2,' ',M3=' ',I2,' ',M4=' ',I2)
569 - 455.000      RETURN
570 - 456.000      97 L1L2=L1+L2
571 - 457.000      L3L4=L3+L4
572 - 458.000      L1ML2=IABS(L1-L2)
573 - 459.000      L3ML4=IABS(L3-L4)
574 - 460.000      IWORK=L1L2+L3L4
575 - 461.000      IF (IWORK.NE.2*(IWORK/2)) RETURN
576 - 462.000      IF (IPTS.LE.IPTSLM) GO TO 99
577 - 463.000      WRITE(6,98) IPTS
578 - 464.000      98 FORMAT(' ',NUMBER OF INTEGRATION POINTS =' ',I4,' EXCEEDS LIMIT ',
579 - 465.000      1      'IN FUNCTION SUBPROGRAM EL2CT1')
580 - 466.000      RETURN
581 - 467.000      99 PI=3.141592653589793
582 - 468.000      DO 100 J=1,IPTS
583 - 469.000      100 P3P*(J)=RNL3(INX3+J)*RNL4(INX4+J)*RMESH(J)**2
584 - 470.000      KMIN1=MAX0(L1ML2,L3ML4)+1
585 - 471.000      IM=IABS(M)
586 - 472.000      IF (KMIN1.GE.IM+1) GO TO 5
587 - 473.000      KMIN1=IM+1
588 - 474.000 CCCCC      IF KMIN1 IS INFLUENCED BY M, THEN CORRECT PARITY MUST BE
589 - 475.000 CCCCC      CHECKED.
590 - 476.000      IWORK=IM+L1L2
591 - 477.000      IF (IWORK.NE.2*(IWORK/2)) KMIN1=KMIN1+1
592 - 478.000      5 KMAX1=MIN0(L1L2,L3L4)+1
593 - 479.000      IF (KMIN1.GT.KMAX1) GO TO 21
594 - 480.000      DO 20 K1=KMIN1,KMAX1,2
595 - 481.000      K=K1-1
596 - 482.000      CALL IR12K(P3P*,K,SS,RMESH,IPTS,0.0)
597 - 483.000      DO 1402 J=1,IPTS
598 - 484.000      1402 SS(J)=SS(J)*RMESH(J)**2

```

```

599 - 485.000      C2=YLMINT(K,L3,L4,M,O=M3,M4)*(-1.0)**M3
600 - 486.000      CCCCCC PERFORM FINAL INTEGRATION FOR THIS TERM
601 - 487.000      FK=0.0
602 - 488.000      DO 17 J=3,IPTS,2
603 - 489.000      IF (J.NE.3) GO TO 15
604 - 490.000      R1=RMESH(1)
605 - 491.000      F1=RNL1(INX1+1)+RNL2(INX2+1)*SS(1)
606 - 492.000      GO TO 16
607 - 493.000      15 R1=R3
608 - 494.000      F1=F3
609 - 495.000      16 F2=RNL1(INX1+J)+RNL2(INX2+J)*SS(J)
610 - 496.000      R3=RMESH(J)
611 - 497.000      F3=RNL1(INX1+J)+RNL2(INX2+J)*SS(J)
612 - 498.000      17 FK=FK+(R3-R1)/6.0*(F1+4.0*F2+F3)
613 - 499.000      CCCCCC DETERMINE ANGULAR PART OF INTEGRAL FOR THIS TERM
614 - 500.000      C1=YLMINT(K,L1,L2,O=M,O=M1,M2)*(-1.0)**(M+M1)
615 - 501.000      TERM=4.0*PI/FLOAT(2*K+1)*C1*C2*FK
616 - 502.000      IF (DETAIL) WRITE(6,900) M,M1,M2,M,M3,M4,
617 - 503.000      1      K,L1,L2,K,L3,L4,
618 - 504.000      2      C1,C2,FK
619 - 505.000      900 FORMAT('0','M=( ',I2,' ) M1=( ',I2,' ) M2=( ',I2,' )
620 - 506.000      1      ' M=( ',I2,' ) M3=( ',I2,' ) M4=( ',I2,'
621 - 507.000      2      ' K=( ',I2,' ) L1=( ',I2,' ) L2=( ',I2,'
622 - 508.000      3      ' K=( ',I2,' ) L3=( ',I2,' ) L4=( ',I2,'
623 - 509.000      4      ' C =',1PE14.6,' C =',E14.6,' B*FK =',E14.6)
624 - 510.000      EL2CT1=EL2CT1+TERM
625 - 511.000      20 CONTINUE
626 - 512.000      21 RETURN
627 - 513.000      END
628 - 514.000      FUNCTION HYBAAA(LB,MB,ALPHA,NXPAN,LAPP,MAPP,RAPP,INXAPP,
629 - 515.000      1      LA,MA,RA,INXA,LAP,MAP,RAP,INXAP,
630 - 516.000      2      RMESH,IPTS,MXPTS,NXLIM,DETAIL)
631 - 517.000      CCCCCC FUNCTION HYBAAA CALCULATES HYBRID INTEGRAL GIVEN THE ANGULAR
632 - 518.000      CCCCCC QUANTUM NUMBERS, THE ALPHA EXPANSION FUNCTIONS FOR CENTER
633 - 519.000      CCCCCC B, AND THE RADIAL FUNCTIONS R(A) FOR CENTER A.
634 - 520.000      CCCCCC WRITTEN BY D. J. MICKISH, JUNE 1972.
635 - 521.000      DIMENSION ALPHA(MXPTS,NXLIM),RAPP(1),
636 - 522.000      1      RA(1),RAP(1),RMESH(1)
637 - 523.000      COMMON /TWOEL/PAPAP(161),SS(161)
638 - 524.000      CCCCCC DETAIL IS USED TO CAUSE PRINTING OF TERMS OF RESULTING SUM
639 - 525.000      LOGICAL DETAIL
640 - 526.000      XN(L,M)=SQRT(FLOAT(2*L+1)*FACT(L+IABS(M))/(2.0*FACT(L+IABS(M))))
641 - 527.000      KLIM=5
642 - 528.000      IPTSLM=161
643 - 529.000      HYBAAA=0.0
644 - 530.000      CCCCCC POLAR ANGLE CONDITION
645 - 531.000      M=MAPP-MB
646 - 532.000      IF (M.EQ.MA-MAP) GO TO 97
647 - 533.000      WRITE(6,96) MA,MAP,MB,MAPP
648 - 534.000      96 FORMAT(' ',MA=' ',I2,' MAP =',I2,' MB =',I2,' MAPP =',I2)
649 - 535.000      RETURN
650 - 536.000      97 IF (IPTS.LE.IPTSLM) GO TO 99
651 - 537.000      WRITE(6,98) IPTS
652 - 538.000      98 FORMAT(' ',NUMBER OF INTEGRATION POINTS =',I4,' EXCEEDS LIMIT ',
653 - 539.000      1      ' IN FUNCTION SUBPROGRAM HYBAAA')
654 - 540.000      RETURN
655 - 541.000      99 PI=3.141592653589793
656 - 542.000      DO 100 J=1,IPTS
657 - 543.000      100 PAPAP(J)=RA(INXA+J)*RAP(INXAP+J)*RMESH(J)**2
658 - 544.000      LAMPLAP=IABS(LA-LAP)

```

```

659 - 545.000 LALAP=LA+LAP
660 - 546.000 KMIN=LALAP+1
661 - 547.000 IM=IABS(M)
662 - 548.000 IF (KMIN.GE.IM+1) GO TO 5
663 - 549.000 KMIN=IM+1
664 - 550.000 CCCCC IF KMIN IS INFLUENCED BY M, THEN CORRECT PARITY MUST BE
665 - 551.000 CCCCC CHECKED.
666 - 552.000 IWORK=IM+LALAP
667 - 553.000 IF (IWORK.NE.2*(IWORK/2)) KMIN=KMIN+1
668 - 554.000 5 KMAX=LALAP+1
669 - 555.000 IF (KMIN.GT.KMAX) GO TO 21
670 - 556.000 DO 20 K1=KMIN,KMAX,2
671 - 557.000 K=K1-1
672 - 558.000 CALL IR12K(PAPAP,K,SS,RMESH,IPTS,0,0)
673 - 559.000 DO 1402 J=1,IPTS
674 - 560.000 1402 SS(J)=SS(J)*RMESH(J)**2
675 - 561.000 C2=YLMINT(K,LA,LAP,M,0=MA,MAP)*(=1,0)**MA
676 - 562.000 KLAPP=K+LAPP
677 - 563.000 IMIN=IABS(K-LAPP)+1
678 - 564.000 IMB=IABS(MB)
679 - 565.000 IF (IMIN.GE.IMB+1) GO TO 1405
680 - 566.000 IMIN=IMB+1
681 - 567.000 IWORK=IMB+KLAPP
682 - 568.000 IF (IWORK.NE.2*(IWORK/2)) IMIN=IMIN+1
683 - 569.000 1405 IMAX=MINO(KLAPP+1,NXPAN)
684 - 570.000 IF (IMIN.GT.IMAX) GO TO 20
685 - 571.000 DO 19 I1=IMIN,IMAX,2
686 - 572.000 I=I1-1
687 - 573.000 CCCCC PERFORM FINAL INTEGRATION FOR THIS TERM
688 - 574.000 FK=0.0
689 - 575.000 DO 17 J=3,IPTS,2
690 - 576.000 IF (J.NE.3) GO TO 15
691 - 577.000 R1=RMESH(J=2)
692 - 578.000 F1=ALPHA(J=2,I+1)*RAPP(INXAPP+J-2)*SS(J=2)
693 - 579.000 GO TO 16
694 - 580.000 15 R1=R3
695 - 581.000 F1=F3
696 - 582.000 16 F2=ALPHA(J=1,I+1)*RAPP(INXAPP+J-1)*SS(J=1)
697 - 583.000 R3=RMESH(J)
698 - 584.000 F3=ALPHA(J,I+1)*RAPP(INXAPP+J)*SS(J)
699 - 585.000 17 FK=FK+(R3=R1)/6.0*(F1+.0*F2+F3)
700 - 586.000 CCCCC DETERMINE ANGULAR PART OF INTEGRAL FOR THIS TERM
701 - 587.000 C1=YLMINT(K,I,LAPP,0=M,0=MB,MAPP)*(=1,0)**(M+MB)
702 - 588.000 TERM=4.0*PI/FL0AT(2+K+1)*C1*C2*FK
703 - 589.000 IF (DETAIL) WRITE(6,900) M,MB,MAPP,MA,MAP,
704 - 590.000 1 K,I,LAPP,K,LA,LAP,
705 - 591.000 2 C1,C2,FK
706 - 592.000 900 FORMAT('0','M=(-)',I2,' MB=(-)',I2,' MAPP=',I2,
707 - 593.000 1 ' M=',I2,' MA=(-)',I2,' MAP=',I2/
708 - 594.000 2 ' ', ' K=',I2,' I=',I2,' LAPP=',I2,
709 - 595.000 3 ' K=',I2,' LA=',I2,' LAP=',I2/
710 - 596.000 4 ' ', ' C =',1PE14.6,' ' C =',1PE14.6,' FK =',E14.6)
711 - 597.000 HYBAAA=HYBAAA+TERM/(XN(I,MB))
712 - 598.000 19 CONTINUE
713 - 599.000 20 CONTINUE
714 - 600.000 HYBAAA=HYBAAA*XN(LB,MB)
715 - 601.000 21 RETURN
716 - 602.000 END
717 - 603.000 FUNCTION COUL(LB,MB,ALPHA,NXPAN,LBP,MBP,ALPHAP,NXPANP,
718 - 604.000 1 LA,MA,RA,INXA,LAP,MAP,RAP,INXAP,

```

```

719 - 605.000      2          RMESH,IPTS,MXPTS,NXLIM,MXPTSP,NXPLIM,DETAIL)
720 - 606.000 CCCCC FUNCTION COUL CALCULATES COULOMB INTEGRALS GIVEN THE ANGULAR
721 - 607.000 CCCCC          QUANTUM NUMBERS, THE ALPHA EXPANSION FUNCTIONS FOR CENTER
722 - 608.000 CCCCC          B, AND THE RADIAL FUNCTIONS R(A) FOR CENTER A.
723 - 609.000 CCCCC          WRITTEN BY D. J. MICKISH, FEBRUARY 1972.
724 - 610.000          DIMENSION ALPHA(MXPTS,NXLIM),ALPHAP(MXPTSP,NXPLIM),
725 - 611.000      1          RA(1),RAP(1),RMESH(1)
726 - 612.000          COMMON /TWDEL/PAPAP(161),SS(161)
727 - 613.000 CCCCC          DETAIL IS USED TO CAUSE PRINTING OF TERMS OF RESULTING SUM
728 - 614.000          LOGICAL DETAIL
729 - 615.000          XN(L,M)=SQRT(REAL(2*L+1)*FACT(L-IABS(M))/(2.0*FACT(L+IABS(M))))
730 - 616.000          IPTSLM=161
731 - 617.000          COUL=0.0
732 - 618.000 CCCCC          POLAR ANGLE CONDITION
733 - 619.000          M=MBP=MB
734 - 620.000          IF (M.EQ.MA=MAP) GO TO 97
735 - 621.000          WRITE(6,96) MA,MAP,MB,MBP
736 - 622.000      96  FORMAT(' ', 'MA =',I2,' MAP =',I2,' MB =',I2,' MBP =',I2)
737 - 623.000          RETURN
738 - 624.000      97  IF (IPTS.LE.IPTSLM) GO TO 99
739 - 625.000          WRITE(6,98) IPTS
740 - 626.000      98  FORMAT(' ', 'NUMBER OF INTEGRATION POINTS =',I4,' EXCEEDS LIMIT ',
741 - 627.000      1          'IN FUNCTION SUBPROGRAM COUL')
742 - 628.000          RETURN
743 - 629.000      99  PI=3.141592653589793
744 - 630.000          DO 100 J=1,IPTS
745 - 631.000      100  PAPAP(J)=RA(INXA+J)*RAP(INXAP+J)*RMESH(J)**2
746 - 632.000          LALAP=LA+LAP
747 - 633.000          LAMLAP=IABS(LA-LAP)
748 - 634.000          IM=IABS(M)
749 - 635.000          IF (KMIN.GE.IM+1) GO TO 5
750 - 636.000          KMIN=IM+1
751 - 637.000 CCCCC          IF KMIN IS INFLUENCED BY M, THEN CORRECT PARITY MUST BE
752 - 638.000 CCCCC          CHECKED.
753 - 639.000          IWORK=IM+LALAP
754 - 640.000          IF (IWORK.NE.2*(IWORK/2)) KMIN=KMIN+1
755 - 641.000      5  KMAX=LALAP+1
756 - 642.000          IF (KMIN.GT.KMAX) GO TO 22
757 - 643.000 CCCCC          K INCREMENTS BY TWO DUE TO PARITY CONDITIONS
758 - 644.000          DO 21 K1=KMIN,KMAX,2
759 - 645.000          K=K1-1
760 - 646.000          CALL IR12K(PAPAP,K,SS,RMESH,IPTS,0.0)
761 - 647.000          DO 1402 J=1,IPTS
762 - 648.000      1402  SS(J)=SS(J)*RMESH(J)**2
763 - 649.000          C2=YLMINT(K,LA,LAP,M,0=MA,MAP)*(=1.0)**MA
764 - 650.000          IMIN=IABS(MB)+1
765 - 651.000          DO 20 I1=IMIN,NXPAN
766 - 652.000          I=I1-1
767 - 653.000          IK=I+K
768 - 654.000          IPMIN=IABS(I=K)+1
769 - 655.000          IMBP=IABS(MBP)
770 - 656.000          IF (IPMIN.GE.IMBP+1) GO TO 1405
771 - 657.000          IPMIN=IMBP+1
772 - 658.000          IWORK=IMBP+IK
773 - 659.000          IF (IWORK.NE.2*(IWORK/2)) IPMIN=IPMIN+1
774 - 660.000      1405  IPMAX=MINO(IK+1,NXPANP)
775 - 661.000          IF (IPMIN.GT.IPMAX) GO TO 20
776 - 662.000          DO 19 IP1=IPMIN,IPMAX,2
777 - 663.000          IP=IP1-1
778 - 664.000 CCCCC          PERFORM FINAL INTEGRATION FOR THIS TERM

```

```

779 - 665.000      FK=0.0
780 - 666.000      DM 17 J=3,IPTS,2
781 - 667.000      IF (J.NE.3) GO TO 15
782 - 668.000      R1=RMESS(J=2)
783 - 669.000      F1=ALPHA(J=2,I+1)*ALPHAP(J=2,IP+1)*SS(J=2)
784 - 670.000      GO TO 16
785 - 671.000      15 R1=R3
786 - 672.000      F1=F3
787 - 673.000      16 F2=ALPHA(J=1,I+1)*ALPHAP(J=1,IP+1)*SS(J=1)
788 - 674.000      R3=RMESS(J)
789 - 675.000      F3=ALPHA(J,I+1)*ALPHAP(J,IP+1)*SS(J)
790 - 676.000      17 FK=FK+(R3-R1)/6.0*(F1+4.0*F2+F3)
791 - 677.000      CCCCC      DETERMINE ANGULAR PART OF INTEGRAL FOR THIS TERM
792 - 678.000      C1=YLMINT(K,I,IP,Q=M,Q=MB,MBP)*(=1,0)**(M+MB)
793 - 679.000      TERM=4.0*PI/FL0AT(2*K+1)*C1*C2*FK
794 - 680.000      IF (DETAIL) WRITE(6,900) M,MB,MBP,MA,MAP,
795 - 681.000      1          K,I,IP,K,LA,LAP,
796 - 682.000      2          C1,C2,FK
797 - 683.000      900 FORMAT('0','M=(=)',I2,' MB=(=)',I2,' MBP=',I2,
798 - 684.000      1          ' M=',I2,' MA=(=)',I2,' MAP=',I2/
799 - 685.000      2          ' ', ' K=',I2,' I=',I2,' IP=',I2/
800 - 686.000      3          ' K=',I2,' LA=',I2,' LAP=',I2/
801 - 687.000      4          ' ', ' C =',1PE14.6,' C =',E14.6,8X,'FK =',E14.6)
802 - 688.000      C0UL=C0UL+TERM/(XN(I,MB)*XN(IP,MBP))
803 - 689.000      19 CONTINUE
804 - 690.000      20 CONTINUE
805 - 691.000      21 CONTINUE
806 - 692.000      C0UL=C0UL*XN(LB,MB)*XN(LBP,MBP)
807 - 693.000      22 RETURN
808 - 694.000      END
809 - 695.000      FUNCTION YLMINT(L,L1,L2,M,M1,M2)
810 - 696.000      CCCCC      FUNCTION YLMINT(L,L1,L2,M,M1,M2) DETERMINES THE VALUE OF THE
811 - 697.000      CCCCC      INTEGRAL OVER ALL SOLID ANGLES OF THREE COMPLEX SPHERICAL
812 - 698.000      CCCCC      HARMONICS Y(L,M),Y(L1,M1),Y(L2,M2). THE PROGRAM CALLS
813 - 699.000      CCCCC      SUBPROGRAM CLEB WHICH SUPPLIES ALL NECESSARY
814 - 700.000      CCCCC      CLEBSCH=GORDON COEFFICIENTS.
815 - 701.000      CCCCC      WRITTEN BY D. J. MICKISH, FEBRUARY, 1972.
816 - 702.000      CCCCC      REFERENCE: A. MESSIAH, QUANTUM MECHANICS, VOL. 1, PG. 2057,
817 - 703.000      CCCCC      JOHN WILEY, 1966.
818 - 704.000      YLMINT=0.0
819 - 705.000      CCCCC      POLAR ANGLE CONDITION
820 - 706.000      IF (M+M1+M2.EQ.0) GO TO 2
821 - 707.000      WRITE(6,1) M,M1,M2
822 - 708.000      1 FORMAT(' ',M=',I2,' M1=',I2,' M2=',I2,' IN FUNCTION 1,
823 - 709.000      1          'SUBPROGRAM YLMINT')
824 - 710.000      RETURN
825 - 711.000      CCCCC      TRIANGLE CONDITION
826 - 712.000      2 IF (IABS(L1-L2).LE.L.AND.L.LE.L1+L2) GO TO 5
827 - 713.000      3 WRITE(6,4) L,L1,L2
828 - 714.000      4 FORMAT(' ',L=',I2,' L1=',I2,' L2=',I2,' IN FUNCTION 1,
829 - 715.000      1          'SUBPROGRAM YLMINT')
830 - 716.000      RETURN
831 - 717.000      CCCCC      PARITY CONDITION
832 - 718.000      5 IWORK=L+L1+L2
833 - 719.000      IF (IWORK.NE.2*(IWORK/2)) GO TO 3
834 - 720.000      FOURPI=4.0+3.141592653589793
835 - 721.000      AL1=FL0AT(L1)
836 - 722.000      AL2=FL0AT(L2)
837 - 723.000      AL=FL0AT(L)
838 - 724.000      AM1=FL0AT(M1)

```

```

839 - 725.000      AM2=FLOAT(M2)
840 - 726.000      AM=FLOAT(M)
841 - 727.000      C1=CLEB(AL1,AL2,AL,AM1,AM2,0.0-AM)
842 - 728.000      C2=CLEB(AL1,AL2,AL,0.0,0.0,0.0)
843 - 729.000      YLMINT=SQRT((2.0*AL1+1)*(2.0*AL2+1)/((2.0*AL+1)*FOURPI))*C1*C2
844 - 730.000      1      *(-1.0)**M
845 - 731.000      RETURN
846 - 732.000      END
847 - 733.000      FUNCTION CLEB(AJ1,AJ2,AJ3,AM1,AM2,AM3)
848 - 734.000      REAL*8 X,G(102),A,B,H,D,E,F,S,Q,T,C
849 - 735.000      DIMENSION I(11)
850 - 736.000      EQUIVALENCE (I(1),I1),(I(2),I2),(I(3),I3),(I(4),I4),(I(5),I5),
851 - 737.000      1      (I(6),I6),(I(7),I7),(I(8),I8),(I(9),I9),(I(10),I10),
852 - 738.000      2      (I(11),I11)
853 - 739.000      LOGICAL LLNFST
854 - 740.000      IM=102
855 - 741.000      DATA LLNFST/.FALSE./
856 - 742.000      CLEB=0.0
857 - 743.000      C      FORM THE FACTORIAL TABLE ON THE FIRST PASS
858 - 744.000      IF (LLNFST) GO TO 30
859 - 745.000      10  LLNFST=.TRUE.
860 - 746.000      G(1)=0.0D+0
861 - 747.000      G(2)=0.0D+0
862 - 748.000      DO 20 J=3,IM
863 - 749.000      X=DFLOAT(J-1)
864 - 750.000      20  G(J)=G(J-1)+DLOG(X)
865 - 751.000      C      CONVERT THE ARGUMENTS TO INTEGER
866 - 752.000      30  J1=IFIX(2.0*AJ1)
867 - 753.000      J2=IFIX(2.0*AJ2)
868 - 754.000      J3=IFIX(2.0*AJ3)
869 - 755.000      M1=IFIX(2.0*AM1)
870 - 756.000      M2=IFIX(2.0*AM2)
871 - 757.000      M3=IFIX(2.0*AM3)
872 - 758.000      C      TEST M1+M2=M3
873 - 759.000      IF (M1+M2=M3) 900,40,900
874 - 760.000      C      TEST TABLE SIZE
875 - 761.000      40  I(10)=(J1+J2+J3)/2+2
876 - 762.000      N=I(10)
877 - 763.000      I(11)=J3+2
878 - 764.000      IF (I(10)=IM) 70,70,50
879 - 765.000      50  WRITE(6,60) I(10),IM
880 - 766.000      60  FORMAT('0','INSUFFICIENT DLOG(N FACT) TABLE IN FUNCTION CLEB',
881 - 767.000      1      ' N REQ. =',I4,' N MAX =',I4)
882 - 768.000      GO TO 920
883 - 769.000      70  I(1)=J1+J2-J3
884 - 770.000      I(2)=J2+J3-J1
885 - 771.000      I(3)=J3+J1-J2
886 - 772.000      I(4)=J1-M1
887 - 773.000      I(5)=J1+M1
888 - 774.000      I(6)=J2-M2
889 - 775.000      I(7)=J2+M2
890 - 776.000      I(8)=J3-M3
891 - 777.000      I(9)=J3+M3
892 - 778.000      C      CHECK I(J)=EVEN, TRIANGULAR INEQUALITY, M LESS J, FIND NO OF TERMS
893 - 779.000      DO 110 J=1,9
894 - 780.000      K=I(J)/2
895 - 781.000      IF (I(J)=2*K) 900,80,900
896 - 782.000      80  IF (K) 900,90,90
897 - 783.000      90  IF (K=N) 100,110,110
898 - 784.000      100 N=K

```

