ANL-7556

ANL-7556
RETURN TO ANL (IDAHO) LIBRARY.

Argonne National Laboratory

THE QUAD SCHEME:
A New Method for Phase Space Integration

by

F. M. Mueller, J. W. Garland,
M. H. Cohen, and K. H. Bennemann

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

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

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

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7556
Mathematics and
Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

THE QUAD SCHEME:
A New Method for Phase Space Integration

by

F. M. Mueller, J. W. Garland,
M. H. Cohen, and K. H. Bennemann

Solid State Science Division

March 1969

TABLE OF CONTENTS

	Page
ABSTRACT	5
I. INTRODUCTION	5
II. TREATMENT OF THE BRILLOUIN ZONE	7
III. DESCRIPTION OF THE PROGRAM	9
IV. FORTRAN LISTINGS	13
REFERENCES	43

THE QUAD SCHEME:
A New Method for Phase Space Integration

by

F. M. Mueller, J. W. Garland,
M. H. Cohen, and K. H. Bennemann

ABSTRACT

The QUAD scheme is a general computational method for using the results of band calculations to compute integrals over momentum space. A local quadratic expansion is used to fit the electronic band structure over a sufficiently small cubic region of the Brillouin zone. By means of this expansion, the contribution of the region to such quantities as the density of states is found by Monte Carlo sampling. The total density of states is formed by adding the contributions from all the small cubic regions. The QUAD procedure should be applicable to many other calculations. Fortran IV programs are included for the problem of density of states and the combined interpolation scheme. Parameters appropriate to relativistic fcc platinum bands are presented.

I. INTRODUCTION

Large-scale electronic computers have made possible detailed comparisons between theoretical calculations of electronic properties of crystalline systems and experimental results. The problem in making such comparisons has been in actually performing summations over momentum space. The summations involve evaluation of singular three-dimensional integrals. In the past the problem could be treated in two ways. Systems could be chosen which were amenable to simple approximations to both the eigenvalues and eigenvectors; the necessary integrals could then be performed analytically. Alternatively, experiments could be chosen which depended only on limited regions of the Brillouin zone (BZ), perhaps regions near the Fermi surface or near specific critical points such as symmetry points.

Such approximations and restrictions are quite appropriate for simple metals and semiconductors. However, the transition metals, rare earths, and actinides cannot be so approximated. Every eigenvector and matrix element in these systems has a strong $\underline{k}$ dependence because of hybridization between either the s and d electrons in the transition metals

and rare earths, or the s, d, and f electrons in the actinides. Mueller and Phillips[1] have shown that these materials have oscillator strengths which vary from 7.0 eV to 0.0 eV over a range in the BZ as short as 10% of a zone diameter. The energy bands of these materials are both numerous and convoluted. Even calculation of a simple property such as the density of states, which may be viewed as a "process" in which all the matrix elements are strictly constant, is difficult.

Recently a general technique, the QUAD scheme, has been developed for calculating the electronic properties of regular solids. The QUAD scheme divides an appropriate sector of the BZ into a sequence of small cubes. Eigenvalues and eigenvectors are found at a series of points within one of these cubes. These points are used to form expansion coefficients to full quadratic order by a least-squares procedure. The density of states over the cell is formed by using the expansion functions and by Monte Carlo sampling.

Such a sampling procedure for finding the density of states is approximate. One could, in principle, perform an exact calculation by finding the fractional volume of the cell enclosed by constant energy contours. Gilat and Raubenheimer[2] used this technique with a linear expansion to treat phonon spectra. A linear expansion is adequate for phonons when the bands are few in number and have little structure. Volume integration may be carried out quite easily in this linear approximation.

Electron bands of transition metals, however, are more numerous and much more convoluted than are phonon bands. The range of validity of the linear-expansion approximation for electron bands of transition metals is less than in phonon bands. Tests indicate that, if the expansion is to be accurate to 0.1% of the band width, a linear expansion breaks down about five times faster in the d-band region of the electronic band structure of copper as in the phonon bands of copper. For the same percent accuracy of approximation a net five times finer would be needed in the electron-band problem as in the phonon problem. Since both the number of diagonalizations and the number of cells would be increased by a factor of roughly $5^3 = 125$, we extended the linear approximation to include full quadratic terms. Volume integration of constant energy contours in the quadratic approximation is difficult, however, and involves the evaluation of many sets of incomplete elliptic integrals (as many as one has histogram boxes and cells). The procedure would be slow, difficult to program, and not easily extended to those cases where we want to consider $\underline{k}$-dependent matrix elements inside the integrand as well as singular energy delta functions.

The QUAD method may or may not be easier to use in more complicated calculations than the density of states in simple systems. Recently,

Saravia and Brust[3] successfully extended the linear-approximation method to evaluate matrix elements for the energy dependence of the long-wavelength dielectric function $\epsilon_2(\omega)$ of diamond. We believe that the linear-approximation method cannot be successfully applied to materials with more convoluted $\underline{k}$ dependence of electronic properties.

In this report we present two sets of Fortran programs used to treat fcc transition-metal bands. The physical and mathematical approximations involved in these programs have been presented in two companion articles.[4,5] The extension of this treatment to other crystal structures and materials should be straightforward.

II. TREATMENT OF THE BRILLOUIN ZONE

The Hamiltonian of a cubic material has eigenvalues which are degenerate under the 48 operations of the cubic point group. The full BZ may therefore be factorized into 48 irreducible BZ wedges, each containing exactly the same eigenvalue information. Although seemingly advantageous, a direct factorization of the full zone into one irreducible wedge can involve several problems. First, if Monte Carlo sampling methods are used to find the density of states (as below), each pseudorandom point must be tested to see if it is inside the wedge. Such testing requires several Fortran IF statements, which are slow compared to arithmetic statements.

A second possible problem induced by the wedge planes occurs when a regular or an irregular interpolation net is constructed over one cell. In an fcc structure, a simple cubic net has points falling inside and outside of the irreducible wedge planes, forming various gap or defect regions. Although a more complicated interpolation net, formed by shrinking the reciprocal-lattice basis vectors, will not overlap the wedge planes, its use requires a nonorthogonal basis set to derive the expansion coefficients needed for interpolation. Use of nonorthogonal bases can either raise the fractional error of the expansion coefficients to unacceptable levels, or leave some of the expansion coefficients indeterminate. For example, the simple cubic net discussed below cannot be used to find expansion coefficients to full third order.

Our treatment of the BZ avoids both of these problems at the outset; nevertheless, it does not need to diagonalize the Hamiltonian at any point outside of a single 1/48th of the BZ.

The QUAD technique treats the fcc BZ in the following manner:

1. The planes forming the square faces are extended so that a new BZ, exactly twice the volume of the old, is formed. The new BZ is a cube. Note that all of the 14 Bravis lattices can, by suitable patching-on of

degenerate pieces, be formed into, first, an orthorhombic system and then, by a scale transformation, into a cubic system. Thus our treatment, here restricted to the fcc system, can be applied to other systems as well.

2. Only points in the positive octant are considered. The volume of the resulting cube is 1/4 of the volume of a single BZ.

3. The 1/2 BZ distance (Γ-X) is divided by three sets of (MESH-1) parallel equispaced planes. The working volume is thus divided into (MESH**3) little cubes, called cells.

4. Each cell is further subdivided by three bisecting planes, forming 27 points at the corners, midedges, midfaces, and midpoint of one cell. Eigenvalues at the 27 points are used to find, by a least-squares procedure, a set of ten expansion coefficients to full quadratic order for each of the nine bands, derived from the model Hamiltonian appropriate to noble and transition metals.[4]

5. Each of the various cells is labeled by an index. Except for such labeling, all cells are treated by the program as logically equivalent. The sampling range for $\underline{k}$ is remapped for each cell so that it has components which are in the range -1 to +1. This remapping procedure greatly facilitates debugging. Since each cell is treated the same, it is only necessary to check in detail that one cell is being properly processed. In addition, accuracy in the matrix inversion procedure is increased, reducing the error in the expansion coefficients from about 1% to about 0.001% (both calculations were done using double precision arithmetic). Also, with the remapping, the 10 x 10 coefficient matrix inversion need be performed only once. In the present program the matrix inversion is done only once per run, reducing total execution time by a factor of about two.

6. Each of the subdivided points, which fall on a regular net of length (Γ-X)/(2*MESH), is mapped back onto one particular irreducible wedge. Thus the necessary energy eigenvalues at these points in 1/48th of the BZ are found only once, are stored in the matrix EMESH, and are drawn out as needed by the cells in the expanded cubic BZ segment. No more diagonalizations need be performed using this method than in other techniques, and the wedge planes need not be considered at all.

7. Monte Carlo sampling points are found throughout one cell, the nine energy eigenvalues appropriate to each point are found from the expansion coefficients CEP, and the energies are sorted into the HIST matrix.

The QUAD technique may be easily extended to include $\underline{k}$-dependent scalar, vector, or tensor quantities. When $\underline{k}$-dependent elements are important, two QUAD expansions are used: one, as above, for the energy eigenvalues, and another for the eigenvectors. From the eigenvectors, the

basis functions, and the various form factors, any k-dependent quantity may be calculated. Since the experimental accuracies are usually of the order of a few percent, about 10,000 Monte Carlo points per histogram box need be calculated to match these accuracies. Modern computers such as the Argonne 360/50/75 can calculate this many points in a few seconds, typically 500 to 1000 per second. Only 1/2 to 1 hour of machine time is needed to calculate a fine histogram of the most complicated experimental quantities. Once the basic band-structure parameters and various form factors are known, full band-structure details can be routinely included in the most difficult calculations using the QUAD technique.

III. DESCRIPTION OF THE PROGRAM

The program presented here forms a fine histogram representation of the density of electron states of relativistic fcc transition metals. As we have suggested above, such a calculation may be viewed as a "process" in which all matrix elements are strictly equal to one or zero. The extension of the QUAD scheme techniques for either manipulating core space or improving running speed to problems where matrix elements have a strong k dependence is straightforward. The program was written for the version 13, release 11 of Fortran IV(H) compiler of the IBM 360/50/75 system as the Computation Center of the Applied Mathematics Division at Argonne National Laboratory. This system has available about 250,000 single-precision words of fast core. The program now uses about 40,000 single-precision words, and 25,000 double-precision words of core. The need for core space could be substantially lowered by simply reducing the dimension of the storage matrices EMESH, IP, and HIST. Moreover, HOST, the integrated density of states, could be eliminated entirely.