```

899 - 785.000 110 I(J)=K+1
900 - 786.000 112 IF (M3) 115,400,115
901 - 787.000 115 IL=0
902 - 788.000 LA=I1-I5
903 - 789.000 LR=I1-I6
904 - 790.000 IF (IL-LA) 120,130,130
905 - 791.000 120 IL=LA
906 - 792.000 130 IF (IL-LB) 140,145,145
907 - 793.000 140 IL=LB
908 - 794.000 C FORM COEFFICIENT OF SUM
909 - 795.000 145 C=(G(I11)-G(I11-1)+G(I1)+G(I2)+G(I3)-G(I10)+G(I4)+G(I5)+G(I6)+
910 - 796.000 1 G(I7)+G(I8)+G(I9))/2.0D+0
911 - 797.000 J1=I1-IL
912 - 798.000 J2=I4-IL
913 - 799.000 J3=I7-IL
914 - 800.000 M1=IL+1
915 - 801.000 M2=IL-LA+1
916 - 802.000 M3=IL-LB+1
917 - 803.000 C=C-G(J1)-G(J2)-G(J3)-G(M1)-G(M2)-G(M3)
918 - 804.000 C=(-1.0D+0)**IL*DEXP(C)
919 - 805.000 IF (N) 900,150,160
920 - 806.000 150 CLEB=SNGL(C)
921 - 807.000 GO TO 300
922 - 808.000 C FORM SUM
923 - 809.000 160 A=DFLOAT(J1-1)
924 - 810.000 B=DFLOAT(J2-1)
925 - 811.000 H=DFLOAT(J3-1)
926 - 812.000 D=DFLOAT(M1)
927 - 813.000 E=DFLOAT(M2)
928 - 814.000 F=DFLOAT(M3)
929 - 815.000 S=1.0D+0
930 - 816.000 Q=DFLOAT(N-1)
931 - 817.000 DO 170 J=1,N
932 - 818.000 T=(A-Q)/(D+Q)*(B-Q)/(E+Q)*(H-Q)/(F+Q)
933 - 819.000 S=1.0D+0+S*T
934 - 820.000 Q=Q-1.0D+0
935 - 821.000 170 CONTINUE
936 - 822.000 CLEB=SNGL(C+S)
937 - 823.000 300 RETURN
938 - 824.000 C SPECIAL FORMULA FOR M3=0 AND M1=0 OR 1/2
939 - 825.000 400 K=I10/2
940 - 826.000 IF (I10-2*K) 410,420,410
941 - 827.000 410 K=1
942 - 828.000 GO TO 430
943 - 829.000 420 K=0
944 - 830.000 430 IF (M1) 115,440,460
945 - 831.000 440 L=0
946 - 832.000 IF (K) 900,480,300
947 - 833.000 460 IF (M1=1) 115,470,115
948 - 834.000 470 L=1
949 - 835.000 480 X=DFLOAT(L)
950 - 836.000 M=I3+(I1+K+1)/2=L
951 - 837.000 M1=I10/2+K
952 - 838.000 M2=I4+I5
953 - 839.000 M3=I6+I7
954 - 840.000 J1=(I1+1-K)/2
955 - 841.000 J2=(I2+1+K-L)/2
956 - 842.000 J3=(I3+1+K-L)/2
957 - 843.000 C=(-1.0D+0)**M*DEXP((G(I11)-G(I11-1)+G(I1)+G(I2)+G(I3)-G(I10))/
958 - 844.000 1 2.0D+0+G(M1)+G(M2)-G(J2)-G(J3)+

```

```

959 - 845.000      2      X*(G(3)=(G(M2)-G(M2+1)+G(M3)-G(M3+1))/2.00+0))
960 - 846.000      CLEB=SNGL(C)
961 - 847.000      GO TO 300
962 - 848.000      900 WRITE(6,910)
963 - 849.000      910 FORMAT('0','PARITY, SUM, OR TRIANGLE ERROR IN FUNCTION GLEW')
964 - 850.000      920 WRITE(6,930) AJ1,AJ2,AJ3,AM1,AM2,AM3
965 - 851.000      930 FORMAT(' ','J1 =',F5.1,' J2 =',F5.1,' J =',F5.1,3X' M1 =',F5.1,
966 - 852.000      1      ' M2 =',F5.1,' M =',F5.1)
967 - 853.000      RETURN
968 - 854.000      END
969 - 855.000      C!ASSIGN M:BO,(FILE,MEVDRVED),(SAVE)
970 - 856.000      C!FORTRAN LS,BO,ADP
971 - 857.000      SUBROUTINE EVDRVE(N,XX,XY,XKZ,IWT,NVEC,IEVFLE,LLDTAL,
972 - 858.000      1NR0UT,LLMTRX)
973 - 859.000      LOGICAL LLOTAL,LLMTRX
974 - 860.000      COMPLEX SMAT(1156),DUMMAT(1156),CMAT(1156)
975 - 861.000      DIMENSION E(34)
976 - 861.500      COMMON SMAT,DUMMAT,CMAT
977 - 862.000      REAL*8 ZHMAT,ZSMAT,ZCOET,ZE
978 - 863.000      DATA ZHMAT/' ' HMAT/' ' ZSMAT/' ' SMAT/' ' ZCOET/' ' COET/' ' ZE/' ' E/'
979 - 864.000      DATA LIMHS/15/
980 - 865.000      DATA LIM0UT/25/
981 - 866.000      C DIMENSION SMAT,DUMMAT,CMAT(N*N) WHERE N=MBLHAX
982 - 867.000      C E(N)
983 - 868.000      IHSFLE=32
984 - 869.000      MBLHAX=34
985 - 870.000      IF(N.LE.MBLHAX) GO TO 63
986 - 871.000      WRITE(6,900) N,MBLHAX
987 - 872.000      900 FORMAT(' ','N =',I3,' EXCEEDS DIMENSION MBLHAX =',I3,
988 - 873.000      1      ' IN SUBPROGRAM EVDRVE')
989 - 874.000      CALL EXIT
990 - 875.000      63 NN=MBLHAX*MBLHAX
991 - 876.000      LINDIM=N*N
992 - 877.000      C NOTE THAT SMAT IS OVERLAP MATRIX
993 - 878.000      C AND THAT DUMMAT IS IDENTITY MATRIX.
994 - 879.000      C CMAT IS HAMILTONIAN,BUT IT WILL SOON BE DESTROYED.<
995 - 887.000      IF(LLDTAL.OR.LLMTRX)CALL COUT21(CMAT,N,N,ZHMAT)
996 - 888.000      IF(LLDTAL.OR.LLMTRX)CALL COUT21(SMAT,N,N,ZSMAT)
997 - 889.000      LIMHS=LIMHS-1
998 - 890.000      IF(LIMHS.LT.0) GO TO 6420
999 - 891.000      WRITE(IHSFLE) N
1000 - 892.000      WRITE(IHSFLE) (CMAT(I),I=1,LINDIM),(SMAT(I),I=1,LINDIM)
1001 - 893.000      6420 IROOTS=N
1002 - 894.000      CALL DIAG(N,E)
1003 - 895.000      WRITE(IEVFLE)XX,XY,XKZ,IWT,N,IROOTS
1004 - 896.000      WRITE(IEVFLE) (E(J),J=1,IROOTS)
1005 - 897.000      DET=1.0
1006 - 898.000      DO 6430 I=1,IROOTS
1007 - 899.000      6430 DET=DET*E(I)
1008 - 900.000      RATIO=E(1)/E(IROOTS)
1009 - 901.000      WRITE(6,92750) DET,RATIO
1010 - 902.000      92750 FORMAT(' ','DET(S) =',1PE15.8,' ES(1)/ES(IROOTS) =',1PE15.8)
1011 - 903.000      IF(RATIO.EQ.0.0) CALL EXIT
1012 - 903.010      READ(23) SMAT,DUMMAT
1013 - 903.020      REWIND 23
1014 - 903.030      C NOTE SMAT IS HAMILTONIAN
1015 - 903.040      C AND DUMMAT IS OVERLAP MATRIX.
1016 - 904.000      CALL DIAG(N,E)
1017 - 905.000      I0UT=NVEC
1018 - 906.000      LIM0UT=LIM0UT-1

```

```

1019 - 907.000      IF (IOUT.EQ.0.AND.LIMOUT.GT.0) IOUT=IROOTS
1020 - 908.000      IF (IOUT.GT.IROOTS) IOUT=IROOTS
1021 - 909.000      IF (IOUT.GT.0) CALL COUT21(DUMMAT,IOUT,N,ZC0ET)
1022 - 910.000      CALL COUT1(E,N,ZE)
1023 - 911.000      WRITE(IEVFLE)(E(J),J=1,IROOTS),(DUMMAT(I),I=1,LINDIM)
1024 - 912.000      RETURN
1025 - 913.000      END
1026 - 914.000      SUBROUTINE COUT21(A,IOUT,N,Z)
1027 - 915.000      COMPLEX A(1)
1028 - 916.000      REAL*8 Z
1029 - 917.000      WRITE(6,5)
1030 - 918.000      5 FORMAT(/)
1031 - 919.000      DO 1 J=1,IOUT
1032 - 920.000      JSUB=(J-1)*N
1033 - 921.000      1 WRITE(6,2) (Z,I,J,A(JSUB+I),I=1,N)
1034 - 922.000      2 FORMAT(1P3(A6,'(','I2','I2')='',2E14.7))
1035 - 923.000      RETURN
1036 - 924.000      END
1037 - 925.000      SUBROUTINE COUT1(X,N,Z)
1038 - 926.000      REAL*8 Z
1039 - 927.000      DIMENSION X(N)
1040 - 928.000      WRITE(6,1)
1041 - 929.000      1 FORMAT(1H0)
1042 - 930.000      WRITE(6,2) (Z,J,X(J),J=1,N)
1043 - 931.000      2 FORMAT(1P4(1X,A6,'(','I2')='',E14.7))
1044 - 932.000      RETURN
1045 - 933.000      END
1046 - 934.000      SUBROUTINE DIAG(MSID,D)
1047 - 935.000      DIMENSION SCRAT(34),AA(34,34),AR(34,34),
1048 - 936.000      1AI(34,34),D(34),E(34),E2(34),TAU(2,34),ZR(34,34),ZI(34,34),
1049 - 937.000      2BB(34,34),CC(34,34),A(1156),B(1156),C(1156)
1050 - 938.000      COMPLEX A,B,C,SCRAT,AA,BB,CC
1051 - 938.500      COMMON A,B,C
1052 - 939.000      COMMON/DIA/SCRAT
1053 - 940.000      EQUIVALENCE (A,AA),(B(1),BB(1,1),AI(1,1)),(B(579),AB(1,1)),
1054 - 941.000      1(C(1),CC(1,1),ZI(1,1)),(C(579),ZR(1,1))
1055 - 942.000 C      FINDS EIGENVALUES AND EIGENVECTORS OF A(PHI)=E(B)(PHI)
1056 - 943.000 C      USES CHOLESKI DECOMPOSITION
1057 - 944.000 C      PUT A AND B IN FORM
1058 - 945.000      NM=34
1059 - 946.000      N=MSID
1060 - 947.000      DO 1 I=1,N
1061 - 948.000      DO 1 J=1,N
1062 - 949.000      K=(I-1)*N+J
1063 - 950.000      1 CC(I,J)=A(K)
1064 - 951.000      DO 2 I=1,N
1065 - 952.000      DO 2 J=1,N
1066 - 953.000      2 AA(I,J)=CC(I,J)
1067 - 954.000      DO 3 I=1,N
1068 - 955.000      DO 3 J=1,N
1069 - 956.000      K=(I-1)*N+J
1070 - 957.000      3 CC(I,J)=B(K)
1071 - 958.000      DO 4 I=1,N
1072 - 959.000      DO 4 J=1,N
1073 - 960.000      4 BB(I,J)=CC(I,J)
1074 - 961.000 C      MATRICES IN 2 DIMENSIONAL FORM DECOMPOSE B VIA CHOLESKI
1075 - 962.000      CALL LB(N,B)
1076 - 963.000      CALL LBINV(N,B)
1077 - 964.000 C      L-1 FORMED IN BB FORM L-1+ AND PUT ON FILE 13
1078 - 965.000      DO 5 I=1,N

```

```

1079 - 966.000      DO 6 J=1,N
1080 - 967.000      6 CC(I,J)= CONJG(BB(J,I))
1081 - 968.000      5 CONTINUE
1082 - 969.000      WRITE (13)(CC)
1083 - 970.000      C  FORM L=1H INTO CC
1084 - 971.000      DO 7 I=1,N
1085 - 972.000      DO 7 J=1,N
1086 - 973.000      CC(I,J)=  CMLX(0.0,0.0)
1087 - 974.000      DO 8 K=1,N
1088 - 975.000      8 CC(I,J)=BB(I,K)*AA(K,J)+CC(I,J)
1089 - 976.000      7 CONTINUE
1090 - 977.000      C  FORM L=1H(L=1)+ INTO AA
1091 - 978.000      DO 9 I=1,N
1092 - 979.000      DO 9 J=1,N
1093 - 980.000      AA(I,J)=CMLX(0.0,0.0)
1094 - 981.000      DO 10 K=1,N
1095 - 982.000      10 AA(I,J)=AA(I,J)+CC(I,K)*CONJG(BB(J,K))
1096 - 983.000      9 CONTINUE
1097 - 984.000      DO 11 I=1,N
1098 - 985.000      DO 11 J=1,N
1099 - 986.000      AR(I,J)=REAL(AA(I,J))
1100 - 987.000      11 AI(I,J)=AIMAG(AA(I,J))
1101 - 988.000      C  WE MAY DIAGONALIZE NOW
1102 - 989.000      CALL HTRIDI(NM,N,AR,AI,D,E,E2,TAU)
1103 - 990.000      DO 12 I=1,NM
1104 - 991.000      DO 12 J=1,NM
1105 - 992.000      ZR(I,J)=DEL(I,J)
1106 - 993.000      IF (I.GT.N.OR.J.GT.N) ZR(I,J)=0.0
1107 - 994.000      12 CONTINUE
1108 - 995.000      CALL IMTQL2(NM,N,D,E,ZR,IABKER)
1109 - 996.000      DO 101 I=1,NM
1110 - 997.000      DO 101 J=1,NM
1111 - 998.000      101 ZI(I,J)=0.0
1112 - 999.000      CALL HTRIBK(NM,N,AR,AI,TAU,N,ZR,ZI)
1113 - 1000.000     C  EIGENVALUES IN D
1114 - 1001.000     C  ROTATED EIGEN VECTORS IN ZR,ZI  BACK TRANSFORM INTO CC
1115 - 1002.000     DO 13 I=1,N
1116 - 1003.000     DO 13 J=1,N
1117 - 1004.000     13 AA(I,J)=CMLX(ZR(I,J),ZI(I,J))
1118 - 1005.000     REWIND 13
1119 - 1006.000     READ (13)(BB)
1120 - 1007.000     DO 14 I=1,N
1121 - 1008.000     DO 14 J=1,N
1122 - 1009.000     CC(I,J)=CMLX(0.0,0.0)
1123 - 1010.000     DO 15 K=1,N
1124 - 1011.000     15 CC(I,J)=CC(I,J)+BB(I,K)*AA(K,J)
1125 - 1012.000     14 CONTINUE
1126 - 1013.000     DO 16 I=1,N
1127 - 1014.000     DO 16 J=1,N
1128 - 1015.000     K=(I-1)*N+J
1129 - 1016.000     16 R(K)=CC(J,I)
1130 - 1017.000     RETURN
1131 - 1018.000     END
1132 - 1019.000     SUBROUTINE LB(N,A)
1133 - 1020.000     DIMENSION A(34,34),B(34)
1134 - 1021.000     COMPLEX A,B,SUM
1135 - 1022.000     COMMON/DIA/B
1136 - 1023.000     C  FORMS LOWER TRIANGULAR MATRIX FOR CHOLESKI DECOMPOSITION
1137 - 1024.000     DO 1 I=1,N
1138 - 1025.000     DO 2 J=1,N

```

```

1139 = 1026.000 IF (J.GT.I) A(I,J)=CMPLX(0.0,0.0);GO TO 2
1140 = 1027.000 IF (I.EQ.J) GO TO 3
1141 = 1028.000 C J IS LESS THAN I
1142 = 1029.000 SUM=AL(I,J)
1143 = 1030.000 IF (J.EQ.1) GO TO 4
1144 = 1031.000 JJ=J-1
1145 = 1032.000 DO 5 K=1,JJ
1146 = 1033.000 5 SUM=SUM+B(K)*CONJG(A(J,K))
1147 = 1034.000 4 B(J)=SUM/CONJG(A(J,J))
1148 = 1035.000 GO TO 2
1149 = 1036.000 3 SUM=AL(I,J)
1150 = 1037.000 IF (J.EQ.1) GO TO 6
1151 = 1038.000 JJ=J-1
1152 = 1039.000 DO 7 K=1,JJ
1153 = 1040.000 7 SUM=SUM+B(K)*CONJG(B(K))
1154 = 1041.000 6 RSUM=REAL(SUM)
1155 = 1042.000 RSUM=SGRT(RSUM)
1156 = 1043.000 B(J)=CMPLX(RSUM,0.0)
1157 = 1044.000 2 CONTINUE
1158 = 1045.000 DO 1 J=1,I
1159 = 1046.000 1 A(I,J)=B(J)
1160 = 1047.000 RETURN
1161 = 1048.000 END
1162 = 1049.000 SUBROUTINE LQINV(N,B)
1163 = 1050.000 DIMENSION B(3*,3*),C(3*)
1164 = 1051.000 COMMON/DIA/ C
1165 = 1052.000 COMPLEX B,C,SUM
1166 = 1053.000 C INVERTS LOWER TRIANGULAR MATRIX INPUT AND OUTPUT IN B SCRATCH INC
1167 = 1054.000 DO 1 I=1,N
1168 = 1055.000 DO 2 J=1,N
1169 = 1056.000 IF (J.GT.I) C(J)=CMPLX(0.0,0.0);GO TO 2
1170 = 1057.000 I=I-1
1171 = 1058.000 SUM=CMPLX(DEL(I,J),0.0)
1172 = 1059.000 IF (I.EQ.0) GO TO 3
1173 = 1060.000 DO 4 K=1,I
1174 = 1061.000 4 SUM=SUM-B(I,K)*B(K,J)
1175 = 1062.000 3 C(J)=SUM/B(I,I)
1176 = 1063.000 2 CONTINUE
1177 = 1064.000 DO 1 J=1,N
1178 = 1065.000 1 B(I,J)=C(J)
1179 = 1066.000 RETURN
1180 = 1067.000 END
1181 = 1068.000 SUBROUTINE HTRIDI(NM,N,AR,AI,D,E,E2,TAU)
1182 = 1069.000 C
1183 = 1070.000 INTEGER I,J,K,L,N,II,NM,JPI
1184 = 1071.000 REAL AR(NM,NM),AI(NM,NM),D(NM),E(NM),E2(NM),TAU(2,NM)
1185 = 1072.000 REAL F,F1,G,GT,H,H1,SI,SCALE
1186 = 1073.000 C REAL SORT,CABS,ABS
1187 = 1074.000 C COMPLEX CMPLX
1188 = 1075.000 C
1189 = 1076.000 C
1190 = 1077.000 C
1191 = 1078.000 C
1192 = 1079.000 C
1193 = 1080.000 C
1194 = 1081.000 C
1195 = 1082.000 C
1196 = 1083.000 C
1197 = 1084.000 C
1198 = 1085.000 C

```

THIS SUBROUTINE IS A TRANSLATION OF A COMPLEX ANALOGUE OF THE ALGOL PROCEDURE TRED1, NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON. HANDBOOK FOR AUTO. COMP., VOL. II-LINEAR ALGEBRA, 212-226(1971).

THIS SUBROUTINE REDUCES A COMPLEX HERMITIAN MATRIX TO A REAL SYMMETRIC TRIAGONAL MATRIX USING UNITARY SIMILARITY TRANSFORMATIONS.

ON INPUT=

8\*215005  
8\*215006  
8\*215008  
8\*215009  
8\*215010  
8\*215011  
8\*215012  
8\*215013  
8\*215014  
8\*215015  
8\*215016  
8\*215017  
8\*215018  
8\*215019  
8\*215020  
8\*215021

```

1199 - 1086.000 C
1200 - 1087.000 C
1201 - 1088.000 C
1202 - 1089.000 C
1203 - 1090.000 C
1204 - 1091.000 C
1205 - 1092.000 C
1206 - 1093.000 C
1207 - 1094.000 C
1208 - 1095.000 C
1209 - 1096.000 C
1210 - 1097.000 C
1211 - 1098.000 C
1212 - 1099.000 C
1213 - 1100.000 C
1214 - 1101.000 C
1215 - 1102.000 C
1216 - 1103.000 C
1217 - 1104.000 C
1218 - 1105.000 C
1219 - 1106.000 C
1220 - 1107.000 C
1221 - 1108.000 C
1222 - 1109.000 C
1223 - 1110.000 C
1224 - 1111.000 C
1225 - 1112.000 C
1226 - 1113.000 C
1227 - 1114.000 C
1228 - 1115.000 C
1229 - 1116.000 C
1230 - 1117.000 C
1231 - 1118.000 C
1232 - 1119.000 C
1233 - 1120.000 C
1234 - 1121.000 C
1235 - 1122.000 C
1236 - 1123.000 C
1237 - 1124.000 C
1238 - 1125.000 C
1239 - 1126.000 C
1240 - 1127.000 C
1241 - 1128.000 C
1242 - 1129.000 C
1243 - 1130.000 C
1244 - 1131.000 C
1245 - 1132.000 C
1246 - 1133.000 C
1247 - 1134.000 C
1248 - 1135.000 C
1249 - 1136.000 C
1250 - 1137.000 C
1251 - 1138.000 C
1252 - 1139.000 C
1253 - 1140.000 C
1254 - 1141.000 C
1255 - 1142.000 C
1256 - 1143.000 C
1257 - 1144.000 C
1258 - 1145.000 C

      NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
      ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
      DIMENSION STATEMENT,
      N IS THE ORDER OF THE MATRIX,
      AR AND AI CONTAIN THE REAL AND IMAGINARY PARTS,
      RESPECTIVELY, OF THE COMPLEX HERMITIAN INPUT MATRIX,
      ONLY THE LOWER TRIANGLE OF THE MATRIX NEED BE SUPPLIED,
      AN OUTPUT=
      AR AND AI CONTAIN INFORMATION ABOUT THE UNITARY TRANS-
      FORMATIONS USED IN THE REDUCTION IN THEIR FULL LOWER
      TRIANGLES. THEIR STRICT UPPER TRIANGLES AND THE
      DIAGONAL OF AR ARE UNALTERED,
      D CONTAINS THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX,
      E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE TRIDIAGONAL
      MATRIX IN ITS LAST N-1 POSITIONS. E(1) IS SET TO ZERO,
      E2 CONTAINS THE SQUARES OF THE CORRESPONDING ELEMENTS OF E.
      E2 MAY COINCIDE WITH E IF THE SQUARES ARE NOT NEEDED,
      TAU CONTAINS FURTHER INFORMATION ABOUT THE TRANSFORMATIONS.
      ARITHMETIC IS REAL EXCEPT FOR THE USE OF THE SUBROUTINES
      CABS AND CMLX IN COMPUTING COMPLEX ABSOLUTE VALUES.
      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,
      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
      ***** FOR I=N STEP =1 UNTIL 1 DO == *****
      DO 300 I1 = 1, N
      I = N + 1 - I1
      L = I - 1
      M = 0 + 0
      SCALE = 0 + 0
      IF (L .LT. 1) GO TO 130
      ***** SCALE ROW (ALGOL TOL THEN NOT NEEDED) *****
      DO 120 K = 1, L
      SCALE = SCALE + ABS(AR(I,K)) + ABS(AI(I,K))
      IF (SCALE .NE. 0 + 0) GO TO 140
      TAU(14L) = 1 + 0
      TAU(24L) = 0 + 0
      E(1) = 0 + 0
      E(2) = 0 + 0
      GO TO 290
      *****
      TAU(1,N) = 1 + 0
      TAU(2,N) = 0 + 0
      DO 100 I = 1, N
      D(I) = AR(I,I)
      *****
      D(1) = 128.000 C
      D(2) = 129.000 C
      D(3) = 130.000 C
      D(4) = 131.000 C
      D(5) = 132.000 C
      D(6) = 133.000 C
      D(7) = 134.000 C
      D(8) = 135.000 C
      D(9) = 136.000 C
      D(10) = 137.000 C
      D(11) = 138.000 C
      D(12) = 139.000 C
      D(13) = 140.000 C
      D(14) = 141.000 C
      D(15) = 142.000 C
      D(16) = 143.000 C
      D(17) = 144.000 C
      D(18) = 145.000 C
      D(19) = 146.000 C
      D(20) = 147.000 C
      D(21) = 148.000 C
      D(22) = 149.000 C
      D(23) = 150.000 C
      D(24) = 151.000 C
      D(25) = 152.000 C
      D(26) = 153.000 C
      D(27) = 154.000 C
      D(28) = 155.000 C
      D(29) = 156.000 C
      D(30) = 157.000 C
      D(31) = 158.000 C
      D(32) = 159.000 C
      D(33) = 160.000 C
      D(34) = 161.000 C
      D(35) = 162.000 C
      D(36) = 163.000 C
      D(37) = 164.000 C
      D(38) = 165.000 C
      D(39) = 166.000 C
      D(40) = 167.000 C
      D(41) = 168.000 C
      D(42) = 169.000 C
      D(43) = 170.000 C
      D(44) = 171.000 C
      D(45) = 172.000 C
      D(46) = 173.000 C
      D(47) = 174.000 C
      D(48) = 175.000 C
      D(49) = 176.000 C
      D(50) = 177.000 C
      D(51) = 178.000 C
      D(52) = 179.000 C
      D(53) = 180.000 C
      D(54) = 181.000 C
      D(55) = 182.000 C
      D(56) = 183.000 C
      D(57) = 184.000 C
      D(58) = 185.000 C
      D(59) = 186.000 C
      D(60) = 187.000 C
      D(61) = 188.000 C
      D(62) = 189.000 C
      D(63) = 190.000 C
      D(64) = 191.000 C
      D(65) = 192.000 C
      D(66) = 193.000 C
      D(67) = 194.000 C
      D(68) = 195.000 C
      D(69) = 196.000 C
      D(70) = 197.000 C
      D(71) = 198.000 C
      D(72) = 199.000 C
      D(73) = 200.000 C
      D(74) = 201.000 C
      D(75) = 202.000 C
      D(76) = 203.000 C
      D(77) = 204.000 C
      D(78) = 205.000 C
      D(79) = 206.000 C
      D(80) = 207.000 C
      D(81) = 208.000 C
      D(82) = 209.000 C
      D(83) = 210.000 C
      D(84) = 211.000 C
      D(85) = 212.000 C
      D(86) = 213.000 C
      D(87) = 214.000 C
      D(88) = 215.000 C
      D(89) = 216.000 C
      D(90) = 217.000 C
      D(91) = 218.000 C
      D(92) = 219.000 C
      D(93) = 220.000 C
      D(94) = 221.000 C
      D(95) = 222.000 C
      D(96) = 223.000 C
      D(97) = 224.000 C
      D(98) = 225.000 C
      D(99) = 226.000 C
      D(100) = 227.000 C
      D(101) = 228.000 C
      D(102) = 229.000 C
      D(103) = 230.000 C
      D(104) = 231.000 C
      D(105) = 232.000 C
      D(106) = 233.000 C
      D(107) = 234.000 C
      D(108) = 235.000 C
      D(109) = 236.000 C
      D(110) = 237.000 C
      D(111) = 238.000 C
      D(112) = 239.000 C
      D(113) = 240.000 C
      D(114) = 241.000 C
      D(115) = 242.000 C
      D(116) = 243.000 C
      D(117) = 244.000 C
      D(118) = 245.000 C
      D(119) = 246.000 C
      D(120) = 247.000 C
      D(121) = 248.000 C
      D(122) = 249.000 C
      D(123) = 250.000 C
      D(124) = 251.000 C
      D(125) = 252.000 C
      D(126) = 253.000 C
      D(127) = 254.000 C
      D(128) = 255.000 C
      D(129) = 256.000 C
      D(130) = 257.000 C
      D(131) = 258.000 C
      D(132) = 259.000 C
      D(133) = 260.000 C
      D(134) = 261.000 C
      D(135) = 262.000 C
      D(136) = 263.000 C
      D(137) = 264.000 C
      D(138) = 265.000 C
      D(139) = 266.000 C
      D(140) = 267.000 C
      D(141) = 268.000 C
      D(142) = 269.000 C
      D(143) = 270.000 C
      D(144) = 271.000 C
      D(145) = 272.000 C
      D(146) = 273.000 C
      D(147) = 274.000 C
      D(148) = 275.000 C
      D(149) = 276.000 C
      D(150) = 277.000 C
      D(151) = 278.000 C
      D(152) = 279.000 C
      D(153) = 280.000 C
      D(154) = 281.000 C
      D(155) = 282.000 C
      D(156) = 283.000 C
      D(157) = 284.000 C
      D(158) = 285.000 C
      D(159) = 286.000 C
      D(160) = 287.000 C
      D(161) = 288.000 C
      D(162) = 289.000 C
      D(163) = 290.000 C
      D(164) = 291.000 C
      D(165) = 292.000 C
      D(166) = 293.000 C
      D(167) = 294.000 C
      D(168) = 295.000 C
      D(169) = 296.000 C
      D(170) = 297.000 C
      D(171) = 298.000 C
      D(172) = 299.000 C
      D(173) = 300.000 C
      D(174) = 301.000 C
      D(175) = 302.000 C
      D(176) = 303.000 C
      D(177) = 304.000 C
      D(178) = 305.000 C
      D(179) = 306.000 C
      D(180) = 307.000 C
      D(181) = 308.000 C
      D(182) = 309.000 C
      D(183) = 310.000 C
      D(184) = 311.000 C
      D(185) = 312.000 C
      D(186) = 313.000 C
      D(187) = 314.000 C
      D(188) = 315.000 C
      D(189) = 316.000 C
      D(190) = 317.000 C
      D(191) = 318.000 C
      D(192) = 319.000 C
      D(193) = 320.000 C
      D(194) = 321.000 C
      D(195) = 322.000 C
      D(196) = 323.000 C
      D(197) = 324.000 C
      D(198) = 325.000 C
      D(199) = 326.000 C
      D(200) = 327.000 C
      D(201) = 328.000 C
      D(202) = 329.000 C
      D(203) = 330.000 C
      D(204) = 331.000 C
      D(205) = 332.000 C
      D(206) = 333.000 C
      D(207) = 334.000 C
      D(208) = 335.000 C
      D(209) = 336.000 C
      D(210) = 337.000 C
      D(211) = 338.000 C
      D(212) = 339.000 C
      D(213) = 340.000 C
      D(214) = 341.000 C
      D(215) = 342.000 C
      D(216) = 343.000 C
      D(217) = 344.000 C
      D(218) = 345.000 C
      D(219) = 346.000 C
      D(220) = 347.000 C
      D(221) = 348.000 C
      D(222) = 349.000 C
      D(223) = 350.000 C
      D(224) = 351.000 C
      D(225) = 352.000 C
      D(226) = 353.000 C
      D(227) = 354.000 C
      D(228) = 355.000 C
      D(229) = 356.000 C
      D(230) = 357.000 C
      D(231) = 358.000 C
      D(232) = 359.000 C
      D(233) = 360.000 C
      D(234) = 361.000 C
      D(235) = 362.000 C
      D(236) = 363.000 C
      D(237) = 364.000 C
      D(238) = 365.000 C
      D(239) = 366.000 C
      D(240) = 367.000 C
      D(241) = 368.000 C
      D(242) = 369.000 C
      D(243) = 370.000 C
      D(244) = 371.000 C
      D(245) = 372.000 C
      D(246) = 373.000 C
      D(247) = 374.000 C
      D(248) = 375.000 C
      D(249) = 376.000 C
      D(250) = 377.000 C
      D(251) = 378.000 C
      D(252) = 379.000 C
      D(253) = 380.000 C
      D(254) = 381.000 C
      D(255) = 382.000 C
      D(256) = 383.000 C
      D(257) = 384.000 C
      D(258) = 385.000 C
      D(259) = 386.000 C
      D(260) = 387.000 C
      D(261) = 388.000 C
      D(262) = 389.000 C
      D(263) = 390.000 C
      D(264) = 391.000 C
      D(265) = 392.000 C
      D(266) = 393.000 C
      D(267) = 394.000 C
      D(268) = 395.000 C
      D(269) = 396.000 C
      D(270) = 397.000 C
      D(271) = 398.000 C
      D(272) = 399.000 C
      D(273) = 400.000 C
      D(274) = 401.000 C
      D(275) = 402.000 C
      D(276) = 403.000 C
      D(277) = 404.000 C
      D(278) = 405.000 C
      D(279) = 406.000 C
      D(280) = 407.000 C
      D(281) = 408.000 C
      D(282) = 409.000 C
      D(283) = 410.000 C
      D(284) = 411.000 C
      D(285) = 412.000 C
      D(286) = 413.000 C
      D(287) = 414.000 C
      D(288) = 415.000 C
      D(289) = 416.000 C
      D(290) = 417.000 C
      D(291) = 418.000 C
      D(292) = 419.000 C
      D(293) = 420.000 C
      D(294) = 421.000 C
      D(295) = 422.000 C
      D(296) = 423.000 C
      D(297) = 424.000 C
      D(298) = 425.000 C
      D(299) = 426.000 C
      D(300) = 427.000 C
      D(301) = 428.000 C
      D(302) = 429.000 C
      D(303) = 430.000 C
      D(304) = 431.000 C
      D(305) = 432.000 C
      D(306) = 433.000 C
      D(307) = 434.000 C
      D(308) = 435.000 C
      D(309) = 436.000 C
      D(310) = 437.000 C
      D(311) = 438.000 C
      D(312) = 439.000 C
      D(313) = 440.000 C
      D(314) = 441.000 C
      D(315) = 442.000 C
      D(316) = 443.000 C
      D(317) = 444.000 C
      D(318) = 445.000 C
      D(319) = 446.000 C
      D(320) = 447.000 C
      D(321) = 448.000 C
      D(322) = 449.000 C
      D(323) = 450.000 C
      D(324) = 451.000 C
      D(325) = 452.000 C
      D(326) = 453.000 C
      D(327) = 454.000 C
      D(328) = 455.000 C
      D(329) = 456.000 C
      D(330) = 457.000 C
      D(331) = 458.000 C
      D(332) = 459.000 C
      D(333) = 460.000 C
      D(334) = 461.000 C
      D(335) = 462.000 C
      D(336) = 463.000 C
      D(337) = 464.000 C
      D(338) = 465.000 C
      D(339) = 466.000 C
      D(340) = 467.000 C
      D(341) = 468.000 C
      D(342) = 469.000 C
      D(343) = 470.000 C
      D(344) = 471.000 C
      D(345) = 472.000 C
      D(346) = 473.000 C
      D(347) = 474.000 C
      D(348) = 475.000 C
      D(349) = 476.000 C
      D(350) = 477.000 C
      D(351) = 478.000 C
      D(352) = 479.000 C
      D(353) = 480.000 C
      D(354) = 481.000 C
      D(355) = 482.000 C
      D(356) = 483.000 C
      D(357) = 484.000 C
      D(358) = 485.000 C
      D(359) = 486.000 C
      D(360) = 487.000 C
      D(361) = 488.000 C
      D(362) = 489.000 C
      D(363) = 490.000 C
      D(364) = 491.000 C
      D(365) = 492.000 C
      D(366) = 493.000 C
      D(367) = 494.000 C
      D(368) = 495.000 C
      D(369) = 496.000 C
      D(370) = 497.000 C
      D(371) = 498.000 C
      D(372) = 499.000 C
      D(373) = 500.000 C
      D(374) = 501.000 C
      D(375) = 502.000 C
      D(376) = 503.000 C
      D(377) = 504.000 C
      D(378) = 505.000 C
      D(379) = 506.000 C
      D(380) = 507.000 C
      D(381) = 508.000 C
      D(382) = 509.000 C
      D(383) = 510.000 C
      D(384) = 511.000 C
      D(385) = 512.000 C
      D(386) = 513.000 C
      D(387) = 514.000 C
      D(388) = 515.000 C
      D(389) = 516.000 C
      D(390) = 517.000 C
      D(391) = 518.000 C
      D(392) = 519.000 C
      D(393) = 520.000 C
      D(394) = 521.000 C
      D(395) = 522.000 C
      D(396) = 523.000 C
      D(397) = 524.000 C
      D(398) = 525.000 C
      D(399) = 526.000 C
      D(400) = 527.000 C
      D(401) = 528.000 C
      D(402) = 529.000 C
      D(403) = 530.000 C
      D(404) = 531.000 C
      D(405) = 532.000 C
      D(406) = 533.000 C
      D(407) = 534.000 C
      D(408) = 535.000 C
      D(409) = 536.000 C
      D(410) = 537.000 C
      D(411) = 538.000 C
      D(412) = 539.000 C
      D(413) = 540.000 C
      D(414) = 541.000 C
      D(415) = 542.000 C
      D(416) = 543.000 C
      D(417) = 544.000 C
      D(418) = 545.000 C
      D(419) = 546.000 C
      D(420) = 547.000 C
      D(421) = 548.000 C
      D(422) = 549.000 C
      D(423) = 550.000 C
      D(424) = 551.000 C
      D(425) = 552.000 C
      D(426) = 553.000 C
      D(427) = 554.000 C
      D(428) = 555.000 C
      D(429) = 556.000 C
      D(430) = 557.000 C
      D(431) = 558.000 C
      D(432) = 559.000 C
      D(433) = 560.000 C
      D(434) = 561.000 C
      D(435) = 562.000 C
      D(436) = 563.000 C
      D(437) = 564.000 C
      D(438) = 565.000 C
      D(439) = 566.000 C
      D(440) = 567.000 C
      D(441) = 568.000 C
      D(442) = 569.000 C
      D(443) = 570.000 C
      D(444) = 571.000 C
      D(445) = 572.000 C
      D(446) = 573.000 C
      D(447) = 574.000 C
      D(448) = 575.000 C
      D(449) = 576.000 C
      D(450) = 577.000 C
      D(451) = 578.000 C
      D(452) = 579.000 C
      D(453) = 580.000 C
      D(454) = 581.000 C
      D(455) = 582.000 C
      D(456) = 583.000 C
      D(457) = 584.000 C
      D(458) = 585.000 C
      D(459) = 586.000 C
      D(460) = 587.000 C
      D(461) = 588.000 C
      D(462) = 589.000 C
      D(463) = 590.000 C
      D(464) = 591.000 C
      D(465) = 592.000 C
      D(466) = 593.000 C
      D(467) = 594.000 C
      D(468) = 595.000 C
      D(469) = 596.000 C
      D(470) = 597.000 C
      D(471) = 598.000 C
      D(472) = 599.000 C
      D(473) = 600.000 C
      D(474) = 601.000 C
      D(475) = 602.000 C
      D(476) = 603.000 C
      D(477) = 604.000 C
      D(478) = 605.000 C
      D(479) = 606.000 C
      D(480) = 607.000 C
      D(481) = 608.000 C
      D(482) = 609.000 C
      D(483) = 610.000 C
      D(484) = 611.000 C
      D(485) = 612.000 C
      D(486) = 613.000 C
      D(487) = 614.000 C
      D(488) = 615.000 C
      D(489) = 616.000 C
      D(490) = 617.000 C
      D(491) = 618.000 C
      D(492) = 619.000 C
      D(493) = 620.000 C
      D(494) = 621.000 C
      D(495) = 622.000 C
      D(496) = 623.000 C
      D(497) = 624.000 C
      D(498) = 625.000 C
      D(499) = 626.000 C
      D(500) = 627.000 C
      D(501) = 628.000 C
      D(502) = 629.000 C
      D(503) = 630.000 C
      D(504) = 631.000 C
      D(505) = 632.000 C
      D(506) = 633.000 C
      D(507) = 634.000 C
      D(508) = 635.000 C
      D(509) = 636.000 C
      D(510) = 637.000 C
      D(511) = 638.000 C
      D(512) = 639.000 C
      D(513) = 640.000 C
      D(514) = 641.000 C
      D(515) = 642.000 C
      D(516) = 643.000 C
      D(517) = 644.000 C
      D(518) = 645.000 C
      D(519) = 646.000 C
      D(520) = 647.000 C
      D(521) = 648.000 C
      D(522) = 649.000 C
      D(523) = 650.000 C
      D(524) = 651.000 C
      D(525) = 652.000 C
      D(526) = 653.000 C
      D(527) = 654.000 C
      D(528) = 655.000 C
      D(529) = 656.000 C
      D(530) = 657.000 C
      D(531) = 658.000 C
      D(532) = 659.000 C
      D(533) = 660.000 C
      D(534) = 661.000 C
      D(535) = 662.000 C
      D(536) = 663.000 C
      D(537) = 664.000 C
      D(538) = 665.000 C
      D(539) = 666.000 C
      D(540) = 667.000 C
      D(541) = 668.000 C
      D(542) = 669.000 C
      D(543) = 670.000 C
      D(544) = 671.000 C
      D(545) = 672.000 C
      D(546) = 673.000 C
      D(547) = 674.000 C
      D(548) = 675.000 C
      D(549) = 676.000 C
      D(550) = 677.000 C
      D(551) = 678.000 C
      D(552) = 679.000 C
      D(553) = 680.000 C
      D(554) = 681.000 C
      D(555) = 682.000 C
      D(556) = 683.000 C
      D(557) = 684.000 C
      D(558) = 685.000 C
      D(559) = 686.000 C
      D(560) = 687.000 C
      D(561) = 688.000 C
      D(562) = 689.000 C
      D(563) = 690.000 C
      D(564) = 691.000 C
      D(565) = 692.000 C
      D(566) = 693.000 C
      D(567) = 694.000 C
      D(568) = 695.000 C
      D(569) = 696.000 C
      D(570) = 697.000 C
      D(571) = 698.000 C
      D(572) = 699.000 C
      D(573) = 700.000 C
      D(574) = 701.000 C
      D(575) = 702.000 C
      D(576) = 703.000 C
      D(577) = 704.000 C
      D(578) = 705.000 C
      D(579) = 706.000 C
      D(580) = 707.000 C
      D(581) = 708.000 C
      D(582) = 709.000 C
      D(583) = 710.000 C
      D(584) = 711.000 C
      D(585) = 712.000 C
      D(586) = 713.000 C
      D(587) = 714.000 C
      D(588) = 715.000 C
      D(589) = 716.000 C
      D(590) = 717.000 C
      D(591) = 718.000 C
      D(592) = 719.000 C
      D(593) = 720.000 C
      D(594) = 721.000 C
      D(595) = 722.000 C
      D(596) = 723.000 C
      D(597) = 724.000 C
      D(598) = 725.000 C
      D(599) = 726.000 C
      D(600) = 727.000 C
      D(601) = 728.000 C
      D(602) = 729.000 C
      D(603) = 730.000 C
      D(604) = 731.000 C
      D(605) = 732.000 C
      D
```

```

1259 = 1146.000 140 D0 150 K = 1, L
AR(I,J,K) = AR(I,J,K) / SCALE
1260 = 1147.000 AI(I,J,K) = AI(I,J,K) / SCALE
1261 = 1148.000 H = H + AR(I,J,K) * AI(I,J,K)
1262 = 1149.000 H = H + AR(I,J,K) * AI(I,J,K) * AI(I,J,K)
1263 = 1150.000 150 CONTINUE
1264 = 1151.000 C
E2(I) = SCALE * SCALE * H
G = SQRT(H)
1265 = 1152.000 E(I) = SCALE * G
1266 = 1153.000 F = CABS(CHPLX(AR(I,J),AI(I,L)))
1267 = 1154.000 F = CABS(CHPLX(AR(I,J),AI(I,L)))
1268 = 1155.000
1269 = 1156.000 C ***** FORM NEXT DIAGONAL ELEMENT OF MATRIX T *****
IF (F.EQ.0.0) G0 T0 160
1270 = 1157.000 TAU(1,L) = (AI(I,L) * TAU(2,I) + AR(I,L) * TAU(3,I)) / F
1271 = 1158.000 SI = (AR(I,L) * TAU(2,I) + AI(I,L) * TAU(3,I)) / F
1272 = 1159.000 H = H + F * G
1273 = 1160.000 G = 1.0 + G / F
1274 = 1161.000 AR(I,L) = G + AR(I,L)
1275 = 1162.000 AI(I,L) = G + AI(I,L)
1276 = 1163.000 IF (L.EQ.1) G0 T0 270
1277 = 1164.000 G0 T0 170
1278 = 1165.000 TAU(1,L) = TAU(1,I)
1279 = 1166.000 160 SI = TAU(2,I)
1280 = 1167.000 AR(I,L) = G
1281 = 1168.000 F = 0.0
1282 = 1169.000 170
1283 = 1170.000 C
D0 240 J = 1, L
1284 = 1171.000 G = 0.0
1285 = 1172.000 GI = 0.0
1286 = 1173.000 ***** FORM ELEMENT OF A+U *****
1287 = 1174.000 C ***** FORM ELEMENT OF A+U *****
D0 180 K = 1, J
1288 = 1175.000 G = G + AR(J,K) * AR(I,K) + AI(J,K) * AI(I,K)
1289 = 1176.000 G = G + AR(J,K) * AI(I,K) + AI(J,K) * AR(I,K)
1290 = 1177.000 GI = GI + AR(J,K) * AI(I,K) + AI(J,K) * AR(I,K)
1291 = 1178.000 180 CONTINUE
1292 = 1179.000 C
JP1 = J + 1
IF (L.LT.JP1) G0 T0 220
1293 = 1180.000
1294 = 1181.000
1295 = 1182.000 C
D0 200 K = JP1, L
1296 = 1183.000 G = G + AR(K,J) * AR(I,K) + AI(K,J) * AI(I,K)
1297 = 1184.000 G = G + AR(K,J) * AI(I,K) + AI(K,J) * AR(I,K)
1298 = 1185.000 GI = GI + AR(K,J) * AI(I,K) + AI(K,J) * AR(I,K)
1299 = 1186.000 200 CONTINUE
1300 = 1187.000 C ***** FORM ELEMENT OF P *****
E(I,J) = G / H
1301 = 1188.000 TAU(2,J) = GI / H
1302 = 1189.000 F = F + E(I,J) * AR(I,J) + TAU(2,J) * AI(I,J)
1303 = 1190.000 240 CONTINUE
1304 = 1191.000
1305 = 1192.000 C
HH = F / (H + H)
1306 = 1193.000 ***** FORM REDUCED A *****
D0 260 J = 1, L
1307 = 1194.000 C
F = AR(I,J)
1308 = 1195.000 G = E(I,J) - HH * F
1309 = 1196.000 E(I,J) = G
1310 = 1197.000 FI = AI(I,J)
1311 = 1198.000 FI = AI(I,J)
1312 = 1199.000 GI = TAU(2,J) - HH * FI
1313 = 1200.000 TAU(2,J) = GI
1314 = 1201.000
1315 = 1202.000 C
D0 260 K = 1, J
1316 = 1203.000 AR(I,J,K) = AR(I,J,K) + F * E(K) - G * AR(I,K)
1317 = 1204.000 AR(I,J,K) = AR(I,J,K) + FI * TAU(2,K) + GI * AI(I,K)
1318 = 1205.000 X

```

```

1319 = 1206.000 C
1320 = 1207.000 C
1321 = 1208.000 C
1322 = 1209.000 C
1323 = 1210.000 C
1324 = 1211.000 C
1325 = 1212.000 C
1326 = 1213.000 C
1327 = 1214.000 C
1328 = 1215.000 C
1329 = 1216.000 C
1330 = 1217.000 C
1331 = 1218.000 C
1332 = 1219.000 C
1333 = 1220.000 C
1334 = 1221.000 C
1335 = 1222.000 C
1336 = 1223.000 C
1337 = 1224.000 C
1338 = 1225.000 C
1339 = 1226.000 C
1340 = 1227.000 C
1341 = 1228.000 C
1342 = 1229.000 C
1343 = 1230.000 C
1344 = 1231.000 C
1345 = 1232.000 C
1346 = 1233.000 C
1347 = 1234.000 C
1348 = 1235.000 C
1349 = 1236.000 C
1350 = 1237.000 C
1351 = 1238.000 C
1352 = 1239.000 C
1353 = 1240.000 C
1354 = 1241.000 C
1355 = 1242.000 C
1356 = 1243.000 C
1357 = 1244.000 C
1358 = 1245.000 C
1359 = 1246.000 C
1360 = 1247.000 C
1361 = 1248.000 C
1362 = 1249.000 C
1363 = 1250.000 C
1364 = 1251.000 C
1365 = 1252.000 C
1366 = 1253.000 C
1367 = 1254.000 C
1368 = 1255.000 C
1369 = 1256.000 C
1370 = 1257.000 C
1371 = 1258.000 C
1372 = 1259.000 C
1373 = 1260.000 C
1374 = 1261.000 C
1375 = 1262.000 C
1376 = 1263.000 C
1377 = 1264.000 C
1378 = 1265.000 C

```

```

      AI(I,J,K) = AI(J,K) = F * TAU(2,K) = G * AI(I,K)
      = FI * L(K) = GI * AR(I,K)
      CONTINUE
      DO 280 K = 1, L
        AR(I,K) = SCALE * AR(I,K)
        AI(I,K) = SCALE * AI(I,K)
      CONTINUE
      TAU(2,L) = -SI
      HH = D(I)
      D(I) = AR(I,I)
      AR(I,I) = HH
      AI(I,I) = SCALE * H
      CONTINUE
      RETURN
      ***** LAST CARD OF HTRIDI *****
      END
      SUBROUTINE INTQ2(INM,N,D,E,Z,IERR)
      INTEGER I,J,K,L,M,N,II,NI,MPL,IERR
      REAL DIM1,Z(INM),Z(INM,NI)
      REAL B1,C,FAG,P,RI,S,MACHEP
      REAL SORT,ABS,SIGN
      THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE INTQ2,
      NUM. MATH. 12, 377-383(1968) BY MARTIN AND WILKINSON,
      AS MODIFIED IN NUM. MATH. 15, 450(1970) BY DUBRULLE.
      HANDBOOK FOR AUTO. COMP., VOL. II-LINEAR ALGEBRA, 241-248(1971).
      THIS SUBROUTINE FINDS THE EIGENVALUES AND EIGENVECTORS
      OF A SYMMETRIC TRIDIAGONAL MATRIX BY THE IMPLICIT QL METHOD.
      THE EIGENVECTORS OF A FULL SYMMETRIC MATRIX CAN ALSO
      BE FOUND IF TRED2 HAS BEEN USED TO REDUCE THIS
      FULL MATRIX TO TRIDIAGONAL FORM.
      ON INPUT=
      NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
      ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
      DIMENSION STATEMENT,
      N IS THE ORDER OF THE MATRIX,
      D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX,
      E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE INPUT MATRIX
      IN ITS LAST N-1 POSITIONS. E(1) IS ARBITRARY,
      Z CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE
      REDUCTION BY TRED2, IF PERFORMED, IF THE EIGENVECTORS
      OF THE TRIDIAGONAL MATRIX ARE DESIRED, Z MUST CONTAIN
      THE IDENTITY MATRIX.
      ON OUTPUT=
      D CONTAINS THE EIGENVALUES IN ASCENDING ORDER. IF AN
      ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT BUT
      UNORDERED FOR INDICES 1,2,...,AIERR-1,

```