A. Main Program

The main routine is used for manipulative purposes only, i.e., to call other routines, allocate storage space, transfer information, or call input-output. Statement cards are included at key points in the programs. Labeled common blocks are used to transmit information to the subroutines. The common blocks have been factored according to operational function.

B. Subroutines

The subroutines, with a brief description of each, are listed below in alphabetical order. Names containing an asterisk are used only by the combined interpolation scheme.

ALL

The ALL routine finds the energy eigenvalues for the nine bands of a fcc transition metal and loads the eigenvalues sequentially into EMESH,

the energy mesh. As discussed above, two nested meshes are used. An outer mesh labels the various cells of length $\left(\frac{2\pi}{a} * \frac{1}{MESH}\right)$; an inner mesh of length $\left(\frac{2\pi}{a} * \frac{1}{2*MESH}\right)$ is the regular eigenvalue net where $\underline{a}$ is the lattice constant. The inner mesh is assumed to be fcc and is generated in the first part of the ALL routine. The mesh points are given in integer arithmetic, and the integer vector [111] is added to them. For example, the point Γ becomes [111] and, assuming MESH = 4, the point X becomes [911]. The matrices IP and IK are logical inverses of each other and contain essentially the same information. The matrix IP yields the index point of a triplet set of numbers, whereas the matrix IK yields the $\underline{k}$-space coordinates of a given index number. Use of both of these matrices allows immediate retrieval of information without searching arrays. Appreciable computation time and core space are saved, and the procedure can be applied to many other computational problems.

CELL

The CELL routine returns the ten possible quadratic products of the vector BK. The first nine products (the harmonics s, p, and d) are scaled to agree with the Kubic Harmonics used in the interpolation scheme. This rescaling was done arbitrarily and represents an historical rather than a necessary choice.

*EIGEN B

The EIGEN B routine finds the highest NEV eigenvalues and eigenvectors of the matrix A to an accuracy of ACC. The secular equation is of order NSUB. If M = 0 only eigenvalues are found. These are stored in the array VALU, starting with the highest. EIGEN B uses the tridiagonal procedure of Givens and Householder, and was originally written by Burton Garbow of the Applied Mathematics Division at Argonne. It is considerably faster than the Jacobi technique.

*FILL

The FILL routine forms the spin-orbit matrix elements of the spin-up, spin-down d-functions. The original 9 x 9 real secular equation of the interpolation scheme is extended to an 18 x 18 complex matrix to include spin. The Hermitian Hamiltonian has been doubled in size to make a real matrix by the usual procedure: $(A + iB) \rightarrow \begin{pmatrix} A & B \\ -B & A \end{pmatrix}$, where $\underline{A}$ and $\underline{B}$ are block matrices. The spin-orbit coupling parameter is EP, and we assume a spin-orbit interaction of the form

$$H_{so} = \frac{EP}{2} \vec{\sigma} \cdot \vec{L},$$

where $\vec{L}$ acts on d states alone.

*HELD

The HELD routine generates the Hamiltonian of the 5 x 5 tight-binding fcc, d-d interaction block, and includes the effects of second-nearest neighbors. HELD is the result of straight coding of Table II of Slater and Koster.[6]

*HSOC

The HSOC routine generates two blocks of the model Hamiltonian. The first is the diagonal 4 x 4 block of the Hamiltonian between plane waves. The second is the off-diagonal 5 x 4 block of the Hamiltonian between tight-binding d states and plane waves. The five d basis functions have the same order as in the HELD routine. The four plane waves (OPW's) are the lowest four which are degenerate in the point W in the BZ.

*MATMPY

The MATMPY routine is a matrix multiplier used by EIGEN B.

MDINV

The MDINV routine calculates the inverse of the double-precision matrix A up to order N. A technique which pivots the matrix on the largest elements is used. The inverse is returned in the matrix A. The original A is destroyed.

*RECIP

The RECIP routine finds that point W, of three possible W points in the positive truncated octant, which is closest to the given point BK. The components of the lowest four degenerate plane waves are stored in the matrix AK and form the plane-wave basis set used in HSOC. The effect of RECIP is cancelled out in the present program by the restrictions imposed by ALL. RECIP becomes important in other cases--particularly when the effects of external magnetic fields are introduced into the Hamiltonian or when eigenvectors rather than eigenvalues are desired. The RECIP routine was kindly lent to us by E. I. Zornberg,[7] who has made several modifications of the original scheme.[4]

RANF

The RANF routine generates a set of pseudorandom numbers uniformly distributed between 0.0 and 1.0. The numbers are generated by a technique using the multiplicative congruence

$$X_{n+1} = X_n(2^{16}+1) \text{ modulo } 2^{31}.$$

This technique assumes an integer word width of 31 binary bits and left truncation of multiplicative overflow. The RANF function was written by Nancy W. Clark of the Applied Mathematics Division at Argonne. Modification to other computer systems should be straightforward and would be based on the integer word width of the system.