```

1379 = 1260.000 C
1380 = 1267.000 C
1381 = 1268.000 C
1382 = 1269.000 C
1383 = 1270.000 C
1384 = 1271.000 C
1385 = 1272.000 C
1386 = 1273.000 C
1387 = 1274.000 C
1388 = 1275.000 C
1389 = 1276.000 C
1390 = 1277.000 C
1391 = 1278.000 C
1392 = 1279.000 C
1393 = 1280.000 C
1394 = 1281.000 C
1395 = 1282.000 C
1396 = 1283.000 C
1397 = 1284.000 C
1398 = 1285.000 C
1399 = 1286.000 C
1400 = 1287.000 C
1401 = 1288.000 C
1402 = 1289.000 C
1403 = 1290.000 C
1404 = 1291.000 C
1405 = 1292.000 C
1406 = 1293.000 C
1407 = 1294.000 C
1408 = 1295.000 C
1409 = 1296.000 C
1410 = 1297.000 C
1411 = 1298.000 C
1412 = 1299.000 C
1413 = 1300.000 C
1414 = 1301.000 C
1415 = 1302.000 C
1416 = 1303.000 C
1417 = 1304.000 C
1418 = 1305.000 C
1419 = 1306.000 C
1420 = 1307.000 C
1421 = 1308.000 C
1422 = 1309.000 C
1423 = 1310.000 C
1424 = 1311.000 C
1425 = 1312.000 C
1426 = 1313.000 C
1427 = 1314.000 C
1428 = 1315.000 C
1429 = 1316.000 C
1430 = 1317.000 C
1431 = 1318.000 C
1432 = 1319.000 C
1433 = 1320.000 C
1434 = 1321.000 C
1435 = 1322.000 C
1436 = 1323.000 C
1437 = 1324.000 C
1438 = 1325.000 C

E HAS BEEN DESTROYED,
Z CONTAINS ORTHONORMAL EIGENVECTORS OF THE SYMMETRIC
TRIANGULAR (OR FULL) MATRIX. IF AN ERROR EXIT IS MADE,
Z CONTAINS THE EIGENVECTORS ASSOCIATED WITH THE STORED
EIGENVALUES,
IERR IS SET TO
ZERO FOR NORMAL RETURNS
IF THE J-TH EIGENVALUE HAS NOT BEEN
DETERMINED AFTER 90 ITERATIONS.
QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
***** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
*****
MACHEP = 2.***E-7)
IERR = 0
IF (N * EQ. 1) GO TO 1001
DO 100 I = 2, N
E(I-1) = E(I)
E(N) = 0.0
DP 2#0 L = 1, N
J = 0
***** LOOK FOR SMALL SUB-DIAGONAL ELEMENT *****
DO 110 M = L, N
IF (M * EQ. N) GO TO 120
IF (ABS(E(M)) * LE. MACHEP * (ABS(D(M)) + ABS(D(M+1))))
GO TO 120
X
GO TO 120
CONTINUE
P = D(I)
IF (M * EQ. L) GO TO 2#0
IF (J * EQ. 30) GO TO 1000
J = J + 1
***** FORM SHIFT *****
G = (D(L) - P) / (2 * D * E(L))
R = SQR(T(G**2 + 1))
G = D(M) + P + E(L) / (G + SIGN(R,G))
S = 1 + 0
C = 1 + 0
P = 0 + 0
M#L = M - L
DO 200 II = 1, M#L
I = M - II
F = S * E(II)
B = C * E(II)
IF (ABS(F) * LT. ABS(G)) GO TO 150
C = G / F
F = G / F

```

```

1439 = 1326.000 R = SORTIC*(1.0)
1440 = 1327.000 E(I+1) = F * R
1441 = 1328.000 S = 1.0 / R
1442 = 1329.000 C = C * S
1443 = 1330.000 GO TO 160
1444 = 1331.000 S = F / G
1445 = 1332.000 R = SORT(S*S+1.0)
1446 = 1333.000 E(I+1) = G * R
1447 = 1334.000 C = 1.0 / R
1448 = 1335.000 S = S * C
1449 = 1336.000 G = D(I+1) = P
1450 = 1337.000 R = (D(I) = G) * S + 2.0 * C * B
1451 = 1338.000 P = S * R
1452 = 1339.000 D(I+1) = G * P
1453 = 1340.000 G = C * R = B
1454 = 1341.000 C ***** FORM VECTOR *****
1455 = 1342.000 DO 180 K = 1, N
1456 = 1343.000 F = Z(K,I+1)
1457 = 1344.000 Z(K,I+1) = S * Z(K,I) + C * F
1458 = 1345.000 Z(K,I) = C * Z(K,I) - S * F
1459 = 1346.000 CONTINUE
1460 = 1347.000 C
1461 = 1348.000 200 CONTINUE
1462 = 1349.000 C
1463 = 1350.000 D(I) = D(I) = P
1464 = 1351.000 E(L) = G
1465 = 1352.000 E(M) = 0.0
1466 = 1353.000 GO TO 105
1467 = 1354.000 2*0 CONTINUE
1468 = 1355.000 C ***** ORDER EIGENVALUES AND EIGENVECTORS *****
1469 = 1356.000 DO 300 II = 2, N
1470 = 1357.000 I = II - 1
1471 = 1358.000 K = I
1472 = 1359.000 P = D(II)
1473 = 1360.000 C
1474 = 1361.000 DO 260 J = II, N
1475 = 1362.000 IF (D(J) .GE. P) GO TO 260
1476 = 1363.000 K = J
1477 = 1364.000 P = D(J)
1478 = 1365.000 260 CONTINUE
1479 = 1366.000 C
1480 = 1367.000 IF (K .EQ. I) GO TO 300
1481 = 1368.000 D(K) = D(I)
1482 = 1369.000 D(I) = P
1483 = 1370.000 C
1484 = 1371.000 DO 280 J = 1, N
1485 = 1372.000 P = Z(J,I)
1486 = 1373.000 Z(J,I) = Z(J,K)
1487 = 1374.000 Z(J,K) = P
1488 = 1375.000 280 CONTINUE
1489 = 1376.000 C
1490 = 1377.000 300 CONTINUE
1491 = 1378.000 C
1492 = 1379.000 GO TO 1001
1493 = 1380.000 C
1494 = 1381.000 C ***** SET ERROR == NO CONVERGENCE TO AN
1495 = 1382.000 1000 IERR = L EIGENVALUE AFTER 30 ITERATIONS *****
1496 = 1383.000 1001 RETURN
1497 = 1384.000 C ***** LAST CARD OF INTGL2 *****
1498 = 1385.000 END

```

```

1499 = 1386+000 C
1500 = 1387+000 C
1501 = 1388+000 C
1502 = 1389+000 C
1503 = 1390+000 C
1504 = 1391+000 C
1505 = 1392+000 C
1506 = 1393+000 C
1507 = 1394+000 C
1508 = 1395+000 C
1509 = 1396+000 C
1510 = 1397+000 C
1511 = 1398+000 C
1512 = 1399+000 C
1513 = 1400+000 C
1514 = 1401+000 C
1515 = 1402+000 C
1516 = 1403+000 C
1517 = 1404+000 C
1518 = 1405+000 C
1519 = 1406+000 C
1520 = 1407+000 C
1521 = 1408+000 C
1522 = 1409+000 C
1523 = 1410+000 C
1524 = 1411+000 C
1525 = 1412+000 C
1526 = 1413+000 C
1527 = 1414+000 C
1528 = 1415+000 C
1529 = 1416+000 C
1530 = 1417+000 C
1531 = 1418+000 C
1532 = 1419+000 C
1533 = 1420+000 C
1534 = 1421+000 C
1535 = 1422+000 C
1536 = 1423+000 C
1537 = 1424+000 C
1538 = 1425+000 C
1539 = 1426+000 C
1540 = 1427+000 C
1541 = 1428+000 C
1542 = 1429+000 C
1543 = 1430+000 C
1544 = 1431+000 C
1545 = 1432+000 C
1546 = 1433+000 C
1547 = 1434+000 C
1548 = 1435+000 C
1549 = 1436+000 C
1550 = 1437+000 C
1551 = 1438+000 C
1552 = 1439+000 C
1553 = 1440+000 C
1554 = 1441+000 C
1555 = 1442+000 C
1556 = 1443+000 C
1557 = 1444+000 C
1558 = 1445+000 C

```

SUBROUTINE HTRIBK(NM,NAR,AI,TAU,M,ZR,ZI)  
INTEGER I,J,K,L,M,N,NM  
REAL AR(NM,NM),AI(NM,NM),TAU(2,NM),ZR(NM,NM),ZI(NM,NM)  
REAL M,N,S,I  
THIS SUBROUTINE IS A TRANSLATION OF A COMPLEX ANALOGUE OF  
THE ALGOL PROCEDURE TRBAK1, NUM. MATH. 11, 181-195 (1968)  
BY MARTIN, REINSCH, AND WILKINSON.  
HANDBOOK FOR AUTO. COMP., VOL. II - LINEAR ALGEBRA, 212-226 (1971).  
THIS SUBROUTINE FORMS THE EIGENVECTORS OF A COMPLEX HERMITIAN  
MATRIX BY BACK TRANSFORMING THOSE OF THE CORRESPONDING  
REAL SYMMETRIC TRIANGULAR MATRIX DETERMINED BY HTRIBA.  
ON INPUT =  
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL  
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM  
DIMENSION STATEMENT,  
N IS THE ORDER OF THE MATRIX,  
AR AND AI CONTAIN INFORMATION ABOUT THE UNITARY TRANS-  
FORMATIONS USED IN THE REDUCTION BY HTRIBI IN THE IR-  
FULL LOWER TRIANGLES EXCEPT FOR THE DIAGONAL OR AR.  
TAU CONTAINS FURTHER INFORMATION ABOUT THE TRANSFORMATIONS,  
M IS THE NUMBER OF EIGENVECTORS TO BE BACK TRANSFORMED,  
ZR CONTAINS THE EIGENVECTORS TO BE BACK TRANSFORMED  
IN ITS FIRST M COLUMNS.  
ON OUTPUT =  
ZR AND ZI CONTAIN THE REAL AND IMAGINARY PARTS,  
RESPECTIVELY, OF THE TRANSFORMED EIGENVECTORS  
IN THEIR FIRST M COLUMNS.  
NOTE THAT THE LAST COMPONENT OF EACH RETURNED VECTOR  
IS REAL AND THAT VECTOR EUCLIDEAN NORMS ARE PRESERVED.  
QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,  
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY  
\*\*\*\*\*  
\*\*\*\*\* TRANSFORM THE EIGENVECTORS OF THE REAL SYMMETRIC  
TRIANGULAR MATRIX TO THOSE OF THE HERMITIAN  
TRIANGULAR MATRIX. \*\*\*\*\*  
DO 50 J = 1, N  
DO 50 K = 1, M  
ZI(K,J) = ZR(K,J) \* TAU(2,K)  
ZR(K,J) = ZR(K,J) + TAU(1,K)  
50 CONTINUE  
IF (N.EQ. 1) GO TO 200  
\*\*\*\*\* RECOVER AND APPLY THE HOUSEHOLDER MATRICES \*\*\*\*\*  
\*\*\*\*\*

```

1559 = 1446.000      DO 140 I = 2, N
1560 = 1447.000      L = I - 1
1561 = 1448.000      H = AI(I,J)
1562 = 1449.000      IF (H.EQ.0.0) GO TO 140
1563 = 1450.000      C
1564 = 1451.000      DO 130 J = 1, M
1565 = 1452.000      S = 0.0
1566 = 1453.000      SI = 0.0
1567 = 1454.000      C
1568 = 1455.000      DO 110 K = 1, L
1569 = 1456.000      S = S + AR(I,K) * ZR(K,J) * AI(I,K) * ZI(K,J)
1570 = 1457.000      SI = SI + AR(I,K) * ZI(K,J) + AI(I,K) * ZR(K,J)
1571 = 1458.000      110 CONTINUE
1572 = 1459.000      C
1573 = 1460.000      S = S / H
1574 = 1461.000      SI = SI / H
1575 = 1462.000      C
1576 = 1463.000      DO 120 K = 1, L
1577 = 1464.000      ZR(K,J) = ZR(K,J) - S * AR(I,K) - SI * AI(I,K)
1578 = 1465.000      ZI(K,J) = ZI(K,J) - SI * AR(I,K) + S * AI(I,K)
1579 = 1466.000      120 CONTINUE
1580 = 1467.000      C
1581 = 1468.000      130 CONTINUE
1582 = 1469.000      C
1583 = 1470.000      140 CONTINUE
1584 = 1471.000      C
1585 = 1472.000      200 RETURN
1586 = 1473.000      C ***** LAST CARD OF HTRIBK *****
1587 = 1474.000      END
1588 = 1475.000      FUNCTION DEL(I,J)
1589 = 1476.000      DEL=0.0
1590 = 1477.000      IF (I.EQ.J) DEL=1.0
1591 = 1478.000      RETURN
1592 = 1479.000      END

```

```

85215064
85215065
85215066
85215067
85215068
85215069
85215070
85215071
85215072
85215073
85215074
85215075
85215076
85215077
85215078
85215079
85215080
85215081
85215082
85215083
85215084
85215085
85215086
85215087
85215088
85215089
85215090
85215091

```

## X. ELECTRONIC POLARON PROGRAM

### 1. Use of the Code

This is a self contained code and will produce the total correlation correction due to polarization for a fcc solid in a single pass for each point and band in question. If a different lattice structure is desired one must change routine PTS which defines the solid's Brillouin zone and modify the card in the main routine defining the solid's volume. The use of this code is simple and transparent and the data definition is quite simple and explicit.

X-2. INPUT DATA DEFINED

- Card 1: ITYM, unformatted\_time in minutes till checkpoint
- Card 2: 48 times 9I4 - gives cubic rotation matrices
- Card 3: 20A4 ZLAB - label to be printed at head of output
- Card 4: 20I4, NØRB - number of orbitals needed  
 NBND - number of bands  
 NØCC - number of occupied bands  
 IBAND(I), I=1 - tells which orbital each band refers to  
 next set DØ 30I = 1, NØRB
- Card 5: I4, 6X, 3F10.5  
 NCØEF - number of ST(Ø) in this orbital  
 UP(I) = average energy of the band for this orbital  
 UH(1,I) =  $\sigma$  hopping coef.  
 UH(2,I) =  $\pi$  hopping coef.
- next card DØ 31 J-1, NCØEF
- Card 6: AJ(J), ZETA(J), WT(J) from  $\psi(r) = \sum_{J=1}^{NCØEF} N WT(J) * r^{AJ(J)}$   
 $x \exp(-ZETZ(J) + r)$
- 31 continue
- 30 continue
- Card 7: 3F10.5  
 A = lattice constant  
 EEXC =  $\hbar W_{ex}$   
 EINF =  $\epsilon_{\infty}$

X-3. LIST OF PROGRAMS

12:00 APR 05, '76 ID= 302-H01  
 !JOB MRL2427,KUNZ,  
 LISTCC

!PCL

COPY CR TO KPLRMLN)

C IMPROVED ELECTRONIC POLARON PROGRAM OF A B KUNZ

C WRITTEN FEBRUARY 1976 FORTRAN IV

C MUST HAVE 1,3,4 BANDS OF VIRTUAL OR OCCUPIED AND 5 MUST LIE BELOW P

C ONLY IS AND/OR IP PER SET OF OCC OR VIRT LEVELS

COMMON ALP(10),GAP(16),DEL(4),KX(20),KY(20),KZ(20),SGN(

120),CBEF(20,4,4),E(20,4),ZETA(5),AJ(5),ZJ(5),VF(4),VH(2,4),ZLAB

Z(20),PST(2,16),K(16),WT(5),MAT(48,3,3),VF(4,20),NUM(3840),

JAM(4,4),TRVC(2,3),IBAND(4)

REAL\*4 KX,KY,KZ,NUM

INTEGER SGN,J

1 FORMAT (2,14)

2 FORMAT (8F10.5)

3 FORMAT (20A4)

4 FORMAT (10I,2,4)

5 FORMAT (14,6X,7F10.5)

6 FORMAT (1H, 'BAND KX = ',F6.3, ' KY = ',F6.3, ' KZ = ',F6.3, '

' BAND NUMBER ',I4, ' SELF ENERGY ',F10.3, ' RY ',F10.3, ' EV ',F10.3

2, ' HY ')

7 FORMAT (1H, 'ELECTRON BAND KX = ',F6.3, ' KY = ',F6.3, ' KZ = ',F6.3,

' BAND NUMBER ',I4, ' SELF ENERGY ',F10.3, ' RY ',F10.3, ' EV ',F10

2,3, ' HY ')

8 FORMAT (1H, ' THE NORMAL END')

INPUT ITYM

CALL CHKPT('PCL',ITYM)

TRVC(1,1)=TRVC(1,2)=TRVC(1,3)=1.0

TRVC(2,1)=-1.0; TRVC(2,2)=TRVC(2,3)=1.0

TRVC(3,1)=TRVC(3,3)=1.0; TRVC(3,2)=-1.0

TRVC(4,1)=TRVC(4,2)=1.0; TRVC(4,3)=-1.0

TRVC(5,1)=TRVC(5,2)=-1.0; TRVC(5,3)=1.0

TRVC(6,1)=TRVC(6,3)=-1.0; TRVC(6,2)=1.0

TRVC(7,1)=1.0; TRVC(7,2)=TRVC(7,3)=-1.0

TRVC(8,1)=TRVC(8,2)=TRVC(8,3)=-1.0

TRVC(9,1)=2.0; TRVC(9,2)=TRVC(9,3)=0.0

TRVC(10,1)=TRVC(10,3)=0.0; TRVC(10,2)=2.0

TRVC(11,1)=TRVC(11,2)=0.0; TRVC(11,3)=2.0

TRVC(12,1)=-2.0; TRVC(12,2)=TRVC(12,3)=0.0

TRVC(13,1)=TRVC(13,3)=0.0; TRVC(13,2)=-2.0

TRVC(14,1)=TRVC(14,2)=0.0; TRVC(14,3)=-2.0

TRVC(15,3)=TRVC(16,3)=TRVC(17,3)=TRVC(18,3)=TRVC(19,2)=TRVC(20,2)=

20.0

TRVC(15,1)=TRVC(15,2)=TRVC(16,2)=TRVC(17,1)=TRVC(19,1)=TRVC(19,3)=

22.0

TRVC(20,1)=TRVC(21,3)=TRVC(23,2)=TRVC(23,3)=TRVC(24,3)=TRVC(25,2)=

22.0

TRVC(16,1)=TRVC(17,2)=TRVC(18,1)=TRVC(18,2)=TRVC(20,3)=TRVC(21,1)=

22.0

TRVC(22,1)=TRVC(22,3)=TRVC(24,2)=TRVC(25,3)=TRVC(26,2)=TRVC(26,3)=

22.0

DO 29 I=1,48

29 READ 1, (MAT(I,J,K), J=1,3), K=1,3)

READ 3, ZLAB

PRINT 4, ZLAB

READ 1, NDRB, NBEF, NBCC, IBAND

RHO=-3.0; NBAND=NBND

H=0.0625

DO 34 I=1,161

K(I)=EXP(RHO)/I\*\*0

```

RHQ=RHJ+H
DO 34 J=1,NORB
34 PSI(J,I)=0.0
DO 30 I=1,NORB
READ 5,NCDEF,VP(I),VH(1,I),VH(2,I)
DO 31 J=1,NCDEF
READ 5,AJ(J),ZETA(J),WT(J)
31 ZJ(J)=((2.*ZETA(J))**(2*AJ(J)+3)/FACT(2*AJ(J)+2))**(1./2.)
DO 32 J=1,161
DO 32 K=1,NCDEF
32 PSI(I,J)=PSI(I,J)+WT(K)*ZJ(K)*(R(J)**(AJ(K)+1))*EXP(-ZETA(K)*R(J))
SUM=0.0
DO 33 J=1,160
33 SUM=SUM+0.5*(R(J+1)-R(J))*(PSI(I,J)**2+PSI(I,J+1)**2)
DO 30 J=1,161
30 PSI(I,J)=PSI(I,J)/SQRT(SUM)
CALL PTS(KX,KY,KZ,SGN,KTOT)
READ 2,A,EEXC,EINF
PI=3.1415926
PRF=2.*PI/A
DO 35 I=1,KTOT
XK=KX(I)**2+KY(I)**2+KZ(I)**2
XK=PRF*SQRT(XK)
DO 35 J=1,NORB
VF(J,I)=0.0
IF (I.EQ.1) VF(J,I)=1.0;GO TO 35
DO 36 K=1,160
36 VF(J,I)=VF(J,I)+0.5*(R(K+1)-R(K))*(PSI(J,K)*PSI(J,K)*SIN(XK*R(K))/
1(XK*R(K))+PSI(J,K+1)**2*SIN(XK*R(K+1))/(XK*R(K+1)))
35 CONTINUE
IS=0
DO 37 I=1,KTOT
37 IS=IS+SGN(I)
VBL=FLOAT(IS)*0.25*A*A*A
PREF=(2.*PI*EEXC/VPL*(1.-1./EINF))**(1./2.)*SQRT(2.*0)
DO 38 I=1,KTOT
XK=KX(I)**2+KY(I)**2+KZ(I)**2
XK=PRF*SQRT(XK)
DO 38 J=1,NORB
IF (I.EQ.1) VF(J,I)=0.0;GO TO 38
VF(J,I)=PREF/XK*VF(J,I)
38 CONTINUE
DO 39 I=1,NORB
39 VF(I,1)=PREF*A
C ORBITALS AND FOURIER COEFFICIENTS SET OBTAIN SCHEMF MATRIX
DO 40 I=1,KTOT
DO 41 J=1,10
41 ALP(J)=0.0
C SET ALP
ISTR=0
IEND=0
IPRVS=0
DO 43 J=1,NORB
IF (ABS(VH(2,J)).LT.0.0001) GO TO 50
GO TO 60
50 CONTINUE
C TYPE IS S
JJ=ISTR+1
IPRVS=IPRVS+J
ISTR=JJ+IPRVS
ALP(JJ)=VP(J)+G(VH(1,J),KX(I),KY(I),KZ(I))
GO TO 43
60 CONTINUE
C TYPE IS P
J1=ISTR+1
J2=ISTR+2+IPRVS

```

```

JJ=ISTR+4+2*IPRVS
JB=ISTR+3+IPRVS
Jb=ISTR+5+2*IPRVS
J9=ISTR+4+2*IPRVS
ISTR=J9
ALP(J1)=VF(J)+E*XX(VH(1,J),VH(2,J),KX(I),KY(I),KZ(I))
ALP(J2)=E*XY(VH(1,J),VH(2,J),KX(I),KY(I),KZ(I))
ALP(J3)=E*XY(VH(1,J),VH(2,J),KX(I),KZ(I),KY(I))
ALP(J5)=VF(J)+E*XX(VH(1,J),VH(2,J),KY(I),KX(I),KZ(I))
ALP(J6)=E*XY(VH(1,J),VH(2,J),KZ(I),KY(I),KX(I))
ALP(J9)=VF(J)+E*XX(VH(1,J),VH(2,J),KZ(I),KY(I),KX(I))
43 CONTINUE
CALL EIGEN(ALP,GAM,NBND,0)
II=0
DO 44 J=1,NBND
JTR=0
DO 2001 KXYZ=1,3
2001 JTR=JTR+KXYZ
E(I,NBND+1-J)=ALP(JTR)
DO 44 K=1,NBND
II=II+1
44 CDEF(I,NBND+1-J,K)=GAM(II)
40 CONTINUE
C SOLVE FOR SELF ENERGY OF EACH BAND FOR EACH POINT IN K-SPACE
DO 70 I=1,KTOT
DO 71 J=1,NBND
C SET NUMERATOR AND DENOMINATOR
IS=0
Ib=1;IBAS=1
IF=NBDC
IF (J.GT.NBDC) IB=NBDC+1;IF=NBND;IBAS=NBCC
IF (IB.GT.IF) GO TO 71
DO 72 K=1,KTOT
WTG=FLBAT(SGN(K))/FLBAT(48)
DO 73 L=1,48
CALL ZONE(KX,KY,KZ,K,L,0,0,0,0,0,0,XX,YY,ZZ,MAT)
C DENOMINATOR SET GET NUMERATOR
XK=KX(I)-XX;XP=ABS(XK)
YK=KY(I)-YY;YP=ABS(YK)
ZK=KZ(I)-ZZ;ZP=ABS(ZK)
XMG=XP+YP+ZP
IF (XMG.LE.1.5) .AND. XP.LE.1.0101 .AND. YP.LE.1.0101 .AND. ZP.LE.1.0
(LJ1) GO TO 81
C WE ARE MAPPED OUT OF ZONE GET BACK IN
DO 81 N=1,26
TX=XK-TRVC(N,1);PX=ABS(TX)
TY=YK-TRVC(N,2);PY=ABS(TY)
TZ=ZK-TRVC(N,3);PZ=ABS(TZ)
TTS=PX+PY+PZ
IF (TTS.LE.1.5) .AND. PX.LE.1.0101 .AND. PY.LE.1.0101 .AND. PZ.LE.1.0
(LJ1) XK=TX;YK=TY;ZK=TZ;GO TO 80
81 CONTINUE
STOP 'NEVER MAPPED BACK TO ZONE'
80 CONTINUE
C MAPPED BACK TO ZONE
XMG=XK*XX+YP*YY+ZK*ZK
C SEARCH FOR CORRECT V IN TABLE
DO 82 N=1,KTOT
IIN=N
AMJ=KX(N)*XX(I)+KY(N)*KY(I)+KZ(N)*KZ(I)
IF (XMG.LE.XMJ+.001 .AND. XMG.GE.XMJ-.001) GO TO 83
82 CONTINUE
83 CONTINUE
C ROTATE IF NEEDED
IF (L.GT.1) GO TO 90
DO 91 N=1,IF

```