Several tests have been made of the accuracy of this pseudorandom generator. Three Chi square tests of the one-dimensional distribution of samples of 10,240 numbers with 1023 degrees of freedom gave the results 977, 1012, and 926. The 95% confidence interval for this test is (936, 1113).

Three Chi square tests of the three-dimensional distribution of triples of numbers from these same samples done with 511 degrees of freedom gave the results 485, 515, and 496. The 95% confidence interval for this test is (450, 577).

For our simple needs this routine considerably overkills the problem of generating pseudorandom numbers. However, other, less-precise techniques would not result in any substantial savings of computational time.

*RF and *RG

The RF and RG routines generate the two form factors $\langle k|d\rangle$ and $\langle k|H|d\rangle$, respectively, used in the interpolation scheme[4] subroutine HSOC. The form of both functions is basically $j_2(x)$, the ordinary spherical Bessel function of order 2.

*SET UP

The SET UP routine reads parameters into the various common block storage locations. The parameters provided in Table I are appropriate

TABLE I. The Data for the QUAD Scheme Programs for the Bands of Platinum

Parameter	Value	Parameter	Value
MESH	00010	P12	1.7928
ND	00060	P13	0.3700
HEAD	THE PLATINUM PARAMETERS	P14	0.2683
CNORM	0.9830	P15	0.0241
P1	-1.0689	P16	0.6545
P2	0.0027	P17	0.9745
P3	0.0033	P18	0.0010
P4	0.6768	P19	-0.0023
P5	-0.4961	P20	-0.0054
P6	-0.4845	P21	-0.0012
P7	-0.0636	MSIZE	00036
P8	+0.0254	NT	01000
P9	0.0193	TOP	0.8000
P10	-0.0003	BOT	0.2000
P11	1.9493		

to fcc platinum. The platinum energy bands have been fit to the accurate and extensive dHvA experiments of Ketterson and Windmiller.[8] Note that several of the parameters Pi have nonband structure functions, e.g., P16 is the Fermi energy. The parameter P17 uniformly widens or narrows the d-band complex.

SNORT

The SNORT routine finds the nine expansion coefficients CEP for the cell whose lowest corner point is given by IVEC. The element N69 in common block /HOT DOG/ is used as a test location to restrict matrix inversion of TR by MDINV to the first entry of SNORT. Every cell point used in SNORT has components restricted to ±1 or 0.

*SYM

The SYM routine is used by the interpolation scheme to symmetrize the Hamiltonian calculated in HELD and HSOC.

*ZEP

The ZEP routine calls the various routines used by the combined interpolation scheme to give the nine energy eigenvalues appropriate to the point CK in the BZ. If the QUAD technique were used with a different band structure method, the ZEP routine could be replaced by the appropriate equivalent. Thus, the generalization of the present method to APW, KKR, and OPW band structure methods should be immediate.

IV. FORTRAN LISTINGS

The Fortran listings of the main program and the subroutines follow:

```
C*****************************************************************
C      ......PROGRAM QUAD ------
C*****************************************************************
C
C      J360--PERFORM MONTE CARLO INTEGRALS
C      SO AS TO OBTAIN DENSITY OF STATES
C       FOR FCC TRANSITION METALS USING
C       THE COMBINED INTERPOLATION SCHEME
C
C.....F.M.MUELLER,J.W.GARLAND,M.H.COHEN,AND K.H.BENNEMANN.-- ----
C       JUNE 1968
C       MODIFIED BY S.G.DAS AUGUST 1969
C
C*****************************************************************
C      ND IS THE TOTAL NUMBER OF HISTOGRAM STATES
C      NT IS THE NUMBER OF POINTS PER BOX
C      TOP IS THE HIGHEST ENERGY VALUE
C      BOT IS THE LOWEST ENERGY VALUE
C*****************************************************************
C
1      FORMAT(F9.6)
```

```
4      FORMAT(I5)
5      FORMAT(1H1,5X,'THE TOTAL NUMBER OF POINTS IS ',I5,//)
6      FORMAT(5X,'THE RANGE IS FROM   ',F10.4,'   TO    ',F10.4,//)
7      FORMAT(1H1,////,10X,'THE DENSITY OF TOTAL ELECTRON STATES',///)
10     FORMAT(1H1,///,11X,'CELL',9X,'ENERGY',13X,'DENSITY OF STATES IN RY
     1 AND EV',9X,'TOTAL STATES',//)
11     FORMAT(5X,2I5,5X,F10.5,10X,E15.5,5X,E15.5,5X,E15.5)
12     FORMAT(I5,I5,4E15.4)
169  0 FORMAT(1H1,10X,'THE TOTAL NUMBER OF POINTS USED HERE IS ',I15,///
     1///)
500    FORMAT(10X,3I15,//)
502    FORMAT(10X,9F10.4,//)
505    FORMAT(1H1,10X,'THE TOTAL NUMBER OF K SPACE POINTS IS',I10,//)
2121   FORMAT(3F10.4)                                                    00002250
2122   FORMAT(3E15.8)                                                    00002300
C*****************************************************************************
C
C*****************************************************************************
C
       DIMENSIONIVEC(3),CEP(9,10),DK(3),C(10)
       DIMENSIONSMI1(9),DMI1(9),SMI(9),DMI2(9),DMIT1(9),DMIT2(9)
       DIMENSIONDENS(2000),Y5(680)
       DIMENSIONDME21(9,300),DE21(9,10),SME1(9,300),SEP1(9,10),SME2(9,
      8300),SEP2(9,10),DME31(9,300),DE31(9,10),DME32(9,300),DE32(9,
      4300),DME22(9,300),DE22(9,10)
       DIMENSIONX(340),Y1(680),Y2(680),Y3(680),Y4(340),X1(680)
       DIMENSION EMESH(9,2000),IP(25,25,25),IK(3,2000)
      1,CK(3),EG(9)
       DIMENSION ADV(3)
C
       REAL*8 HIST(13,500),HOST(13,500),HLOW(13),SUM,SNUM,SC
C
       COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
      7DME22,DE22,SMI,SMI1,DMI1,DMI2,DMIT1,DMIT2
          COMMON /DEBUG/ HIST,HOST                                       00003350
       COMMON/HOTDOG/N69
       COMMON/BLOK/EMESH,MESH,IK,NTOT,MC,IP
C
C*****************************************************************************
C
C      SET CONSTANTS
       N69 = 0
       SQ3=SQRT(3.)
       DO 111 I = 1,13
       HLOW(I) = 0.0D0
111    CONTINUE
C
C*****************************************************************************
C
C      SET THE MATRIX OF MESH POINTS AND THE NUMBER OF CALLS
       READ (5,4) MESH
       READ (5,4) ND
       XND=ND
C
C      SET UP DATA FROM CARDS
       CALL SET UP
C      PARAMETER DEFINITION COMPLETED
C
C*****************************************************************************
C
C      FILL UP LATTICE POINTS
       CALL ALL
C
C*****************************************************************************
C
       WRITE (6,505) NTOT
       DO 102 NC = 1,NTOT
```

```
      WRITE(6,500) (IK(I,NC),I = 1,3)
      WRITE(6,502) (EMESH (J6,NC),J6 = 1,9)
      WRITE(6,502) (DME21 (J6,NC),J6 = 1,9)
      WRITE(6,502) (DME22 (J6,NC),J6 = 1,9)
      WRITE(6,502) (DME31 (J6,NC),J6 = 1,9)
      WRITE(6,502) (DME32 (J6,NC),J6 = 1,9)
      WRITE(6,502) (SME1  (J6,NC),J6 = 1,9)
      WRITE(6,502) (SME2  (J6,NC),J6 = 1,9)
102   CONTINUE
C
C*****************************************************************************
C
C     SET UP SCALE
      IMESH=2*MESH+1
      XL=IMESH-1
      SCALE=8./XL
C
C     CLEAR HISTOGRAMS AND SET UP PARAMETERS
      DO62I=1,ND
      DO 62 J = 1,13
      HIST(J,I) = 0.0D0