```

      DO 91 I1=IB,IF
      KK=N-IBAS
      LL=I1-IBAS
91  AM(KK,LL)=COEF(K,N,I1)
      GO TO 100
90  CONTINUE
C    PERFORM ROTATION IF NEEDED
      IF (IB=IF.EQ.1) GO TO 92
      IF (IF=IB.EQ.2) GO TO 93
      GO TO 94
92  CONTINUE
C    ONLY S BAND
      AM(1,1)=COEF(K,IBAS+1,IBAS+1)
      GO TO 100
93  CONTINUE
C    ONLY P BAND
      DO 95 I1=1,3
      DO 95 I2=1,3
      AM(I1,I2)=0.0
      DO 96 I3=1,3
96  AM(I1,I2)=AM(I1,I2)+FLB*AT(MAT(L,I1,I3))*COEF(K,I2+IBAS,I3+IBAS)
95  CONTINUE
      GO TO 100
94  CONTINUE
C    S P BAND
      AM(1,1)=COEF(K,1+IBAS,1+IBAS)
      DO 97 I1=2,4
      DO 97 I2=2,4
      AM(I1,I2)=0.0
      DO 97 I3=2,4
97  AM(I1,I2)=AM(I1,I2)+FLB*AT(MAT(L,I1-1,I3-1))*COEF(K,I2+IBAS,I3+IBAS)
      AM(1,2)=AM(1,3)=AM(1,4)=AM(4,1)=AM(3,1)=AM(2,1)=0.0
C    ALL ROTATIONS ARE DONE
100 CONTINUE
      DO 74 N=IB,IF
      IS=IS+1
      NUM(IS)=0.0
      DO 101 N=IB,IF
101  NUM(IS)=NUM(IS)+VF*(JBAND(N),NN)*COEF(I,J,N)*AM(M-IBAS,N-IBAS)
      NUM(IS)=NUM(IS)*NUM(IS)*WTG
74  CONTINUE
73  CONTINUE
72  CONTINUE
C    WE CAN GET SELF ENERGY
      IF (J.GT.NBND) GO TO 300
      CALL LUPTB(IS,NUM,SELFE,SE1,SE2,I,J,EEXC,NBND,KTOT,E,IB,IF)
      PRINT 6,KX(I),KY(I),KZ(I),J,SELFE,SE1,SE2
      GO TO 400
300 CONTINUE
      CALL LUPTV(IS,NUM,SELFE,SE1,SE2,I,J,EEXC,NBND,KTOT,E,IB,IF)
      PRINT 7,KX(I),KY(I),KZ(I),J,SELFE,SE1,SE2
400 CONTINUE
71  CONTINUE
70  CONTINUE
      PRINT 8
      CALL EXIT
      END
      SUBROUTINE LUPTB(IS,NUM,SELFE,SE1,SE2,I,J,EEXC,NBND,KTOT,E,IB,IF)
1)
      DIMENSION NUM(1),E(20,4)
      REAL NUM

```

```

OCCUPIED LEVEL POSITIVE SELF ENERGY
DELTA=0.01
ESTR=0.01
F0FEN==ESTR
IC=0
DO 201 I=1,KTBT
DO 201 J=1,4B
DO 201 K=IB,IF
IC=IC+1
201 F0FEN=F0FEN+NUM(IC)/(ESTR+E(II,JJ)-E(I,K)+EEXC)
DO 202 I=1,200
F0FEO=F0FEN
ESTR=ESTR+DELTA
F0FEN==ESTR
IC=0
DO 203 J=1,KTBT
DO 203 K=1,4B
DO 203 L=IB,IF
IC=IC+1
203 F0FEN=F0FEN+NUM(IC)/(ESTR+E(II,JJ)-E(J,L)+EEXC)
IF (F0FEN/F0FEO.LE.0.0) SELFE=ESTR;GO TO 204
204 CONTINUE
204 SE1=13.6*SELFE
SE2=SELFE/2.0
RETURN
END
SUBROUTINE LUPTV (IS,NUM,SELFE,SE1,SE2,II,JJ,EEXC,NBND,KTBT,E,IB,IF
1)
DIMENSION NUM(1),E(20,4)
REAL NUM
C VIRTUAL LEVEL NEGATIVE SELF ENERGY
DELTA==.01
ESTR==.01
F0FEN==ESTR
IC=0
DO 301 I=1,KTBT
DO 301 J=1,4B
DO 301 K=IB,IF
IC=IC+1
301 F0FEN=F0FEN+NUM(IC)/(ESTR+E(II,JJ)-E(I,K)-EEXC)
DO 302 I=1,200
F0FEO=F0FEN
ESTR=ESTR+DELTA
F0FEN==ESTR
IC=0
DO 303 J=1,KTBT
DO 303 K=1,4B
DO 303 L=IB,IF
IC=IC+1
303 F0FEN=F0FEN+NUM(IC)/(ESTR+E(II,JJ)-E(J,L)-EEXC)
IF (F0FEN/F0FEO.LE.0.0) SELFE=ESTR;GO TO 304
304 CONTINUE
304 SE1=13.6*SELFE
SE2=SELFE/2.0
RETURN
END
SUBROUTINE PTS(KX,KY,KZ,SGN,II) 117
C THIS SUBROUTINE GENERATES THE BRILLUIN ZONE FOR A FCC SUBSTANCE 118
C WRITTEN BY A B KUNZ WINTER 1970 U OF ILLINOIS 119
DIMENSION KX(1),KY(1),KZ(1),SGN(1)
REAL KX,KY,KZ 121
INTEGER SGN 122
1 FORMAT (14) 123
2 FORMAT (45H11/48 BRILLUIN ZONE FOR A FCC SOLID A B KUNZ ) 124

```

3	FORMAT (27H(1)E X AXIS IS DIVIDED INTO, I6, 2X, 6HP(1)TS)	125
4	FORMAT (24H(1)E NUMBER OF POINTS IS, I6)	126
5	FORMAT (1X, 3F12.6, 13, I6)	127
6	FORMAT (16)	128
7	FORMAT (4F12.6)	129
	IWT=0.75	130
	HALF=0.5	131
	PRINT 2	132
	IPTS=5	
	PRINT 3, IPTS	134
	ITS=IPTS+1	
	ITS=ITS/2	
	IF (IPTS.LE.1) CALL EXIT	135
	XPTS=IPTS-1	136
	DEL=1.0/XPTS	137
	XI=-DEL	138
	YI=XI	139
	ZI=XI	140
	II=0	141
	X=XI	142
	DO 11 I=1, IPTS	143
	X=X+DEL	144
	Y=YI	145
	DO 12 J=1, I	146
	Y=Y+DEL	147
	Z=ZI	148
	DO 13 K=1, J	149
	Z=Z+DEL	150
	IF (X.LE.HALF) GO TO 14	151
	BOUND=1.5-X	152
	BOUND=1.5-X-Y	153
	IF (Y.LE.BOUND.AND.Z.LE.BOUND) GO TO 14	154
	GO TO 13	155
14	II=II+1	156
	KX(II)=X	157
	KY(II)=Y	158
	KZ(II)=Z	159
	SGN(II)=48	160
	IF (I.EQ.1) GO TO 301	161
	IF (I.EQ.IPTS.AND.J.EQ.1.AND.K.EQ.1) GO TO 306	
	IF (K.EQ.J.AND.X.EQ.I.AND.I.EQ.ITS) GO TO 307	
	IF (K.EQ.1.AND.J.EQ.1) GO TO 304	162
	IF (K.EQ.J.AND.K.EQ.I) GO TO 303	165
	IF (K.EQ.1.AND.J.EQ.I) GO TO 305	
	IF (I.EQ.IPTS.AND.K.EQ.J) GO TO 305	169
	IF (I.EQ.IPTS.AND.K.EQ.1) GO TO 305	168
	IF (J.EQ.1) GO TO 302	
	IF (K.EQ.J) GO TO 302	166
	IF (K.EQ.1) GO TO 302	
	IF (I.EQ.IPTS) GO TO 302	170
	GO TO 13	171
301	SGN(II)=1	172
	GO TO 13	173
302	SGN(II)=24	174
	GO TO 13	175
303	SGN(II)=8	176
	GO TO 13	177
304	SGN(II)=6	178
	GO TO 13	179
305	SGN(II)=12	180
	GO TO 13	
306	SGN(II)=3	

	GO TO 13	
307	SGN(II)=4	
13	CONTINUE	181
12	CONTINUE	182
11	CONTINUE	183
	PRINT 4, II	184
	RETURN	187
	END	188
	SUBROUTINE ZONE(KX,KY,KZ,I,J,X,Y,Z,XX,YY,ZZ,MAT)	
	DIMENSION KX(1),KY(1),KZ(1),SGN(1),MAT(48,3,3)	
	REAL KX,KY,KZ	
	XK=KX(1)	
	YK=KY(1)	
	ZK=KZ(1)	
	XX=XK*FL0AT(MAT(J,1,1))+YK*FL0AT(MAT(J,1,2))+ZK*FL0AT(MAT(J,1,3))	
	YY=XK*FL0AT(MAT(J,2,1))+YK*FL0AT(MAT(J,2,2))+ZK*FL0AT(MAT(J,2,3))	
	ZZ=XK*FL0AT(MAT(J,3,1))+YK*FL0AT(MAT(J,3,2))+ZK*FL0AT(MAT(J,3,3))	
	XX=XX-X	
	YY=YY-Y	
	ZZ=ZZ-Z	
	RETURN	
	END	
	FUNCTION FACT(I)	104
	J=1	105
	N=J	106
	IF (J.LE.0) GO TO 1	107
3	J=J-1	108
	IF (J.EQ.0) GO TO 2	109
	N=N*J	110
	GO TO 3	111
2	FACT=N	112
	RETURN	113
1	FACT=1.0	114
	RETURN	115
	END	116
	FUNCTION Q(X,A,P,C)	
	PI=3.14159	93
	Q= X*(COS(PI*A)*COS(PI*B)+COS(PI*A)*COS(PI*C)+COS(PI*B)*COS(PI*C))	94
	I**4.0	95
	RETURN	96
	END	97
	FUNCTION EXX(VS,VP,XX,YY,ZZ)	
	PI=3.1415926	
	X=PI*XX	
	Y=PI*YY	
	Z=PI*ZZ	
	EXX=2.*VS*(COS(X)*COS(Y)+COS(X)*COS(Z))	
	+2.*VP*(COS(X)*COS(Y)+COS(X)*COS(Z)+2.*COS(Y)*COS(Z))	
	RETURN	
	END	
	FUNCTION EXY(VS,VP,XX,YY,ZZ)	
	X=3.1415926*XX	
	Y=3.1415926*YY	
	EXY=(-2.)*VS*VP*(SIN(X)*SIN(Y))	
	RETURN	
	END	
	SUBROUTINE EIGEN(A,R,N,MV)	EIGE 410
	DIMENSION A(1),R(1)	EIGE 420
C	.....	EIGE 440
C	IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE	EIGE 460
C	C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION	EIGE 470
C	STATEMENT WHICH FOLLOWS.	EIGE 480
C	DOUBLE PRECISION A,P,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,	EIGE 500
C	COSX2,SINCS,RANGE	EIGE 510

C	DOUBLE PRECISION A,R,ANORM,ANRMAX,THR,X,Y,SINX,SINX2,COSX,	EIGE 500
C	1 CSX2,SINCS,RANGE	EIGE 510
C	THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS	EIGE 530
C	APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS	EIGE 540
C	ROUTINE.	EIGE 550
C	THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO	EIGE 570
C	CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. SQRT IN STATEMENTS	EIGE 580
C	40, 68, 75, AND 78 MUST BE CHANGED TO DSQRT. ABS IN STATEMENT	EIGE 590
C	62 MUST BE CHANGED TO DABS. THE CONSTANT IN STATEMENT 5 SHOULD	EIGE 600
C	BE CHANGED TO 1.00-12.	EIGE 610
C	.....	EIGE 630
C	5 RANGE=1.0E-6	
C	IF (MV=1) 10,25,10	EIGE 680
C	10 IG=N	EIGE 690
C	DO 20 J=1,N	EIGE 700
C	IG=IG+N	EIGE 710
C	DO 20 I=1,N	EIGE 720
C	IJ=IJ+1	EIGE 730
C	R(IJ)=0.0	EIGE 740
C	IF (I=J) 20,15,20	EIGE 750
C	15 R(IJ)=1.0	EIGE 760
C	20 CONTINUE	EIGE 770
C		EIGE 780
C	COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANRMAX)	EIGE 790
C		EIGE 800
C	25 ANORM=0.0	EIGE 810
C	DO 35 I=1,N	EIGE 820
C	DO 35 J=I,N	EIGE 830
C	IF (I=J) 30,35,3	EIGE 840
C	30 IA=I+(J+J-1)/2	EIGE 850
C	ANORM=ANORM+A(IA)*A(IA)	EIGE 860
C	35 CONTINUE	EIGE 870
C	IF (ANORM) 165,165,40	EIGE 880
C	40 ANORM=1.414* SQRT(ANORM)	
C	ANRMAX=ANORM*RANGE/FLBAT(N)	EIGE 900
C		EIGE 910
C	INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR	EIGE 920
C		EIGE 930
C	IND=0	EIGE 940
C	THR=ANORM	EIGE 950
C	45 THR=THR/FLBAT(N)	EIGE 960
C	50 L=1	EIGE 970
C	55 M=L+1	EIGE 980
C		EIGE 990
C	COMPUTE SIN AND COS	EIGE1000
C		EIGE1010
C	60 NG=(M*M-1)/2	EIGE1020
C	LG=(L*L-L)/2	EIGE1030
C	LM=L+NG	EIGE1040
C	62 IF (ABS(A(LM))-THR) 130,65,65	
C	65 IND=1	EIGE1060
C	LL=L+LG	EIGE1070
C	MM=M+MG	EIGE1080
C	X=0.5*(A(LL)+A(MM))	EIGE1090
C	68 Y=-A(LM)/ SQRT(A(LM)*A(LM)+X*X)	
C	IF (X) 70,75,75	EIGE1110
C	70 Y=-Y	EIGE1120
C	75 SINX=Y/ SQRT(2.0*(1.0+( SQRT(1.0+Y*Y))))	
C	SINX2=SINX*SINX	EIGE1140
C	76 COSX= SQRT(1.0-SINX2)	
C	COSX2=COSX*COSX	EIGE1160
C	SINCS =SINX*COSX	EIGE1170

C		EIGE1180
C	ROTATE L AND M COLUMNS	EIGE1190
C		EIGE1200
	ILQ=N*(L-1)	EIGE1210
	IMQ=N*(M-1)	EIGE1220
	DO 125 I=1,N	EIGE1230
	IQ=(I*I-1)/2	EIGE1240
	IF(I=L) 80,115,80	EIGE1250
80	IF(I=M) 85,115,90	EIGE1260
85	IM=I+MQ	EIGE1270
	GO TO 95	EIGE1280
90	IM=M+IQ	EIGE1290
95	IF(I=L) 100,105,105	EIGE1300
100	IL=I+LQ	EIGE1310
	GO TO 110	EIGE1320
105	IL=L+IQ	EIGE1330
110	X=A(IL)*COSX-A(IM)*SINX	EIGE1340
	A(IM)=A(IL)*SINX+A(IM)*COSX	EIGE1350
	A(IL)=X	EIGE1360
115	IF(MV=1) 120,125,120	EIGE1370
120	ILR=ILQ+I	EIGE1380
	IMR=IMQ+I	EIGE1390
	X=R(ILR)*COSX-R(IMR)*SINX	EIGE1400
	R(IMR)=R(ILR)*SINX+R(IMR)*COSX	EIGE1410
	R(ILR)=X	EIGE1420
125	CONTINUE	EIGE1430
	X=2.0*A(LM)*SINCS	EIGE1440
	Y=A(LL)*COSX2+A(MM)*SINX2-X	EIGE1450
	X=A(LL)*SINX2+A(MM)*COSX2+X	EIGE1460
	A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)	EIGE1470
	A(LL)=Y	EIGE1480
	A(MM)=X	EIGE1490
C		EIGE1500
C	TESTS FOR COMPLETION	EIGE1510
C		EIGE1520
C	TEST FOR M = LAST COLUMN	EIGE1530
C		EIGE1540
130	IF(M=N) 135,140,135	EIGE1550
135	M=M+1	EIGE1560
	GO TO 60	EIGE1570
C		EIGE1580
C	TEST FOR L = SECOND FROM LAST COLUMN	EIGE1590
C		EIGE1600
140	IF(L=(N-1)) 145,150,145	EIGE1610
145	L=L+1	EIGE1620
	GO TO 55	EIGE1630
150	IF(IND=1) 160,155,160	EIGE1640
155	IND=0	EIGE1650
	GO TO 50	EIGE1660
C		EIGE1670
C	COMPARE THRESHOLD WITH FINAL NORM	EIGE1680
C		EIGE1690
160	IF(THR=ANRMX) 165,165,45	EIGE1700
C		EIGE1710
C	SORT EIGENVALUES AND EIGENVECTORS	EIGE1720
C		EIGE1730
165	IQ==N	EIGE1740
	DO 185 I=1,N	EIGE1750
	IQ=IQ+N	EIGE1760
	LL=I+(I*I-1)/2	EIGE1770
	JQ=N*(I-2)	EIGE1780
	DO 185 J=I,N	EIGE1790
	JQ=JQ+N	EIGE1800
	MM=J+(J*J-J)/2	EIGE1810

```

IF(A(LL)=A(MM)) 170,185,185
170 X=A(LL)
A(LL)=A(MM)
A(MM)=X
IF(MV=1) 175,185,175
175 DO 180 K=1,N
ILR=IQ+K
IMR=JQ+K
X=R(ILR)
R(ILR)=R(IMR)
180 R(IMR)=X
185 CONTINUE
RETURN
END
!ASSIGN M:SI,(FILE,KPLRN)
!ASSIGN M:BO,(FILE,KPLRNA),(SAVE)
!FORTRAN LS,BO,RC
!LOAD (EF,(KPLRNA),(KCHKPT),(KXTRAS)),(MAP)
!RUN
!DATA
20
1 0 0 0 1 0 0 0 1
1 0 0 0 -1 0 0 0 -1
-1 0 0 0 1 0 0 0 -1
-1 0 0 0 -1 0 0 0 1
0 1 0 0 0 1 1 0 0
0 -1 0 0 0 1 -1 0 0
0 -1 0 0 0 -1 1 0 0
0 1 0 0 0 -1 -1 0 0
0 0 1 1 0 0 0 1 0
0 0 -1 -1 0 0 0 1 0
0 0 1 -1 0 0 0 -1 0
0 0 -1 1 0 0 0 -1 0
-1 0 0 0 0 1 0 -1 0
-1 0 0 0 0 -1 0 1 0
0 0 -1 0 -1 0 1 0 0
0 0 1 0 -1 0 0 0 0
0 1 0 -1 0 0 0 0 -1
0 -1 0 1 0 0 0 0 1
-1 0 0 0 -1 0 0 0 -1
-1 0 0 0 1 0 0 0 1
1 0 0 0 -1 0 0 0 1
1 0 0 0 1 0 0 0 -1
0 -1 0 0 0 -1 -1 0 0
0 1 0 0 0 -1 1 0 0
0 1 0 0 0 1 -1 0 0
0 -1 0 0 0 1 1 0 0
0 0 -1 -1 0 0 0 -1 0
0 0 1 1 0 0 0 -1 0
0 0 -1 1 0 0 0 1 0
0 0 1 -1 0 0 0 1 0
1 0 0 0 0 -1 0 1 0
1 0 0 0 0 1 0 -1 0
0 0 1 0 1 0 -1 0 0
0 0 -1 0 1 0 1 0 0
0 -1 0 1 0 0 0 0 1

```

0	1	0	-1	0	0	0	0	1
-1	0	0	0	0	-1	0	-1	0
-1	0	0	0	0	1	0	1	0
0	0	-1	0	-1	0	-1	0	0
0	0	1	0	-1	0	1	0	0
0	-1	0	-1	0	0	0	0	-1
0	1	0	1	0	0	0	0	-1

SI BANDS SPHERICAL APPROXIMATION

2	2	1	1	2
1		0.0		-0.1
0		0.5		1.0
1		1.0		0.06
0		0.5		1.0
10.20		0.65		12.0

### Electronic structure and optical properties of metallic calcium\*

Daniel J. Mickish,<sup>†</sup> A. Barry Kunz, and Sokrates T. Pantelides<sup>‡</sup>

Department of Physics and Materials Research Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

(Received 15 October 1973)

Self-consistent Hartree-Fock orbitals of calcium atoms in a crystalline environment were used in conjunction with the localized linear-combination-of-atomic-orbital method to obtain one-electron bands for metallic calcium. Correlation corrections are made by means of Overhauser's simplified method. The metal-to-semimetal-to-metal electronic transitions with increasing pressure are predicted to occur at smaller lattice spacings than predicted by calculations employing local exchange. A rigorous calculation of  $\epsilon_2$ , the imaginary part of complex dielectric function, made within the dipole and one-electron approximation predicts correctly the general shape and width of the 1s-valence emission spectra and the important structures superimposed on the broad 1s-conduction absorption spectra. There is good agreement with the specific-heat data, but the details of the Fermi surface as described by de Haas-van Alphen data have remained elusive—large percentage errors occur when the predicted areas are compared with the experimental results. The de Haas-van Alphen orbit areas are extremely sensitive to the accuracy of the calculations due to the flatness of the bands near the Fermi level.

#### I. INTRODUCTION

The purpose of this paper is to present rigorous Hartree-Fock energy bands for a real metal. The metal calcium has been chosen in order to exploit its atomic closed-shell properties and to use its metal to semimetal electronic transition properties observed under increasing pressure as a test of the accuracy of Hartree-Fock bands. Additional tests are made using available optical and de Haas-van Alphen data.

Nearly all calculations to date made on metals have approximated the true Hartree-Fock nonlocal exchange through the use of the formalism of Hohenberg, Kohn, and Sham.<sup>1,2</sup> Their work establishes the existence of a universal functional of the local density of an interacting electron gas which is capable of providing the correlated ground-state energy of the system, provided that the ground state is nondegenerate. Gilbert<sup>3</sup> has recently shown that if the external potential is nonlocal then the energy is a unique functional of the density operator, not the local density. This is a severe restriction since it complicates the use of local exchange potentials in real crystals as a means of obtaining arbitrarily accurate solutions. If (nonlocal) ion-core external potentials are used, then local exchange potentials are incapable of describing the true exchange. If, however, (local) nuclear external potentials are used, then the exchange potentials proportional to the local density to the one-third power<sup>2,4-6</sup> are not accurate since they require that the external potential be slowly varying.

In order to achieve a higher degree of accuracy in calculations of the electronic structure of solids one is faced with two known choices. The first choice involves the retaining of the Hohenberg-Kohn<sup>1</sup> theorem and the search for a universal func-

tional of the electron density which is far from slowly varying. There is no way of determining this functional, *a priori*, other than by using a functional expansion starting from the approximate expression valid for a homogeneous electron gas at high densities. The expansion must be capable of describing the effects of the singular Coulomb potentials which occur in real materials. The second choice involves the use of the well-known functional

$$E[\psi] = \langle \psi | H | \psi \rangle / \langle \psi | \psi \rangle$$

and systematic approximations in the application of this functional. Although at first thought this approach may lead to elaborate and tedious calculations, we have found that through the use of powerful mathematical transformations such as those that exist in local orbitals theory<sup>7-14</sup> and the efficient use of the random-access capabilities of modern computers, the second choice of attack is fully competitive with the first in terms of computation time.

The results presented in this paper were obtained by first making a one-electron approximation to  $E[\psi]$  and minimizing the energy leading to self-consistent Hartree-Fock Bloch eigenfunctions and energies. Several groups<sup>7-22</sup> have developed mathematical and computational techniques which allow rather accurate determination of the Hartree-Fock bands for a variety of crystals. The agreement among the various Hartree-Fock bands obtained by the different techniques is very good in spite of the use of approximations such as muffin-tin potentials in some cases and the wide diversity of basis functions employed.<sup>15</sup> Crystalline Hartree-Fock Bloch functions may be used as zero-order wave functions to accurately calculate the correlation energy as in Kunz's<sup>23</sup> method based on Toy-

TABLE I. Exponents and coefficients of Slater-type basis functions (STO's) used to construct self-consistent Hartree-Fock local orbitals for the equilibrium lattice  $\phi_{at} = \sum_{\lambda} C_{at,\lambda} \chi_{\lambda}$ . Coefficients to be used for "virtual" orbitals are indicated by an asterisk.

$\lambda$	STO $\chi_{\lambda}$	Exponent	$C_{1s,\lambda}$	$C_{2s,\lambda}$	$C_{3s,\lambda}$	$C_{4s,\lambda}$	$C_{5s,\lambda}^*$	$C_{6s,\lambda}^*$	$C_{7s,\lambda}^*$
1	1s	20.0000	0.97737	0.28339	0.09749	0.00450	0.0	0.0	0.0
2	3s	22.0757	0.02629	0.00390	0.00016	0.01387	0.0	0.0	0.0
3	3s	15.1048	0.01113	-0.19536	-0.06793	-0.06203	0.0	0.0	0.0
4	3s	9.9852	-0.00192	0.61682	-0.24971	0.15126	0.0	0.0	0.0
5	3s	7.2167	0.00102	-0.27025	-0.19569	-0.22970	0.0	0.0	0.0
6	3s	3.7902	0.00036	-0.00467	-0.68237	0.41196	0.0	0.0	0.0
7	3s	2.5921	0.00014	0.00001	0.47653	-0.40514	0.0	0.0	0.0
8	4s	1.4168	-0.00003	-0.00012	0.00077	0.66866	0.0	0.0	0.0
9	4s	0.8587	0.00001	0.00005	0.00257	-1.40270	0.0	0.0	0.0
10	4s	1.3500	0.0	0.0	0.0	0.0	1.0	0.0	0.0
11	5s	1.5200	0.0	0.0	0.0	0.0	0.0	1.0	0.0
12	6s	1.6800	0.0	0.0	0.0	0.0	0.0	0.0	1.0
			$C_{2p,\lambda}$	$C_{3p,\lambda}$	$C_{4p,\lambda}^*$	$C_{5p,\lambda}^*$			
1	2p	10.0000	0.67883	0.22133	0.0	0.0			
2	4p	17.3718	0.01298	0.00284	0.0	0.0			
3	4p	10.4808	0.24110	0.07190	0.0	0.0			
4	4p	7.5803	0.15816	-0.00661	0.0	0.0			
5	4p	4.7717	0.00239	-0.46585	0.0	0.0			
6	4p	2.9791	0.00100	-0.55991	0.0	0.0			
7	4p	1.8414	-0.00024	-0.06887	0.0	0.0			
8	3p	1.2200	0.0	0.0	1.0	0.0			
9	4p	1.4000	0.0	0.0	0.0	1.0			
			$C_{3d,\lambda}^*$	$C_{4d,\lambda}^*$					
1	3d	1.2200	1.0	0.0					
2	4d	1.4000	0.0	1.0					

zawa's<sup>24</sup> electronic polaron. Semiclassical methods for determining the correlation corrections can also be implemented in the case of insulators. These include the Mott-Littleton<sup>25</sup> method and the screened-exchange-plus-Coulomb-hole method.<sup>26</sup> The resulting correlated bands closely match experiment for the cases which they have been calculated.<sup>16,26</sup> In the case of metals, a simplified theory of electron correlation by Lundquist and Overhauser<sup>27,28</sup> may be used to correct the Hartree-Fock bands. Of course, the inclusion of correlation corrections via the well-known techniques of multiconfiguration self-consistent-field<sup>29</sup> or configuration interaction<sup>30</sup> is a definite possibility, although no such calculation has been performed on a crystalline material. It is interesting to note that the only approximation to the exchange operator that does give eigenvalues close to the Hartree-Fock values is Liberman's<sup>31,32</sup> energy-dependent excitation Hamiltonian.

## II. CALCULATION

### A. Local orbitals, Bloch eigenfunctions, and bands

The local orbitals equation<sup>7-9,11</sup> used in this paper

$$(F_a + U_a - \rho U_a \rho) \phi_{at} = \epsilon_{at} \phi_{at} \quad (1)$$

has been evaluated self-consistently to first order in overlap as indicated by Kunz.<sup>13</sup> The operator  $F_a$  is that part of the Hartree-Fock operator which involves only center  $a$  and  $U_a$  is simply the remainder, which can be considered as the potential field produced by the environment.

The appropriateness of retaining terms to only first order in overlap has been rigorously studied in a recent work by Kunz.<sup>33</sup> It was found that the approximation was justified even in some covalently bonded systems and, therefore, we believe that it is certainly justified in the present case of the calcium system.

Basis functions used for the determination of local orbitals for calcium were obtained from an earlier very accurate calculation of the atom.<sup>34</sup> The atomic eigenfunctions were used to begin the iteration process. The local orbitals are given in Table I. Also appearing in the table are the additional orbitals which were included to allow description of the unoccupied conduction band levels. The equilibrium lattice constant  $a_0$  was taken as 10.54 bohr (5.58 Å).

The last iteration in the search for self-consistency involved the determination of Bloch eigenfunctions rather than a new set of local orbitals. The Bloch basis consisted of linear combinations

of self-consistent local orbitals and compact Slater-type orbitals (STO's). This method can be considered a complete generalization of the linear combination-of-atomic-orbitals (LCAO) method.<sup>15,22,35-37</sup> There are two significant advantages to using the LCAO method. First, the local orbitals completely span the occupied Hartree-Fock space (assuming no approximations were invoked) and so basis functions built as Bloch sums of the local orbitals will also span that space. Second, the resulting integrals over local orbitals are independent of the crystal momentum  $\vec{k}$ . Once the integrals are determined for diagonalizing  $F = F_A + U_A$  at any one point in the Brillouin zone, the same integrals can be used with only a change in multiplying phase factors to diagonalize  $F$  at any other point. This method has proved to be the most economical, since evaluation of the bands is required at a large number of points in the Brillouin zone in order to make adequate comparisons with experimental optical and de Haas-van Alphen data.

There are some interesting concepts involved in going from the closed-shell calcium atom to the "open-shell" calcium metal when using the local orbitals and LCAO techniques. We wish to first point out that the local orbitals equation [Eq. (1)] possesses the point group symmetry of the crystal ( $O_h$  in the present case) and therefore the eigenfunctions  $|\phi_{ai}\rangle$  will transform according to the irreducible representation of that point group. The self-consistent density matrix is determined by an expression constructed of local orbitals (LO):

$$\rho_{LO}(\vec{r}, \vec{r}') = \sum_{i,j} |\phi_{ai}(\vec{r})\rangle S_{ai,bj}^{-1} \langle \phi_{bj}(\vec{r}') |. \quad (2)$$

This form of the density matrix is completely equivalent<sup>7,8</sup> to the canonical Hartree-Fock matrix

$$\rho(\vec{r}, \vec{r}') = \sum_{n,\vec{k}} |\psi_{n,\vec{k}}(\vec{r})\rangle \langle \psi_{n,\vec{k}}(\vec{r}') |, \quad (3)$$

where the  $\psi$ 's are eigenfunctions of the Hartree-Fock operator. There does exist a somewhat pathological case where the restricted Hartree-Fock approximation fails to be adequate, causing the breakdown of the equivalence of Eqs. (2) and (3). This occurs when the free atomic system can be described as closed shell but the fragment (here the atom in a crystal field) cannot be described as a set of closed-shell local orbitals. In the case of calcium this situation would manifest itself by an apparent crossing of the free atomic-local orbital energy levels. The atomic  $3d$  level, for example, could split into  $\Gamma_{25'}$  and  $\Gamma_{12}$  levels with the  $\Gamma_{25'}$  crossing the  $4s-\Gamma_1$  level such that it would lie below the  $\Gamma_1$  level. When the local orbitals are filled according to the energy criteria, the last two  $4s$

electrons would be assigned to the lower  $\Gamma_{25'}$  degenerate level instead of the usual  $\Gamma_1$  level giving rise to an open shell and not treatable by means of restricted Hartree-Fock. Although we have not explicitly carried out an open-shell local-orbitals calculation on calcium, we believe that the very large (1.25 Ry) energy difference between  $\Gamma_1$  and  $\Gamma_{25'}$  obtained from the equilibrium bands (Fig. 1) supports the assumption that the local orbitals level  $\Gamma_{25'}$  does not lie below  $\Gamma_1$  and that the local-orbital system is therefore closed.

Second, we wish to point out that in the present case the environment  $U_A$  used in the local orbitals Eq. (1) is approximated by the first term of its expansion in cubic harmonics. The first term possesses spherical symmetry (the second  $Y_4^m$  symmetry) and so the solutions to the local-orbitals equation transform according to the irreducible representations of the three dimensional rotation group. The density matrix is now constructed of local orbitals  $|\phi_{ai}\rangle$  which have only pure  $s$  and pure  $p$  symmetry. We now indicate to what degree this "spherical approximation" to  $U_A$  causes Eq. (2) to differ from Eq. (3). Of special concern is the realization that some states having  $d$ -like or higher symmetry must be occupied since the  $4s$  and  $3d$  or higher atomic levels must merge to obtain metallic behavior in the crystal. The states near the Fermi level (e.g.,  $X_1, L_1, U_1, W_2, K_1$ ), however, possess very little  $d$  like or higher symmetries being composed of mostly  $s$ - and  $p$ -like symmetry. A qualitative measure of the number of  $d$ -like or higher states making up the occupied orbitals can be determined by the degree to which bands from the second Brillouin zone lie below the Fermi level. As seen in Fig. 1, there are few states from the second zone lying below the Fermi level.

We therefore maintain that although  $\rho_{LO}(\vec{r}, \vec{r}')$  only approximates  $\rho(\vec{r}, \vec{r}')$  due to the spherical approximation to  $U_A$ , the approximation is extremely good. In addition, the last operation of determining the Bloch eigenfunctions from  $\rho_{LO}(\vec{r}, \vec{r}')$  is effectively a final iteration toward self-consistency without spherical symmetrization of  $U_A$ . Any resulting  $\psi$ 's partially containing Bloch sums of  $d$ -like STO's are permitted to be occupied if energetically favorable. As a consequence bands describing metallic behavior are found (see Fig. 1,  $a/a_0 = 1.0$ ).

Two technical points are in order. Exponents of the STO's used to construct the additional Bloch basis for description of the conduction bands were chosen such that the contribution to any overlap matrix element over Bloch orbitals due to the inclusion of the sixth shell of atoms was less than 0.001. Numerical evaluation of the Clebsch-Gordan coefficients incurred was performed to high

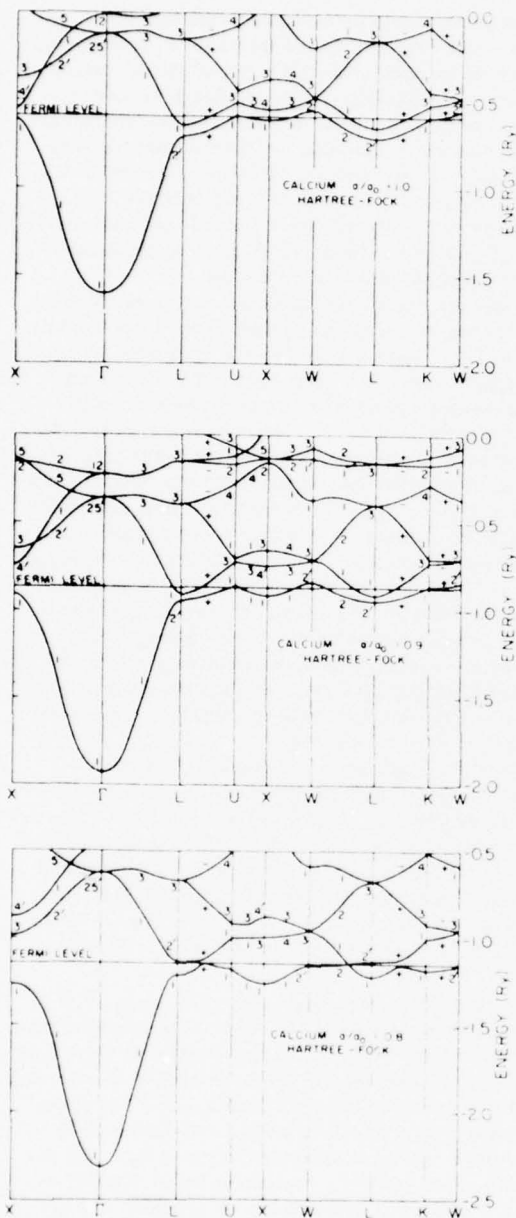


FIG. 1. Band structure of calcium for various lattice constants in the Hartree-Fock approximation ( $a_0 = 10.54$  bohr).

precision using a program written by Wills.<sup>36</sup>

The two-step process of first determining the Hartree-Fock orbitals self-consistently within the local orbitals formalism and then performing a final iteration to determine the canonical Hartree-Fock eigenfunctions was performed for lattice

spacings  $a$  ( $a/a_0 = 1.0, 0.9, 0.8, 0.7$ ). With this information we were able to make extensive comparisons with other calculations and with data from compression experiments. Due to near-linear dependence problems encountered for  $a/a_0 = 0.7$ , we were not able to obtain accurate bands for all  $\vec{k}$  at this lattice spacing. The Hartree-Fock bands for the larger three lattice constants are presented in Fig. 1.

### B. Overhauser correlation correction

Having obtained the Hartree-Fock band structure for metallic calcium, we proceeded to include polarization and correlation effects in the calculation. We choose to make these corrections using the model suggested by Lundquist<sup>27</sup> and Overhauser<sup>28</sup> due to its simplicity and its agreement with an earlier study<sup>39</sup> which concentrated on an electron gas in the high-density limit.

The fundamental approximation of the model is the replacement of the complete spectrum of excitation energies by a single-plasmon branch. The formalism is developed by first determining the interaction coefficient  $M_q$  for the coupling of a test charge to the plasmon of wave vector  $\vec{q}$ . The coefficient is determined by requiring fulfillment of the  $f$ -sum rule for an electron gas. The interaction Hamiltonian between the test charge at  $\vec{r}$  and the plasmon modes is found to be

$$H'(\vec{r}) = \sum_{\vec{q}} M_q (a_{\vec{q}} e^{i\vec{q}\cdot\vec{r}} + a_{\vec{q}}^\dagger e^{-i\vec{q}\cdot\vec{r}}), \quad (4)$$

with

$$M_q = \left( \frac{2\pi e^2 \hbar \omega_p^2}{q^2 \omega_q} \right)^{1/2} \quad (5)$$

and

$$\omega_p^2 = 4\pi N e^2 / m \Omega, \quad (6)$$

the square of the classical plasma frequency. The operators  $a_{\vec{q}}^\dagger$  and  $a_{\vec{q}}$  are plasmon creation and annihilation operators, respectively.

The plasmon frequency spectrum

$$\omega_q^2 = \omega_p^2 \epsilon_q / (\epsilon_q - 1) \quad (7)$$

is obtained by equating the electrostatic energy of interaction associated with each Fourier component as determined by classical field theory and that determined by second-order perturbation theory. The dielectric function  $\epsilon_q$  remains to be determined and is derived using perturbation theory on an electron-gas subject to a self-consistent potential. The change in the exchange and correlation potential due to the perturbation is simply found by assuming the local Kohn-Sham potential.<sup>2</sup> The generalized result is

$$\epsilon_q = 1 - Q(x) [1 - G(x)Q(x)], \quad (8)$$

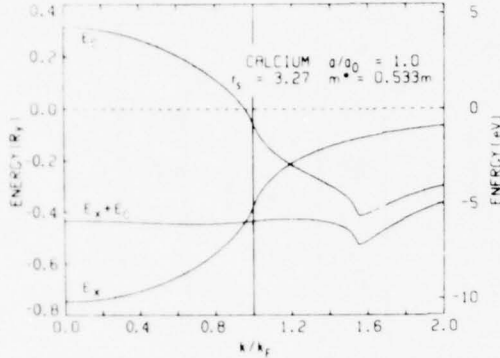


FIG. 2. Exchange energy  $E_x$  and correlation energy  $E_c$  and their sum for an electron gas with density of calcium and effective mass  $m^* = 0.533m$ .

with  $x = q/2k_F$ ,  $G(x) = x^2 g(x)$ ,  $Q(x) = Cf(x)/x^2$ ,  $C = me^2/\pi\hbar^2 k_F$ , and

$$f(x) = \frac{1}{2} + [(1-x^2)/4x] \ln |(1+x)/(1-x)|. \quad (9)$$

The limiting behavior of  $g(x)$  is determined by means of arguments involving compressibility relations (for  $x \rightarrow 0$ ) and pair correlation functions (for  $x \rightarrow \infty$ ). The behavior at intermediate  $x$  was determined by requiring the subsequently determined correlation energy to duplicate that calculated by a previous paper.<sup>40</sup> The function adopted for  $g(x)$  is

$$g(x) = 1.1/(1+10x^2+1.5x^4)^{1/2} \quad (10)$$

The equations reviewed so far have described the interaction of a fixed test charge with the plasmons. These equations are then extended to include the exchange and correlation potential associated with an electron. The extension simply requires that the interaction coefficient be replaced by

$$M_q' = M_q [1 - G(x)]. \quad (11)$$

Using second-order perturbation theory, the one-electron correlation energy  $E_c(\vec{k})$  is determined by taking the difference between the total system interaction energy with  $|\vec{k}\rangle$  occupied and with  $|\vec{k}\rangle$  empty:

$$E_c(\vec{k}) = E_c(k) = \sum_{|\vec{q}\rangle > k_F} \frac{M_q^2 [1 - G(x)]^2}{E(\vec{k}) - E(\vec{k} - \vec{q}) - \hbar\omega_q} - \sum_{|\vec{q}\rangle < k_F} \frac{M_q^2 [1 - G(x)]^2}{E(\vec{k} + \vec{q}) - E(\vec{k}) - \hbar\omega_q} \quad (12)$$

The one-electron energy  $E(k)$  for an interacting electron gas is the sum of the three contributions:

$$E(k) = \hbar^2/2m + E_x(k) + E_c(k), \quad (13)$$

where the exchange energy  $E_x(k)$  has the usual

form

$$E_x(\vec{k}) = E_x(k) = -(2e^2 k_F / \pi) f(y), \quad (14)$$

with  $f(y)$  given by Eq. (9) and  $y = k/k_F$ .

Overhauser shows that the infinite slope at  $k = k_F$  appearing in  $E_x(k)$  is exactly canceled by a similar property in  $E_c(k)$ . In addition, he shows that the sum of these two terms is essentially constant for all values of  $k$ , and that  $E_c(k)$  is relatively small when  $k \approx k_F$ . This implies that a Hartree-Fock calculation which treats the point-charge lattice correctly will give substantially correct one-electron energies for  $k \approx k_F$ , since the correlation corrections are minimal there. The most severe drawback of using this model to include correlation corrections to the bands is its  $\vec{k}$ -direction independence.

Correlation corrections made to the  $a/a_0 = 1.0$  Hartree-Fock bands are indicated graphically in Fig. 2 by the curve marked  $E_c$ . Also included in Fig. 2 are curves for the free-electron exchange  $E_x$  and the sum of these two curves. These curves are analogous to those appearing in Fig. 2 of Ref. 28, but modified for calcium with an equivalent-sphere radius  $r_s = 3.27$  and effective mass  $m^* = 0.533m$ . The effective mass was determined by forcing the valence band width of the free-electron bands (including exchange) to match the width of the Hartree-Fock bands. The purpose in using  $m^*$  is to include in an elementary way the effect of the lattice in free-electron bands and the correlation energy.