      HOST(J,I) = 0.0D0
62    CONTINUE
C
C*****************************************************************************
C
      READ(5,4)NT
      READ(5,1)TOP,BOT
      DEL=(TOP-BOT)/ND
      NW = MESH*MESH*MESH*NT
      WRITE(6,169) NW
      WRITE(6,5)NT
      WRITE(6,6)BOT,TOP
C
C*****************************************************************************
      SNUM=0.
      MD=2*ND
C
C*****************************************************************************
C     BEGIN LOOP ON INDIVIDUAL LATTICE BOXES
C*****************************************************************************
C
      DO100IX=1,MESH
      DO100IY=1,MESH
      DO100IZ=1,MESH
C
      IVEC(1)=IX
      IVEC(2)=IY
      IVEC(3)=IZ
C
C     SET UP COEFFICIENTS
C
C*****************************************************************************
      CALLSNORT(IVEC,CEP)
C*****************************************************************************
C
C     RUN ON INNER LOOP USING EXPANSION COEFFICIENTS
C
      DO50NC=1,NT
C     SCALE UP VECTORS
      DO63I=1,3
      DK(I) = RANF(-1)*2-1
63    CONTINUE
C     READY TO ADD AND CALL CELL
      CALLCELL(DK,C)
C     LOAD IT IN 9 LEVELS
C
```

```
      DO40JG=1,9
C
      ER=0.
      PD1=0.0
      PD2=0.0
      PD31=0.0
      PD32=0.0
      PS1=0.0
      PS2=0.0
C
      DO30IG=1,10
C
      ER=ER+C(IG)*CEP(JG,IG)
      PS1=PS1+C(IG)*SEP1(JG,IG)
      PS2=PS2+C(IG)*SEP2(JG,IG)
      PD1=PD1+C(IG)*DE21(JG,IG)
      PD2=PD2+C(IG)*DE22(JG,IG)
      PD31=PD31+C(IG)*DE31(JG,IG)
      PD32=PD32+C(IG)*DE32(JG,IG)
C*****************************************************************
30    CONTINUE
C*****************************************************************
C
      IF(ER.LT.BOT)GOTO85
      IF(ER.GT.TOP)GOTO80
      NSET=(ER-BOT)/DEL+1.
C
C*****************************************************************
C     COUNT TWO ELECTRONS
C*****************************************************************
C
      HIST(13,NSET)=HIST(13,NSET)+2.0D0*(PS1+PS2)
C
80    CONTINUE
      GO TO 86
85    CONTINUE
C
      HLOW(JG) = HLOW(JG)+2.0D0
      HLOW(10) = HLOW(10)+2.0D0
      HLOW(11) = HLOW(11)+2.0D0*(PD1+PD2)
      HLOW(12) = HLOW(12)+2.0D0*(PD31+PD32)
      HLOW(13) = HLOW(13)+2.0D0*(PS1+PS2)
C
86    CONTINUE
C
      SNUM=SNUM+2.0D0
C
40    CONTINUE
50    CONTINUE
100   CONTINUE
C*****************************************************************
C     RENORMALIZE TO PROPER NUMBER OF ELECTRONS
C*****************************************************************
C
      SC = 18.0D0/SNUM
C
      DO 110 J = 1,13
      SUM = HLOW(J)*SC
      DO110I=1,ND
      HIST(J,I)=HIST(J,I)*SC
      SUM=SUM+HIST(J,I)
      HOST(J,I) = SUM
C     CONVERT TO DENSITY
      HIST(J,I) = HIST(J,I)/DEL
110   CONTINUE
C
```

```
C*****************************************************************
C     SET UP THE OUTPUT
C*****************************************************************
C
      DO 300 K = 1,13
      WRITE(6,7)
      NG=ND/55
      DO200I=1,NG
      WRITE(6,10)
      NTOP=I*55
      NBOT=NTOP-54
      DO200J=NBOT,NTOP
      ENG=J-1
      ENG=ENG*DEL+BOT
      HS=HIST(K,J)/13.59
      WRITE(6,11)K,J,ENG,HIST(K,J),HS,HOST(K,J)
C     PUNCH12,K,J,ENG,HIST(K,J),HS,HOST(K,J)
200   CONTINUE
300   CONTINUE
C
C*****************************************************************
      STOP
      END


      SUBROUTINE ALL
C*****************************************************************
C     J353-FILLS ALL POSSIBLE LATTIC POINTS
C     NOTE THAT IK IS ONE HIGHER THAN ZERO
C.....F.M.MUELLER,J.W.GARLAND,M.H.COHEN,AND K-H.BENNEMANN.......
C      JUNE 1968
C      MODIFIED BY S.G.DAS AUGUST 1969
C*****************************************************************
C     MESH IS NUMBER OF CELLS FROM
C     GAMMA TO X
C     IK SORTS VECTORS,EMESH ARE EIGENVALUES
C     NTOT IS THE NUMBER OF DOUBLE CELL POINTS
C*****************************************************************
C
      DIMENSIONDME21(9,300),DE21(9,10),SME1(9,300),SEP1(9,10),SME2(9,
     8300),SEP2(9,10),DME31(9,300),DE31(9,10),DME32(9,300),DE32(9,
     4300),DME22(9,300),DE22(9,10)
      DIMENSIONSMI1(9),DMI1(9),SMI(9),DMI2(9),DMIT1(9),DMIT2(9)
      DIMENSION EMESH(9,2000),IP(25,25,25),IK(3,2000)
      DIMENSION BK(3),EG(9)
C
      COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
     7DME22,DE22,SMI,SMI1,DMI1,DMI2,DMIT1,DMIT2
      COMMON/BLOK/EMESH,MESH,IK,NTOT,MC,IP
C
C*****************************************************************
C     DEFINE AND FILL-UP EMESH AND IK
C*****************************************************************
C
C     FOLLOWING ASSUMES F.C.C. LATTICE
C
      IMESH=2*MESH+1
C     IK FIRST
      INDEX=0
C
      DO50IY=1,IMESH
      DO50IX=1,IY
      DO50IZ=1,IX
C
C     PUT IN L PLANE
      IT=IX+IY+IZ-3
```

```
      IF(MESH*3.LT.IT) GOTO50
C
      INDEX=INDEX+1
      I=INDEX
      IP(IX,IY,IZ)=INDEX
      IK(1,I)=IX
      IK(2,I)=IY
      IK(3,I)=IZ
C
   50 CONTINUE
C
      NTOT=INDEX
C
C*****************************************************************
C     IK NOW DONE
C*****************************************************************
C
C     SET SCALE OF ZONE
C
      XL=IMESH-1
      SCALE=8./XL
C
C     LOAD EMESH
C
      DO100I=1,NTOT
C
C     RESET VECTOR SO THAT GAMMA=ZERO
C
      DO60J=1,3
      BK(J)=(IK(J,I)-1)*SCALE
   60 CONTINUE
C
C     CALL MATRIX AND EIGENVALUE ROUTINES
C
      CALL ZEP (BK,EG)
C
C*****************************************************************
C     FILE INTO EMESH
C*****************************************************************
C
      DO70J=1,9
C
      EMESH(J,I) = EG(J)
      SME1(J,I)=SMI1(J)
      SME2(J,I)=SMI(J)
      DME31(J,I)=DMIT1(J)
      DME32(J,I)=DMIT2(J)
      DME21(J,I)=DMI1(J)
      DME22(J,I)=DMI2(J)
C
   70 CONTINUE
C
  100 CONTINUE
C
C*****************************************************************
C     MESH LOADED
C*****************************************************************
C
      RETURN
      END


      SUBROUTINECELL(BK,C)
C*****************************************************************
C     J351-FOR USE IN DENSITY OF STATES
C     NOTE THAT A DIFFERENT SET OF EXPANSION FUNCTIONS
```

```
C      CAN BE USED BY MODIFYING THIS ROUTINE
C.....F.M.MUELLER,J.W.GARLAND,M.H.COHEN,AND K.H.BENNEMANN.......
C       JUNE 1968
C*****************************************************************
C
      DIMENSIONBK(3),C(10)
C
C*****************************************************************
C      SET UP VALUE AT THIS POINT OF KSPACE
C*****************************************************************
C
      SQ3=SQRT(3.)
C
      C(1)=1.
      C(2)=BK(1)
      C(3)=BK(2)
      C(4)=BK(3)
      C(5)=BK(1)*BK(2)
      C(6)=BK(1)*BK(3)
      C(7)=BK(2)*BK(3)
      C(8)=(BK(1)*BK(1)-BK(2)*BK(2))*.5
      C(9)=.5*(3.*BK(3)*BK(3)-1.)/SQ3
      C(10)=BK(1)*BK(1)+BK(2)*BK(2)+BK(3)*BK(3)
C
C*****************************************************************
C
      RETURN
C
      END


      FUNCTION CONVO(X)
C
C*****************************************************************
C       THIS FUNCTION IS A DUMMY
C       OTHER ENTRY TO CONVERT FROM FORTRAN 2 TO FORTRAN 4
C*****************************************************************
C
C       F.M.MUELLER MARCH 1966
C
C*****************************************************************
C
      CONVO = 1
      RETURN
C
      ENTRY ABSF(X)
      ABSF = ABS(X)
      RETURN
C
      ENTRY COSF(X)
      COSF = COS(X)
      RETURN
C
C
      ENTRY SINF(X)
      SINF = SIN(X)
      RETURN
C
      ENTRY SQRTF(X)
      SQRTF = SQRT(X)
      RETURN
C
      ENTRY SIGNF(X,Y)
      SIGNF = SIGN(X,Y)
      RETURN
C
C*****************************************************************
C
      END
```

```
      SUBROUTINE EIGENB(NSUB,NEV,M,ACC)
C*****************************************************************************
C     MODIFICATION OF ANLF202
C*****************************************************************************
C
      DIMENSION A(36,36),B(36,36),VALU(36),VAUL(36),DIAG(36),SUPERD(36),
     1 Q(36),S(36),C(36),D(36),IND(36),U(36)
      COMMON A,VALU
    2 N=NSUB
C