The relationship between free-electron bands and Hartree-Fock bands ( $a/a_0 = 1.0$ ) with and without correlation corrections is shown in Fig. 3. In panel (A) the free electron bands for an electron gas with the density of calcium and  $m^* = 0.533m$  is shown. In panel (B) the exchange energy  $E_x$  of Fig. 2 has been added to the free-electron bands, and in panel (C) the correlation energy  $E_c$  of Fig. 2 has also been added. Hartree-Fock bands are presented in panel (D), and in panel (E) the correlation energy of Fig. 2 has been added to the Hartree-Fock bands. The effective mass  $m^*$  was chosen to force the width of the valence band of panel (B) to match that appearing in panel (D). Panel (C) represents the best calcium bands determined within the free-electron approximation and Overhauser's method of including correlation effects. Panel (E) represents our final Hartree-Fock bands corrected for electron correlation.

In order to determine the correlation correction appropriate for Hartree-Fock Bloch state with energy  $E(\vec{k}^{\text{HF}})$  and wave vector  $\vec{k}^{\text{HF}}$  we use the corresponding free-electron-plus-exchange wave-vector magnitude  $k$  determined by solving the equation

$$E^{\text{HF}}(\vec{k}^{\text{HF}}) = E(k) = (\hbar^2/2m^*)k^2 + E_x(k) \quad (15)$$

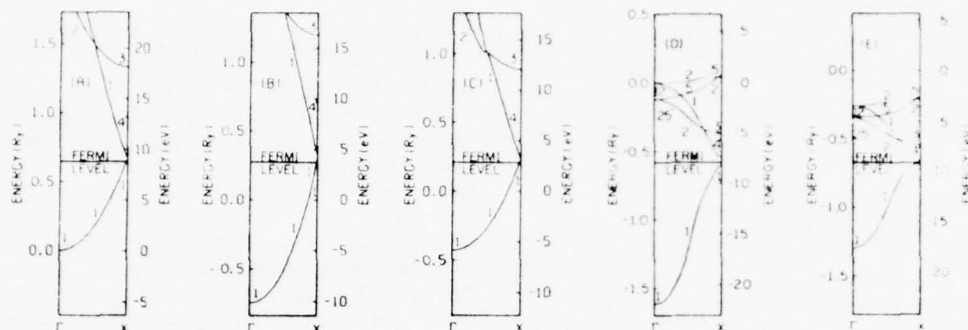


FIG. 3. The relationship between free-electron bands and Hartree-Fock bands ( $a/a_0 = 1.0$ ) with and without correlation corrections. Panel (A): free-exchange bands for an electron gas with the density of calcium and  $m^* = 0.533m$ . Panel (B): free electron bands with exchange energy  $E_x$ . Panel (C): free-electron bands with exchange-correlation energy  $E_x + E_c$ . Panel (D): Hartree-Fock bands. Panel (E): Hartree-Fock bands with correlation energy  $E_c$ .

for  $k$ . The correlation correction  $E_c(k)$  is calculated for this  $k$  and added to  $E(\vec{k}^{\text{HF}})$ . The correlation corrected bands for lattice spacings  $a$  ( $a/a_0 = 1.0, 0.9$ , and  $0.8$ ) are presented in Fig. 4.

In Fig. 5 we present for comparison, using the same scale, augmented-plane-wave (APW) bands for the equilibrium lattice spacing  $a/a_0 = 1.0$  obtained by McCaffrey, Anderson, and Papaconstantopoulos.<sup>41</sup> The Kohn-Sham<sup>2</sup> ( $\alpha = \frac{2}{3}$ ) local exchange was used. The band energies (Ry) at points of high symmetry for calcium for  $a/a_0 = 1.0$  are listed in Table II. Three cases are included: Hartree-Fock, Hartree-Fock with electron correlation corrections, and APW with a local exchange approximation ( $\alpha = \frac{2}{3}$ ).<sup>41</sup>

### C. Density of states

The bands are calculated at 89 uniformly distributed points  $\vec{k}_i$  within the irreducible wedge of the Brillouin zone. The density of states is calculated using the Lehmann and Taut<sup>42</sup> adaption of the Gilat and Raubenheimer<sup>43</sup> "linear analytic" (LA) method. Instead of the discrete nature of sampling that is usually employed in other methods,<sup>44</sup> analytical or continuous integration is performed throughout small tetrahedra whose vertices are defined by the points  $\vec{k}_i$ . The integration is accomplished by calculating numerically the area of a constant-energy plane surface that lies within each tetrahedron.

A recent analysis of methods for calculating spectral properties of solids has been made by Gilat<sup>44</sup> who finds: "Generally, the LA method is to be preferred in all cases where either high resolution and accuracy are desired or computational speed is needed, or both."

We express the density of states  $N(E)$  in units of states  $\text{Ry}^{-1} \text{atom}^{-1}$  and have multiplied by 2 in order to take spin into consideration:

$$N(E) = \frac{2v_c}{(2\pi)^3} \sum_i \oint \frac{dS}{|\nabla_{\vec{k}} E_i(\vec{k})|} \quad (16)$$

where  $v_c$  is the volume of the Wigner-Seitz cell. One need only integrate the density of states of the valence band over energy until the area equals 2 (in the case of divalent materials such as calcium) to find the Fermi level  $E_F$ . Due to the unique manner of obtaining self-consistency, the matching of hole states in the first zone with the electron states in the second zone is exact. The electronic specific heat  $C_e = \gamma T$  is determined by the density of states at  $E_F$  by means of the equation<sup>45</sup>

$$\gamma = \frac{1}{3} \pi^2 N(E_F) k_B^2 \quad (17)$$

or

$$\gamma \left[ \frac{\text{mJ}}{\text{mole} \cdot \text{K}^2} \right] = 0.17322 N(E_F) \left[ \frac{\text{states}}{\text{Ry atom}} \right] \quad (18)$$

Densities of states for the valence and conduction bands of calcium in the Hartree-Fock approximation for lattice spacings  $a$  ( $a/a_0 = 1.0, 0.9$ , and  $0.8$ ) are presented in Fig. 6. The densities of states corrected for electron correlation and for the same lattice spacings are presented in Fig. 7. In Fig. 8 we reproduce for comparison the density of states derived from the APW-local-exchange calculation.<sup>41</sup>

### D. Zero density of states at Fermi level

There is an interesting result<sup>46</sup> associated with the theory of a free-electron gas which is frequently referred to when discussing the inadequacies of the Hartree-Fock approximation. It is the rapid falloff to zero of the density of states as the Fermi level is approached. Zero density of states implies zero specific heat. Since this is not observed experimentally the conclusion is that the Hartree-Fock theory possesses serious inadequacies.

TABLE II. The energies (Ry) at points of high symmetry for calcium for  $a/a_0 = 1.0$ . Also included are the density of states at the Fermi surface  $N(E_F)$  [states  $\text{Ry}^{-1} \text{atom}^{-1}$ ] and the constant  $\gamma$  [ $\text{mJ K}^{-2} \text{mole}^{-1}$ ] entering the specific-heat expression. Column under "HF" gives Hartree-Fock results, "HF +  $E_c$ " gives Hartree-Fock results corrected for electron correlation by Overhauser's method and "Local  $E_x$ " gives the results obtained from an APW calculation using the Kohn-Sham local exchange approximation (Ref. 41).

	HF	HF + $E_c$	Local $E_x$
$\Gamma_1$	-1.6071	-1.2949	0.0290
$\Gamma_{25}'$	-0.1223	-0.3484	0.5590
$\Gamma_{12}$	-0.0131	-0.2578	0.6530
$X_1$	-0.6133	-0.6815	0.2902
$X_4'$	-0.5469	-0.6578	0.4020
$X_3$	-0.3728	-0.5455	0.4023
$L_2'$	-0.7084	-0.7106	0.3068
$L_1$	-0.6493	-0.6835	0.2643
$L_3$	-0.1526	-0.3742	0.7220
$W_2'$	-0.5503	-0.6601	0.3392
$W_3$	-0.4786	-0.6185	0.3977
$W_1$	-0.1989	-0.4114	0.5407
$K_1$	-0.5867	-0.6761	0.3265
$K_3$	-0.5116	-0.6384	0.4006
$K_1$	-0.4477	-0.5990	0.3788
$K_4$	-0.0708	-0.3054	0.6250
$E_F$	-0.587	-0.6750	0.3250
$N(E_F)$	4.19	14.19	11.20
$\gamma$	.73	2.46	1.94

cies.

There is a general argument<sup>47</sup> supporting the concept of finite energy differences between adjacent occupied and unoccupied levels and consequently a zero density of states at the Fermi level. It is that an electron in an occupied level is acted upon by an effective  $N - 1$  electron charge density, whereas an electron in a virtual orbital is acted upon by an  $N$  electron charge density. This occurs because the exchange term in the Hartree-Fock operator serves to effectively remove an electron from the total charge distribution acting on an occupied state as a means of preventing terms involving the self-interaction from being included in the energy expression. Although the electronic charge removed is just one in  $10^{23}$ , it is localized about the electron in the occupied state (the removed charge constitutes a "Fermi hole"). The degree of interaction of this "hole" with the electron and environment and consequently the local-

ized nature of the "hole" determines the size of the energy difference between the occupied and unoccupied levels. If the Fermi hole is completely delocalized, then the energy of interaction would simply be equal to the energy difference associated with environments composed of  $N$  and  $N - 1$  electrons, which is of the order of  $e^2 \Omega^{1/3} \approx 10^{-7} \text{ eV}$  for  $\Omega \approx 1 \text{ cm}^3$ . The hole is known to be nonspherical

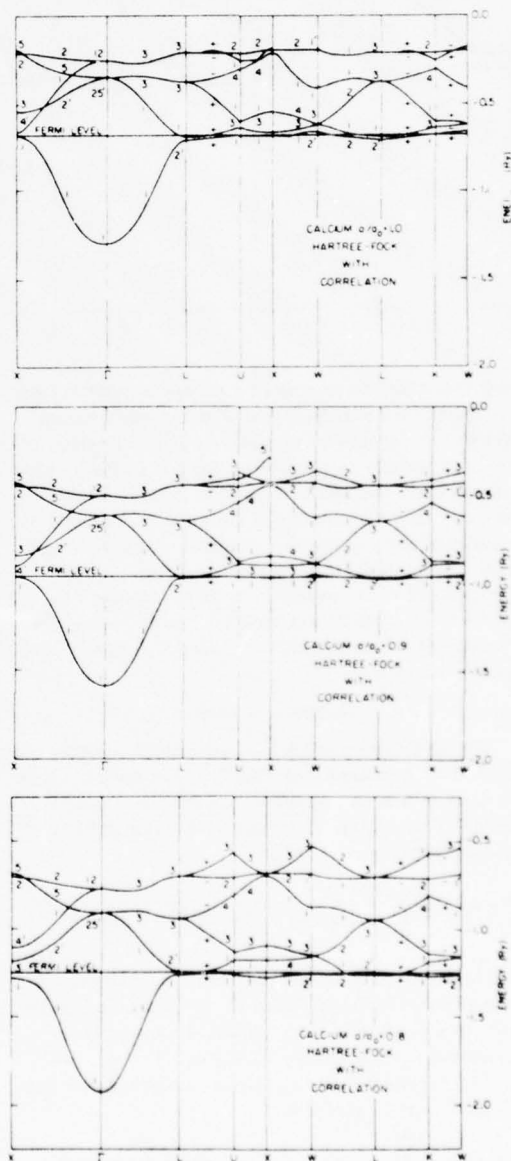


FIG. 4. Band structure of calcium for various lattice constants  $a$  in the Hartree-Fock approximation corrected for electron correlation effects.



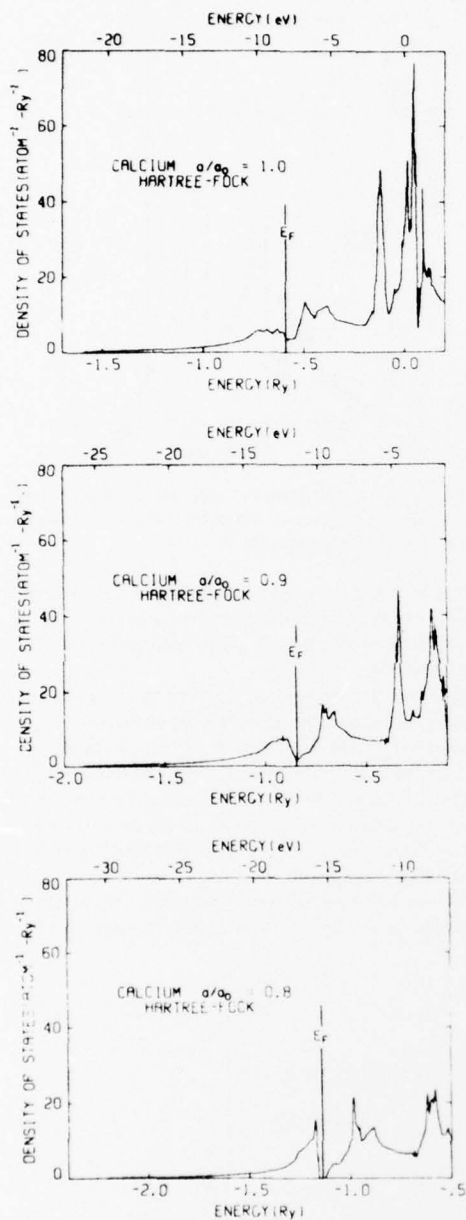


FIG. 6. Density of states of valence and conduction bands derived from the Hartree-Fock bands.

quite reasonable especially when it is realized that in most studies the momentum matrix elements are assumed constant. The oscillator strength for direct calcium 1s transitions used in the calculation of  $\epsilon_2$  is given in Fig. 10. The experimental 1s absorption and emission data are reproduced in Fig. 11 from the work of Finkelshteyn and Nemnonov.<sup>52</sup>

The broken line represents the calcium emission curve after corrections are made for the presence of CaO. The Fermi level is found from the intersection of the emission and absorption bands.

#### F. Fermi surface

The description of the shape of the Fermi surface has been one of the most difficult challenges of theoretical calculations on metals. Our attempt to predict de Haas-van Alphen orbit areas are summarized in Table III. We give results for both the uncorrelated and correlated Hartree-Fock bands. Also presented are the recent experimental results obtained by Jenkins and Datars<sup>53</sup> and the results obtained from an APW ( $\alpha = \frac{2}{3}$ ) calculation.<sup>41</sup> Percentage errors of the theoretical results from the experimental results are given in parentheses. In Fig. 12 we present the de Haas-van Alphen orbits corresponding to those listed in Table III. The solid lines indicate orbits obtained from uncorrelated Hartree-Fock bands and the broken lines indicate orbits obtained from correlated Hartree-Fock bands.

### III. DISCUSSION

#### A. Pressure-induced electronic transitions

One of the most interesting properties of calcium is its high-pressure-induced anomalies in electrical conductivity. Suggestions that these anomalies exist were first made by Altmann and Cracknell.<sup>54</sup> Almost simultaneously Stager and Drickamer<sup>55</sup> experimentally confirmed the predictions. The experimental resistance versus pressure results<sup>56</sup> are summarized in Fig. 13 for two temperatures. There is an increase in resistance with increasing pressure, as expected for a metal, at all pressures except in the region 300–400 kbar. In this region calcium acts as a semiconductor with respect to its temperature dependence on resistance. The phenomenon has been at least qualitatively explained by most calculations on calcium as being due to metal-to-semimetal-to-metal electronic transitions caused by increasing pressure. We refer the reader to the recent paper by McCaffrey, Anderson, and Papaconstantopoulos<sup>41</sup> for a review of both the experimental and theoretical works on calcium. We will use their work as a standard of previous work.

Both the Hartree-Fock bands (Fig. 1) and the Hartree-Fock bands with correlation corrections (Fig. 4) support the metal-to-semimetal-to-metal electronic transition explanation of the resistance anomaly. For  $a/a_0 = 1.0$  the Fermi surface cuts the bands in such a way that there is a significant density of states both above and below the Fermi level (Figs. 6 and 7). Calcium is therefore predicted to behave as a metal at equilibrium lattice

spacing. The 4s and 3d atomic levels merge to form a partially filled band that possesses essentially the same structure (except for valence band width) as in previous studies. It should be noted that, in contradiction with most earlier studies as in Fig. 5, the bands from the first and second

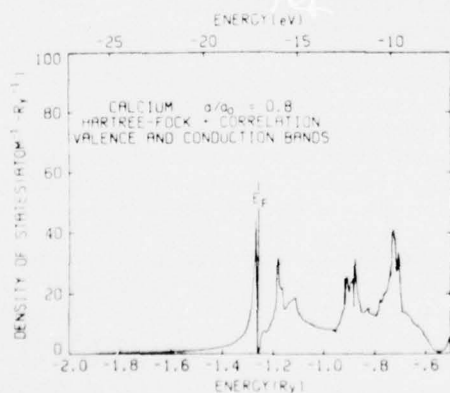
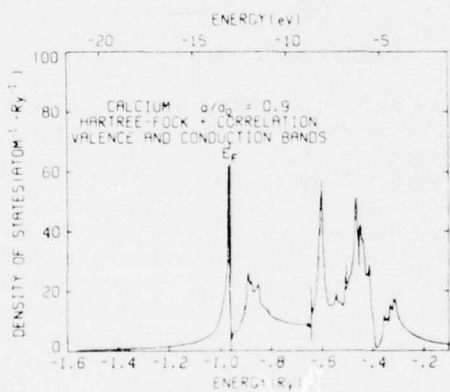
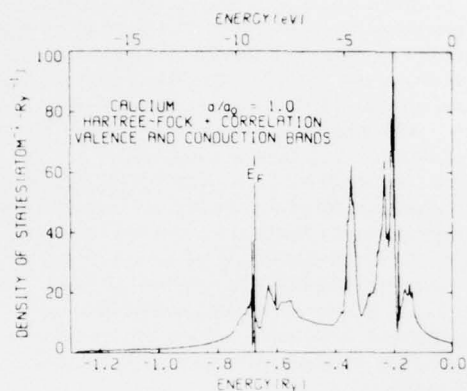


Fig. 7. Density of states of valence and conduction bands derived from the Hartree-Fock results corrected for electron correlation.

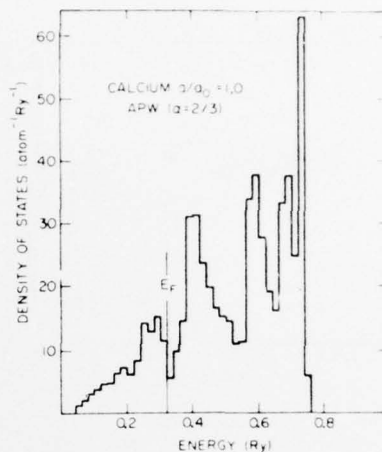


FIG. 8. Density of states of calcium for equilibrium lattice constant  $a_0$  obtained using the APW method and a local exchange approximation (Ref. 41).

Brillouin zones do not cross near the Fermi level. In fact, for the uncorrelated Hartree-Fock case, these bands show an unusual parallelism near the zone boundary.

When the lattice spacing is reduced to  $a/a_0 = 0.9$ , the parallelism in the Hartree-Fock bands (Fig. 1) is removed by the converging of the bands at  $L_1$  and  $L_2'$  near the Fermi level. Figure 4 shows the Hartree-Fock bands after correlation corrections have been made. The density of states surrounding the Fermi level has diminished greatly in both cases (Figs. 6 and 7).

The metal-to-semimetal electronic transition is predicted to be complete at a lattice spacing of

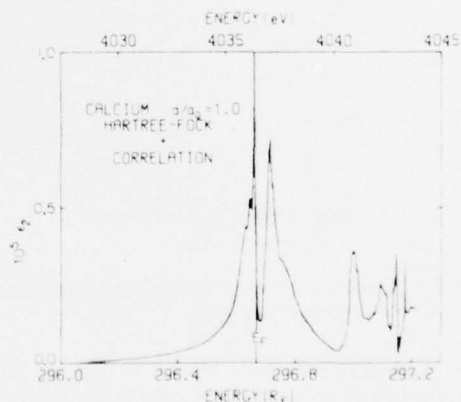


FIG. 9. Imaginary part  $\epsilon_2$  of the complex dielectric function for direct calcium 1s transitions. Correlation corrections for valence and conduction electrons have been made by Overhauser's method. The scale has been shifted so that experimental Fermi level is matched.

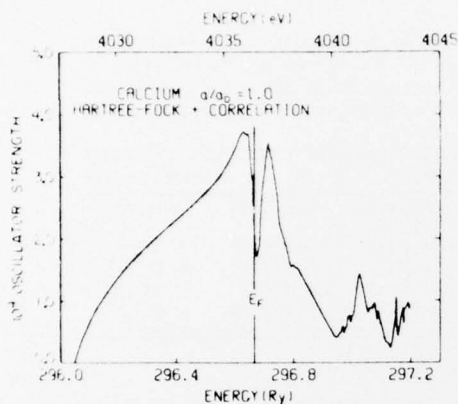


FIG. 10. Oscillator strength for direct calcium 1s transitions. Correlation corrections for valence and conduction electrons have been made by Overhauser's method. The scale has been shifted so that the experimental Fermi level is matched.

$a/a_0 = 0.8$ . In both the uncorrelated and correlated Hartree-Fock cases, the Fermi level cuts the bands at only a single point in the irreducible wedge. The density of states is accordingly infinitely small at the Fermi level. At this lattice spacing the bands from the first and second Brillouin zones have converged and crossed along the direction  $W-L$ . The surfaces of the two zones make contact at only this single point in the irreducible wedge and consequently the Fermi surface must pass through this point in order to match the electrons in the second zone with the number of holes in the first.

It is interesting to note that the uncorrelated and correlated Hartree-Fock density of states for  $a/a_0 = 0.8$  shows a definite range of energies (0.2 eV) for which the density is negligibly small. So, at a reduced lattice spacing of  $a/a_0 = 0.8$ , calcium possesses both the property of being a metal (the Fermi surface cuts the bands) and the property of a semiconductor (the existence of a "band gap" of 0.2 eV in the density of states about the Fermi level). The term "semimetal" is therefore appropriately applied to calcium at this lattice spacing. We find that our Hartree-Fock calculation correctly predicts semiconductor-like dependence of resistance on temperature at high pressures.

Although we were unable to obtain accurate bands for all  $\vec{k}$  with a lattice spacing of  $a/a_0 = 0.7$ , the trends indicate that the bands at  $X_3$  will continue to drop sufficiently so that  $X_3$  will lie below the Fermi level and the bands near the point of crossing along  $W-L$  will rise above the Fermi level. Since the Fermi surface will cut the bands at a large number of points, there will be a signifi-

cant density of states surrounding the Fermi level and calcium will return to a metal.

The lattice spacings at which the electronic transitions occur can be interpolated from a plot of the correlated Hartree-Fock energies  $X_3$ ,  $W'_2$ ,  $L'_2$ , and  $E_F$  as a function of  $a/a_0$  as in Fig. 14. The metal-to-semimetal transition occurs at approximately  $a/a_0 = 0.82$  when  $L'_2$  and  $W'_2$  converge to the same energy ( $E_F$ ). The semimetal-to-metal transition at approximately  $a/a_0 = 0.7$  may not be accurate due to linear-dependence problems mentioned earlier.

The first transition at  $a/a_0 = 0.82$  can be compared with the APW value<sup>41</sup> of 0.93 and the "experimental" value of 0.875 determined by Altmann, Harford, and Blake<sup>57</sup> using an equation of state. Our value of  $a/a_0 = 0.725$  for the second transition can be compared with the APW value of 0.8. The correlated Hartree-Fock bands predict electronic transitions at smaller lattice spacings than does the APW bands, but both are in about 6% error with the "experimental" value.

#### B. Optical properties

The optical spectra of crystalline solids are primarily dependent on the density of states of the bands involved in the transition. In some regions of the spectra transition, matrix elements can and do significantly enhance or depress peaks in the density of states and therefore cannot be ignored when calculating emission and absorption spectra. We rigorously include dipole transition

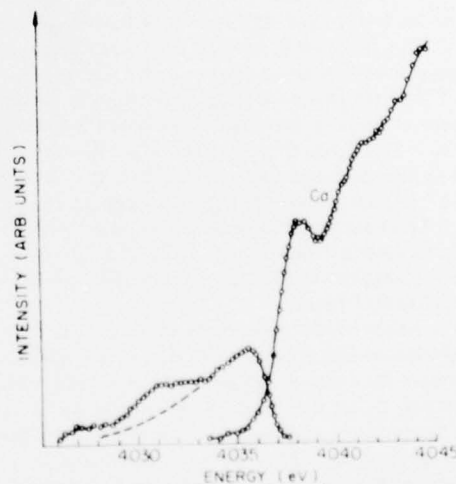


FIG. 11.  $K$  emission band (broken line) after elimination of the effects due to CaO contamination.  $K$  absorption band is shown at right. Intersection locates Fermi level. Vertical axis indicates absorption in arbitrary units. Reproduced from Ref. 52.

TABLE III. Experimental and theoretical de Haas-van Alphen data for  $a/a_0 = 1.0$ . Experimental results are from Jenkins and Datars (Ref. 53). Theoretical results are from our Hartree-Fock (HF), Hartree-Fock with correlation corrections (HF +  $E_c$ ) and from the McCaffrey, Anderson, and Papaconstantopoulos augmented-plane-wave (APW) (Ref. 41) calculations. Percentage errors from experiment are given in parentheses.

Orbit	$\vec{B}$	Experiment			HF	HF + $E_c$	APW ( $\alpha = 2/3$ )
		$P$ ( $10^{-7} \text{ G}^{-1}$ )	$\nu$ ( $10^4 \text{ G}$ )	$A[(2\pi/a)^2]$	$A[(2\pi/a)^2]$	$A[(2\pi/a)^2]$	$A[(2\pi/a)^2]$
$\alpha$	[100]	0.7843	1275	0.09584	0.043 (55%)	0.049 (49%)	0.12 (25%)
$\beta$	[110]	0.5814	1720	0.1293	0.10 (23%)	0.10 (23%)	0.05 (61%)
$\gamma$	[110]	3.077	325	0.12443	0.0032 (97%)	0.0 (100%)	0.009 (93%)
$\delta$	[110]	0.7874	1270	0.09546	0.089 (6.8%)	0.093 (2.6%)	0.15 (57%)

matrix elements in our calculation of  $\epsilon_2$ , the imaginary part of the complex dielectric function.

The experimental results (Fig. 11) show a 1s-valence emission band having a width exceeding 7 eV. The experimental 2p-valence emission band<sup>58</sup> is essentially identical in every respect. A 3p-valence emission band spectrum obtained by Kingston,<sup>59</sup> however, has a width of 3 eV. Since the 1s band is flat, the emission band width must be due to the valence band width, assuming electron-hole interactions and local field effects are negligible.<sup>50</sup> The Hartree-Fock valence band width for  $a/a_0 = 1.0$  (Fig. 1) exceeds 13 eV and thus is greater than experiment. This is consistent with the results obtained in the cases of the alkali halides,<sup>15</sup> the rare gas solids,<sup>15,16</sup> and beryllium.<sup>17</sup> However, when electron correlation corrections are made, good agreement with experiment is obtained.<sup>15</sup> This is also true in the case of calcium since the correlated Hartree-Fock bands for  $a/a_0 = 1.0$  (Fig. 4) show a valence band width of approximately 8 eV. This compares favorably with two<sup>52,58</sup> of the three existing experimental emission curves especially when one considers the extent of the low-energy tails on both the experimental data and on the calculation of  $\epsilon_2$  (Fig. 9). It is interesting to note that essentially all calculations employing a local exchange approximation (see, e.g., Fig. 5) yield a valence band width of about 3 eV which compares favorably with the one<sup>59</sup> remaining experimental result.

The shape of the valence-1s emission spectra as predicted by our calculated  $\epsilon_2$  curve is somewhat different than that experimentally observed. The peak just below the Fermi level is very close to the Fermi level and the tail toward lower energies is more skewed. The effect of the oscillator strength (Fig. 10) when included with the density of states (Fig. 7) to form  $\epsilon_2$  is seen to be a gradual, almost linear, enhancement of the spectrum from the low energy onset to within 0.5 eV of the Fermi level. From that point to the Fermi level, the oscillator strength falls abruptly. Although the transitions involved are from a 1s band to an s-

and/or d-like valence band, and the atomic selection rules would predict zero oscillator strength, there are a significant number of points in  $\vec{k}$  space away from  $\Gamma$  which do allow nonzero oscillator strengths.

Our agreement with the experimental absorption data is less apparent. However, there is a remarkable agreement for energies above the Fermi level between the correlated Hartree-Fock density of states (Fig. 7), that obtained from an APW calculation with a local exchange approximation (Fig. 8), and a recent Korrington-Kohn-Rostoker (KKR) calculation<sup>61</sup> also employing a local exchange approximation. The only significant differences are a smaller peak just above  $E_F$  and a greater distance between peaks of about 12% in the correlated Hartree-Fock case.

The calculation of  $\epsilon_2$  does not show the broad band exhibited in the experimental absorption spectrum. The oscillator strengths for this region of the spectrum are rather small and depress the peaks present in the density of states. There are two structures on the broad band, lying at about 4038.3 and 4041.7 eV. We associate the first and second peaks seen in our calculation of  $\epsilon_2$  with these structures since the absolute positions of the peaks are correct within 2 eV and the distance between them is correct within 0.5 eV.

There appear to be two possible interpretations of the origin of the broad band. The first is that our inclusion of only dipole matrix elements is a poor approximation, since large photon energies are involved and the approximation  $e^{i\vec{k}\cdot\vec{r}} \approx e^{i\vec{k}\cdot\vec{a}} \approx 1$  is not true ( $a$  is the radius of final l-el charge density;  $\vec{k}$  is the wavevector of photon). The transitions involved are from s to s bands or s to d bands and therefore have small dipole oscillator strength. Higher multipole transitions may be of considerable importance. The second interpretation of the origin of the broad band is simply that it is due to multiple band-to-band excitations, collective phonon excitations, or electron-hole interactions.

It is interesting to note a very recent study of the 1s absorption spectrum by McCaffrey and Papa-

constantopoulos<sup>62</sup> using their computed APW bands; the bands we have been using as a standard. Although their density of states agrees remarkably well with ours, the oscillator strength and optical-absorption spectrum significantly differ from ours.

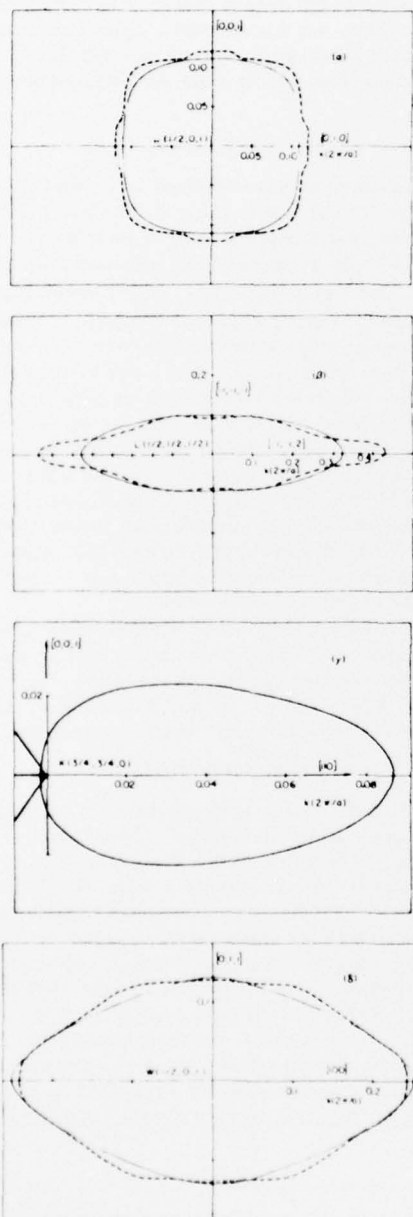


FIG. 12. de Haas-van Alphen  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  orbits of calcium for  $a/a_0 = 1.0$ . Solid lines indicate orbits obtained from Hartree-Fock bands, broken lines indicate orbits obtained from correlation corrected Hartree-Fock bands.

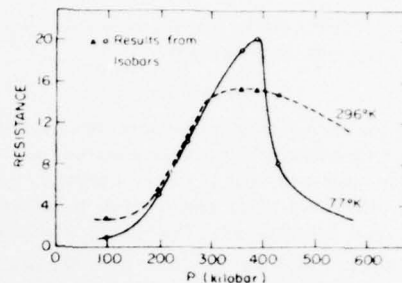


FIG. 13. Resistance versus pressure for calcium. Reproduced from Ref. 56.

This is probably due almost exclusively to the manner in which the dipole matrix elements are approximated in the APW case. The elements were determined by evaluating the matrix elements of the position vector  $\vec{r}$  rather than the matrix element of the gradient operator  $\vec{\nabla}$ . This would not normally have been an approximation except that the 1s Bloch eigenfunction was replaced by a 1s atomic orbital. The approximations also included integration only within the APW sphere.

The APW oscillator strength was found to be a very smooth function of energy and monotonically increasing in contrast with our results shown in Fig. 10. Notwithstanding the approximations, the APW was found to match experiment much better

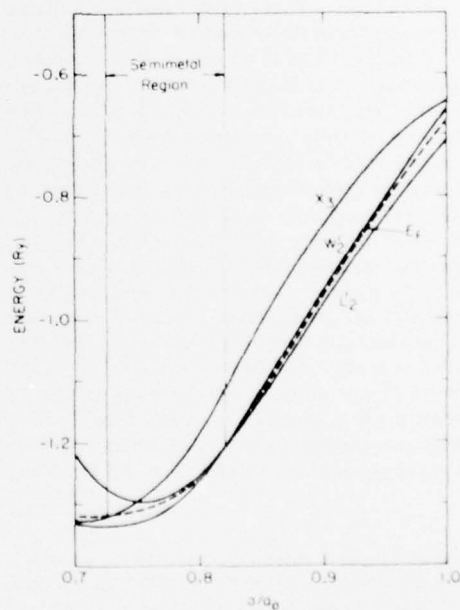


FIG. 14. Variation with lattice spacing of correlated Hartree-Fock energy bands at selected symmetry points.

than the present case in that the broad band on the high-energy side was obtained.

### C. Fermi surface

Since the correlation corrections determined by means of Overhauser's simplified method are small for electrons near the Fermi energy, there is little difference in the predicted de Haas-van Alphen orbit areas between the results predicted by the uncorrelated and the correlated Hartree-Fock bands (Table III and Fig. 12). Our results are no better than those obtained by means of the method employing local exchange approximations (see Table III), and so our Hartree-Fock calculations have failed in an area where accurate results have consistently remained elusive. The experimental data suggests that the Fermi surface in the first Brillouin zone is made up of interconnecting arms.<sup>63</sup> The Hartree-Fock results predict a connected Fermi surface, but the cross section at the point of connection (area of orbit  $\gamma$ ) is 97% too small. Making correlation corrections only serves to disconnect the Fermi surface (area of orbit  $\gamma$  becomes zero).

Our prediction of the constant  $\gamma$  [Eq. (17)] appearing in the specific-heat expression also has an error comparable to those obtained by means of local exchange approximations. As given in Table II we obtain a value of  $\gamma = 0.73$  ( $\text{mJ K}^{-2} \text{mole}^{-1}$ ) for the Hartree-Fock case and  $\gamma = 2.46$  ( $\text{mJ K}^{-2} \text{mole}^{-1}$ ) for the correlated Hartree-Fock case. These values can be compared with the experimentally obtained value of 2.9 ( $\text{mJ K}^{-2} \text{mole}^{-1}$ )<sup>64</sup> after inclusion of an enhancement factor<sup>65</sup> of 1.28 due to electron-phonon interactions. Our estimates of  $\gamma$  then become 0.93 and 3.15, respectively, with errors of 68% and 9%. We see in calculating this physical constant that the small correlation correction to electrons near the Fermi surface is significant.

It should be noted that the calculation of de Haas-van Alphen areas for calcium is very sensitive to the position of the Fermi level due to the flatness of the bands there. Although no qualitative estimates have been made, we observe that a slight lowering of the Fermi level (which was determined<sup>42</sup> with an error of 1%) would improve the predicted areas of orbits  $\alpha$ ,  $\gamma$  and  $\delta$  due to the resulting increase in size of those areas. However, the resulting area of the orbit  $\beta$  would be reduced

and the resulting value for the constant  $\gamma$  appearing in the specific-heat expression would be increased, both changes being in a direction opposite to that required for improvement. Obviously, the positioning of the Fermi level is not the sole contribution to the errors contained in our calculated de Haas-van Alphen data. Approximations such as the  $\vec{k}$ -independent electron-correlation corrections must surely make non-negligible contributions.

### V. CONCLUSIONS

In this paper we have obtained an *ab initio* band structure for calcium in the self-consistent Hartree-Fock approximation. Correlation corrections are made by means of Overhauser's simplified method. This method of calculation produces good results from an experimental point of view and employs no adjustable parameters. We are able to observe the metal-to-semimetal-to-metal electronic transition with increasing pressure, match the valence band width with the experimental emission spectra, and make preliminary identification of structures observed in the absorption spectra by means of a rigorous calculation of  $\epsilon_2$ . The details of the Fermi surface as indicated by de Haas-van Alphen data have remained elusive—we predict areas which have large percentage errors when compared to experiment.

We finally conclude that *ab initio* methods are a practical way of studying band structures in general. We are able to compute the bands at a large number of points in the Brillouin zone due to the efficiencies of local orbital and LCAO methods.

### ACKNOWLEDGMENTS

The authors wish to express their appreciation to Professor H. G. Drickamer, Professor B. Sonntag, Professor J. Dow, Professor F. C. Brown and Dr. D. L. Wilhite and Dr. D. F. Scofield for their useful discussions and criticisms relating to both the experimental and theoretical questions raised during the course of this study. We also express our appreciation to the staff of the MRL Xerox  $\Sigma$ -5 computing center and to R. F. Illyes, in particular, for their assistance. Finally, we wish to thank Dr. R. N. Euwema for his continued encouragement and Professor R. J. Maurer for his firm support of this research project.

\*Work supported in part by the Army Research Office under Contract No. DA-HCO4 69 C0007, by the National Science Foundation under Grant No. GH-33634 and by the Aerospace Research Laboratory, Air Force Systems Command, USAF, Wright-Patterson AFB, Ohio, Contract No. F33615-72-C-1506.

<sup>†</sup>Present address: Photo Products Department, Experimental Station Laboratory, E. I. du Pont de Nemours and Co., Wilmington, Del. 19898.

<sup>‡</sup>Present address: Department of Applied Physics, Stanford University, Stanford, Calif. 94305.

<sup>§</sup>P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864

- (1964).
- <sup>2</sup>W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
- <sup>3</sup>T. L. Gilbert (unpublished).
- <sup>4</sup>R. Gaspar, Acta Phys. **3**, 263 (1954).
- <sup>5</sup>J. C. Slater, Phys. Rev. **81**, 385 (1951).
- <sup>6</sup>J. C. Slater, in *Computational Methods in Band Theory*, edited by P. Marcus, J. Janak, and A. Williams (Plenum, New York, 1971), p. 447.
- <sup>7</sup>W. H. Adams, J. Chem. Phys. **34**, 89 (1961).
- <sup>8</sup>W. H. Adams, J. Chem. Phys. **37**, 2009 (1962).
- <sup>9</sup>T. L. Gilbert, *Molecular Orbitals in Chemistry, Physics and Biology*, edited by P. O. Lowdin and B. Pullman (Academic, New York, 1964), p. 405.
- <sup>10</sup>T. L. Gilbert, *Sigma Molecular Orbital Theory*, edited by O. Sinanoglu and K. B. Wiberg (Yale U. P., New Haven, Conn., 1970), p. 249.
- <sup>11</sup>P. W. Anderson, Phys. Rev. Lett. **21**, 13 (1968).
- <sup>12</sup>P. W. Anderson, Phys. Rev. **181**, 25 (1969).
- <sup>13</sup>A. B. Kunz, Phys. Status Solidi **36**, 301 (1969).
- <sup>14</sup>A. B. Kunz, Phys. Rev. B **8**, 1690 (1973).
- <sup>15</sup>D. J. Mickish and A. B. Kunz, J. Phys. C **6**, 1723 (1973).
- <sup>16</sup>A. B. Kunz and D. J. Mickish, Phys. Rev. B **8**, 779 (1973).
- <sup>17</sup>D. G. Shankland, Bul. Am. Phys. Soc. **11**, 387 (1966).
- <sup>18</sup>F. E. Harris, L. K. Kumar, and H. J. Monkhorst, Phys. Rev. B **7**, 2850 (1973).
- <sup>19</sup>L. Dagens and F. Perrot, Phys. Rev. B **8**, 1281 (1973).
- <sup>20</sup>N. O. Lipari and W. B. Fowler, Phys. Rev. B **2**, 3354 (1970).
- <sup>21</sup>L. Dagens and F. Perrot, Phys. Rev. B **5**, 641 (1972).
- <sup>22</sup>R. N. Euwema, D. L. Wilhite, and G. T. Surratt, Phys. Rev. B **7**, 818 (1973).
- <sup>23</sup>A. B. Kunz, Phys. Rev. B **6**, 606 (1972).
- <sup>24</sup>Y. Toyozawa, Prog. Theor. Phys. **12**, 421 (1954).
- <sup>25</sup>N. F. Mott and M. J. Littleton, Trans. Faraday Soc. **34**, 485 (1938).
- <sup>26</sup>W. B. Fowler, Phys. Rev. **151**, 657 (1966).
- <sup>27</sup>B. I. Lundquist, Phys. Kondens. Mater. **6**, 193 (1967).
- <sup>28</sup>A. W. Overhauser, Phys. Rev. B **3**, 1888 (1971).
- <sup>29</sup>T. L. Gilbert, Phys. Rev. A **6**, 580 (1972).
- <sup>30</sup>See articles in *Advances in Chemical Physics*, edited by R. Lefebvre and C. Moser (Interscience, New York, 1969), Vol. XIV.
- <sup>31</sup>D. A. Liberman, Phys. Rev. **171**, 1 (1968).
- <sup>32</sup>T. C. Collins, R. N. Euwema, D. J. Stuckel, and G. G. Wepler, Int. J. Quantum Chem. **S4**, 77 (1971).
- <sup>33</sup>A. B. Kunz, Phys. Rev. B **7**, 5369 (1973).
- <sup>34</sup>M. Synek, A. E. Raines, and C. C. J. Roothaan, Phys. Rev. **141**, 174 (1966).
- <sup>35</sup>J. C. Slater and G. F. Koster, Phys. Rev. **94**, 1498 (1954).
- <sup>36</sup>R. C. Chaney, T. K. Tung, C. C. Lin, and E. E. Lafon, J. Chem. Phys. **52**, 361 (1970).
- <sup>37</sup>E. E. Lafon and C. C. Lin, Phys. Rev. **152**, 579 (1966).
- <sup>38</sup>J. G. Wills, Oak Ridge National Laboratory Report No. ORNL-TM-1949 (1967) (unpublished).
- <sup>39</sup>J. Gell-Mann and K. A. Brueckner, Phys. Rev. **106**, 364 (1957).
- <sup>40</sup>K. S. Singwi, A. Sjölander, M. P. Tosi, and R. H. Land, Phys. Rev. B **1**, 1044 (1970).
- <sup>41</sup>J. W. McCaffrey, J. R. Anderson, and D. A. Papaconstantopoulos, Phys. Rev. B **7**, 674 (1973).
- <sup>42</sup>G. Lehmann and M. Taut, Phys. Status Solidi B **51**, 469 (1972).
- <sup>43</sup>G. Gilat and L. J. Raubenheimer, Phys. Rev. **141**, 390 (1966).
- <sup>44</sup>G. Gilat, J. Comp. Phys. **10**, 432 (1972).
- <sup>45</sup>See for example W. A. Harrison, *Solid State Theory*, (McGraw-Hill, New York, 1970), p. 239.
- <sup>46</sup>See, for example, C. Kittel, *Quantum Theory of Solids* (Wiley, New York, 1963), p. 86.
- <sup>47</sup>See for example S. Raimes, *The Wave Mechanics of Electron in Metals* (North-Holland, Amsterdam, 1967), p. 177.
- <sup>48</sup>We see no fall to zero of the density of states at  $k_F$  in either the Hartree-Fock case or the correlated case. But this is an obvious contradiction since  $\nabla_{\mathbf{k}}^2 E_c(\mathbf{k})$  possesses a singularity at  $k_F$ . If the density of states in the correlated case does not fall to zero, then there must exist a compensating singularity in the Hartree-Fock term  $\nabla_{\mathbf{k}}^2 E_{HF}(\mathbf{k})$  and the Hartree-Fock density must fall to zero at  $k_F$ . The contradiction is resolved by simply noting that the 89-point (LA)  $k$ -space integration technique we use is incapable of resolving the very weak singularity in  $\nabla_{\mathbf{k}}^2 E_c(\mathbf{k})$  or in  $\nabla_{\mathbf{k}}^2 E_{HF}(\mathbf{k})$ . The singularity in  $\nabla_{\mathbf{k}}^2 E_{HF}(\mathbf{k})$  is thus as weak as or weaker than the singularity in  $\nabla_{\mathbf{k}}^2 E_c(\mathbf{k})$ .
- <sup>49</sup>H. Ehrenreich and M. H. Cohen, Phys. Rev. **115**, 786 (1959).
- <sup>50</sup>M. Blume and R. E. Watson, Proc. Roy. Soc. A **270**, 127 (1962).
- <sup>51</sup>A. B. Kunz, D. J. Mickish, and T. C. Collins, Phys. Rev. Lett. **31**, 756 (1973).
- <sup>52</sup>L. D. Finkel'shteyn and S. A. Nemmonov, Phys. Met. Metallogr. **38**, 38 (1969).
- <sup>53</sup>R. M. Jenkins and W. R. Datars, Phys. Rev. B **7**, 2269 (1973).
- <sup>54</sup>S. L. Altmann and A. P. Cracknell, Proc. Phys. Soc. **84**, 761 (1964).
- <sup>55</sup>R. A. Stager and H. G. Drickamer, Phys. Rev. **131**, 2524 (1963).
- <sup>56</sup>H. G. Drickamer, Solid State Physics **17**, 1 (1965).
- <sup>57</sup>S. L. Altmann, A. R. Harford, and R. G. Blake, J. Phys. F **1**, 791 (1971).
- <sup>58</sup>H. W. Skinner, T. G. Bullen and J. E. Johnston, Phil. Mag. **45**, 1070 (1957).
- <sup>59</sup>R. H. Kingston, Phys. Rev. **81**, 944 (1951).
- <sup>60</sup>For a discussion of these problems, see, for example, P. Best, Phys. Rev. B **3**, 4377 (1971).
- <sup>61</sup>P. O. Nilsson, G. Arbman, and D. E. Eastman, Solid State Commun. **12**, 627 (1973).
- <sup>62</sup>J. W. McCaffrey and D. A. Papaconstantopoulos, Solid State Commun. **13**, 1055 (1974).
- <sup>63</sup>For a graphic illustration of the Fermi surface suggested by experimental de Haas-van Alphen data see Fig. 4 in Ref. 53.
- <sup>64</sup>L. M. Roberts, Proc. Phys. Soc. Lond. **B70**, 738 (1957).
- <sup>65</sup>A. O. E. Animalu and V. Heine, Philos. Mag. **12**, 1249 (1965).

Appendix 2 List of Students and Post Doctoral Personnel Supported  
by this Contract

Students Supported

Present Position

Peter W. Deutsch	U. S. Army Ballistics Research Laboratory
S.K.V. Mirmira	University of Illinois Chicago Circle
T. Shima	University of Illinois Urbana-Champaign
J. Marks	University of Illinois Urbana-Champaign

Post Doctoral Persons Supported

S. T. Pantelides	Research Physicist IBM Yorktown
V.E.M.J. Van Doren	Assistant Professor, University of Antwerp
K-L. Yip	Research Physicist Xerox Corp.
K-O. Almladh	University of Lund

Appendix 3 List of Publications Supported in Part by this Contract

1972

The role of the electronic polaron in the soft x-ray absorption of the lithium halides, with J. T. Devreese and T. C. Collins, Journal of Physics C, 5, 3259.

1973

Suggestion for improving the use of Gaussian lobe functions for atomic or molecular calculations, with P. W. Deutsch, Journal of Chemical Physics 58, 1779.

Systematic extensions of the local-orbitals method, Physical Review B, 7, 5369.

Energy bands in LiF and solid Ar, with D. J. Mickish, Journal of Physics C, 6, 1723.

Energy bands in LiH, with D. J. Mickish, Journal of Physics C, 6, L83.

Approximation to the method of local orbitals, Journal of Physics B, 6, L47.

Study of electronic structure and the optical properties of the solid rare gases, with D. J. Mickish, Physical Review B, 8, 779.

On the interaction of a hydrogen atom with a lithium metal surface, with D. J. Mickish and P. W. Deutsch, Solid State Communication 13, 35.

The absorption of soft x-rays by insulators with a forbidden exciton transition, with D. J. Mickish and T. C. Collins, Physical Review Letters 31, 756.

An excited state of alkali-halides, with T. C. Collins and J. T. Devreese, International Journal Quantum Chemical Symposium #7, 551.

1974

Distinction between optical absorption edges and photoemission edges in solids, Journal of Physics C, 7, L231-L234.

Development of a Non-local one-particle excitation Hamiltonian for atomic molecular, and solid state systems, with T. C. Collins, Journal of Physics B: Atomic Molecular Physics 7, L69-L72.

1974

- Model calculation for radiative recombination in Zn-N-doped  $\text{GaAs}_{1-x}\text{P}_x$  in the direct and indirect composition region, with J. C. Campbell, N. Holonyak, A. B. Kunz, and M. G. Craford, *Physical Review B*, 9, 4314-4322.
- Optical properties of LiF, with D. J. Mickish and T. C. Collins, *Physical Review B*, 9, 4461-4467.
- Effect of composition on "quasidirect" recombination and LED performance in the direct region of  $\text{GaAs}_{1-x}\text{P}_x$ :N, with J. C. Campbell, N. Holonyak, and M. G. Craford, *Applied Physical Letters* 25, 44-46.
- Ab initio study of the electronic properties of magnesium oxide, with S. T. Pantelides and D. J. Mickish, *Solid State Communication* 15, 203-205.
- Recombination processes involving Zn and N in  $\text{GaAs}_{1-x}\text{P}_x$ , with J. C. Campbell, N. Holonyak, and M. H. Lee, *Physical Review B*, 10, 1755-1757.
- Electronic structure and optical properties of metallic calcium, with D. J. Mickish and S. T. Pantelides, *Physical Review B*, 10, 1369-1383.
- Correlation effects in energy-band theory, with D. J. Mickish and S. T. Pantelides, *Physical Review B*, 10, 2602-2613.
- Approximate calculation of the electronic structure of solid surfaces and interfaces between material media, with V. E. Van Doren, *Physical Review B*, 10, 3578-3582.
- Ab initio energy band methods, in "Elementary Excitations in Solids, Molecules and Atoms," A, Plenum Press, 159-187.
- Soft x-ray spectra of molecules, In "Elementary Excitations in Solids, Molecules and Atoms," A, Plenum Press, 313-327.
- A review of the experimental properties of insulating solids, In "Elementary Excitations in Solids, Molecules and Atoms," B, Plenum Press, 343-344.
- Electronic structure and properties of MgO, with S. T. Pantelides, and D. J. Mickish, *Physical Review B*, 12, 5203-5212.
- Calculation of optical absorptive edges and photo-emission edges in solids, In. Proceedings of IV International Conference on Vacuum Ultraviolet Radiation Physics, Pergamon Press, 331-327.

1974

Index dispersion above the fundamental band edge in nitrogen-doped GaAs<sub>1-y</sub>P<sub>y</sub> (y = 0.38, E<sub>N</sub> > E<sub>Γ</sub>)\* with J. J. Coleman, N. Holonyak, M. J. Ludowise, M. Altarelli, W. O. Groves and D. L. Keune, Physical Review Letters 32, 1566-1564.

1975

Electronic structure of LiH and NaH, with D. J. Mickish, Physical Review B, 11, 1700-1704.

Influence of electron correlation on surface states of nonmetallic solids, Solid State Communication 16, 541-544.

Comments on the energy-band structure of solid-rare-gas mixtures, with T. C. Collins, D. Esterling, D. C. Licciardello, and D. J. Mickish, Physical Review B, 11, 3210-3213.

A comparison of experimental and theoretical photoelectron energy distribution curves for solid Ar and Kr, with D. J. Mickish, T. Shima, S.K.V. Mirmira, F. J. Hinpsel, V. Saile, N. Schwentuer, and E. E. Koch, Solid State Communication 17, 761-763.

Optical-absorption and photo-emission edges in insulating solids, Physical Review B, 12, 5890-5906.

Resonant Enhancement (?) of the recombination probability of the nitrogen-trap  $\Gamma$  - band edge crossover in GaAs<sub>1-x</sub>P<sub>x</sub>: N(E<sub>N</sub> = E<sub>Γ</sub> ; x ≡ x<sub>n</sub>) with J. J. Coleman, N. Holonyak, W. O. Groves, D. L. Keune, and M. G. Craford, Solid State Communication 16, 319-322.

## REFERENCES

1. Stephan, Guy, thesis, university of Rennes (unpublished, 1969).
2. Sonntag, B., Physical Review 9, 3601 (1974).
3. Gudat, W., Kunz, Ch., and Petersen, H., Physical Review Letters 32, 1370 (1974); Kunz, A. B., Journal of Physics C, 7, L231 (1974); Pantelides, S. T., and Brown, F. C., Physical Review Letters 33, 298 (1974).
4. Brown, F. C., Gahwiller, Ch., Fujita, H., Kunz, A. B., Schiefley, W., and Carrera, N., Physical Review B, 2, 2126 (1970); Brown, F. C., Gahwiller, Ch., Kunz, A. B., and Lipari, N. O., Physical Review Letters 25, 927 (1970); Scheifley, W., thesis, University of Illinois (unpublished, 1972).
5. Kunz, A. B., Physical Review B, 2, 5015 (1970); Kunz, A. B., and Lipari, N. O., Journal of Physics and Chemistry of Solids 32, 1141 (1971); Lipari, N. O., and Kunz, A. B., Physical Review B, 3, 491 (1971); Kunz, A. B., and Lipari, N. O., Physical Review B, 4, 1374 (1971).
6. Rossler, U., Physica Status Solidi 45, 483 (1971).
7. Kunz, A. B., and Mickish, D. J., Physical Review B, 8, 799 (1972).
8. Menzel, W. P., Lin, C. C., Fouquet, D., Lafon, E. E., and Chaney, R. C., Physical Review Letters 30, 1313 (1973).
9. Graber, M., thesis, Princeton University (unpublished, 1970).
10. Brown, F. C., Gahwiller, Ch., and Kunz, A. B., Solid State Communication 9, 487 (1971).
11. Hayes, W., Kunz, A. B., and Koch, E. E., Journal of Physics C (Solid State Physics) 4, L200 (1971).
12. Åberg, T., and Dehmer, J. L., Journal of Physics C 6, 1450 (1973).
13. Kunz, A. B., Mickish, D. J., and Collins, T. C., Physical Review Letters 31, 756 (1973).
14. Mickish, D. J., Kunz, A. B., Collins, T. C., Physical Review B, 9, 4461 (1974).
15. Onodera, Y., and Toyozawa, Y., Journal of the Physical Society of Japan 23, 833 (1967).
16. Collins, T. C., and Kunz, A. B., Physical Review A, to be published.

17. Knox, R. S., Theory of Excitons, Academic Press Inc., New York, 1963.
18. Seitz, F., The Modern Theory of Solids, McGraw-Hill Book Company, New York, 1940.
19. Kunz, A. B., and Collins, T. C., Journal of Physics B (Atomic and Molecular Physics) 7, L69 (1974).
20. Fetter, A. L., and Walecka, J. D., Quantum Theory of Many-Particle Systems, McGraw-Hill Book Company, New York, 1971.
21. Gilbert, T. L., in Molecular Orbitals in Chemistry, Physics and Biology, P. O. Lowdin and B. Pullman editors, Academic Press, New York, 1964.
22. Hedin, L., Physical Review 139, A796 (1965); Arkiv for Fysik 30, 19 (1965).
23. Brinkman, W. F., and Goodman, B., Physical Review 149, 597 (1966).
24. Lipari, N. O., and Fowler, W. B., Physical Review B, 2, 3354 (1970).
25. Fry, J. L., Physical Review 179, 892 (1969); Ph.D. thesis, University of California, Riverside, 1966, (unpublished).
26. Fowler, W. B., Physical Review 151, 657 (1966).
27. Toyozawa, Y., Progress of Theoretical Physics (Kyoto) 12, 421 (1954).
28. Kunz, A. B., Physical Review B, 6, 606 (1972).
29. Pantelides, S. T., Mickish, D. J., and Kunz, A. B., Physical Review, to be published.
30. Deutsch, P. W., Ph.D. thesis, University of Illinois, 1973, (unpublished); Deutsch, P. W., and Kunz, A. B., Journal of Chemical Physics, to be published.
31. In the case of ionization the site changes charge state. Still the electric field due to this is one or two orders of magnitude greater at the central site than for even the nearest neighbor in most cases, therefore justifying, in this case, the approximation.
32. In addition to the smallness of the virtual state relaxation in this atom in a field limit, one has the relaxation being further diminished in a solid due to overlap corrections which diffuse the virtual state even more. This is often good reason to neglect this effect entirely for excited electron. In LiF, however, even for excited levels this correction is small but so are the overlaps so that the retention if unnecessary is reasonable. The excited overlaps for the other crystals considered are rather larger than for LiF.
33. Koster, G. F., and Slater, J. C., Physical Review 154, 763 (1954).

34. Using a short range or truncated coulomb interaction probably introduces small error here since actual studies seem to show the exciton wave function is largely confined to the unit cell about the hole.
35. Callaway, J., Journal of Mathematics and Physics 5, 783 (1964).
36. Haensel, R., Keitel, G., Kock, E. E., Kosuch, N., and Skibowski, M., Physical Review Letters 25, 1281 (1970).