C     CALCULATE NORM OF MATRIX
C
    3 ANORM2=0.0
    4 DO 6 I=1,N
    5 DO 6 J=1,N
      IF(ABS(A(I,J)).LT.1E-12)A(I,J)=0.0
    6 ANORM2=ANORM2+A(I,J)**2
C
C     GENERATE IDENTITY MATRIX
    7 ANORM=SQRT (ANORM2)
    9 IF (M)10,45,10
   10 DO 40 I=1,N
   12 DO 40 J=1,N
   20 IF (I-J)35,25,35
   25 B(I,J)=1.0
   30 GO TO 40
   35 B(I,J)=0.0
   40 CONTINUE
C
C     PERFORM ROTATIONS TO REDUCE MATRIX TO JACOBI FORM
C
      DIMENSION WVEC(36),PVEC(36),QVEC(36)
   45 CONTINUE
   50 NN=N-2
   52 IF (NN)890,170,55
   55 DO 160 I=1,NN
   58 I1=I+1
   60 I2=I+2
   65 SUM=0.0
   68 DO 70 J=I2,N
   70 SUM=SUM+A(I,J)**2
   72 IF (SUM) 75,160, 75
   75 SUM=SQRT (SUM+A(I,I1)**2)
   80 WVEC(I1)=SQRT (     1.0+ABS (A(I,I1))/SUM)
   82 DIV=SIGN (     SUM*WVEC(I1),A(I,I1))
   83 DO 85 J=I2,N
   85 WVEC(J)=A(I,J)/DIV
   88 NI=N-I
   90 CALL MATMPY (NI,NI,WVEC(I1),A(I1,I1),PVEC(I1))
   95 CALL MATMPY (1,NI,WVEC(I1),PVEC(I1),SCALAR)
   98 SCALAR=0.5*SCALAR
  100 DO 105 J=I1,N
  105 QVEC(J)=PVEC(J)-SCALAR*WVEC(J)
  112 A(I,I1)=-SIGN (SUM,A(I,I1))
  114 DO 120 K=I1,N
  116 DO 120 L=I1,K
  118 A(K,L)=A(K,L)-(WVEC(K)*QVEC(L)+WVEC(L)*QVEC(K))
      IF(ABS(A(K,L)).LT.1E-12)A(K,L)=0.0
  120 A(L,K)=A(K,L)
  125 IF (M)130,160,130
  130 CALL MATMPY (N,NI,WVEC(I1),B(I1,1),QVEC)
  140 DO 150 K=I1,N
  145 DO 150 L=1,N
  150 B(K,L)=B(K,L)-WVEC(K)*QVEC(L)
  160 CONTINUE
C
C     MOVE JACOBI FORM ELEMENTS AND INITIALIZE EIGENVALUE BOUNDS
```

```
C
  170 DO 200 I=1,N
      VALU(I)=ANORM
      VALL(I)=-1.*ANORM
 200  DIAG(I)=A(I,I)
  210 DO 230 I=2,N
  220 SUPERD(I-1)=A(I-1,I)
  230 Q(I-1)=(SUPERD(I-1))**2
C
C     DETERMINE SIGNS OF PRINCIPAL MINORS
C
  235 TAU=0.0
  240 I=1
  260 MATCH=0
  270 T2=0.0
  275 T1=1.0
  277 DO 450 J=1,N
  280 P=DIAG(J)-TAU
  290 IF(T2) 300, 330, 300
  300 IF(T1) 310, 370, 310
  310 T=P*T1-Q(J-1)*T2
  320 GO TO 410
  330 IF(T1) 335, 350, 350
  335 T1=-1.0
  340 T=-P
  345 GO TO 410
  350 T1=1.0
  355 T=P
  360 GO TO 410
  370 IF(Q(J-1)) 380, 350, 380
  380 IF(T2) 400, 390, 390
  390 T=-1.0
  395 GO TO 410
  400 T=1.0
C
C     COUNT AGREEMENTS IN SIGN
C
  410 IF(T1) 425, 420, 420
  420 IF(T) 440, 430, 430
  425 IF(T) 430, 440, 440
  430 MATCH=MATCH+1
  440 IF (ABS (T)-1.0E20) 445,445,442
  442 T1=T1/T
  443 T=1.0
  445 T2=T1
  450 T1=T
C
C     ESTABLISH TIGHTER BOUNDS ON EIGENVALUES
C
  460 DO 530 K=1,N
  465 IF (K-MATCH) 470, 470, 520
  470 IF(TAU-VALL(K)) 530, 530, 480
  480 VALL(K)=TAU
  490 GO TO 530
  520 IF(TAU-VALU(K)) 525, 530, 530
  525 VALU(K)=TAU
  530 CONTINUE
  540 IF (VALU(I)-VALL(I)-ACC)570,570,550
  550 IF(VALU(I)) 560, 580, 560
  560 IF (ABS (VALL(I)/VALU(I)-1.0)-ACC)570,570,580
  570 I=I+1
  575 IF (I-NEV)540,540,590
  580 TAU=(VALL(I)+VALU(I))/2.0
  585 GO TO 260
C
C     JACOBI EIGENVECTORS BY ROTATIOAL TRIANGULARIZATION
C
  590 IF (M)593, 890, 593
```

```
  593 CONTINUE
  595 DO 610 I=1,N
  600 DO 610 J=1,N
  610 A(I,J)=0.0
  615 DO 850 I=1,N
  620 IF (I-1)625,625,621
  621 IF (VALU(I-1)-VALU(I)-5.0E-7)730,730,622
  622 IF (VALU(I-1))623,625,623
  623 IF (ABS (VALU(I)/VALU(I-1)-1.0)-5.0E-7)730,730,625
  625 COS=1.0
  628 SIN=0.0
  630 DO 700 J=1,N
  635 IF (J-1)680,680,640
  640 T=SQRT (T1**2+T2**2)
  645 COS=T1/T
  648 SIN=T2/T
  650 S(J-1)=SIN
  660 C(J-1)=COS
  670 D(J-1)=T1*COS+T2*SIN
  680 T1=(DIAG(J)-VALU(I))*COS-BETA*SIN
  690 T2=SUPERD(J)
897   FORMAT(8E12.4)
  700 BETA=SUPERD(J)*COS
  710 D(N)=T1
  720 DO 725 J=1,N
  725 IND(J)=0
  730 SMALLD=ANORM
  735 DO 780 J=1,N
  740 IF (IND(J)-1)750,780,780
  750 IF (ABS (SMALLD)-ABS (D(J)))780,780,760
  760 SMALLD=D(J)
  770 NN=J
  780 CONTINUE
  790 IND(NN)=1
  800 PRODS=1.0
  805 IF (NN-1)810,850,810
  810 DO 840 K=2,NN
  820 II=NN+1-K
  830 A(II+1,I)=C(II)*PRODS
  840 PRODS=-PRODS*S(II)
  850 A(1,I)=PRODS
C
C     FORM MATRIX PRODUCT OF ROTATION MATRIX WITH JACOBI VECTOR MATRIX
C
  855 DO 885 J=1,N
  860 DO 865 K=1,N
  865 U(K)=A(K,J)
  885 CALL MATMPY(N,N,U,B,A(1,J))
890   RETURN
      END


      SUBROUTINE FILL
C*****************************************************************************
C     RETURNS THE SYMMETRIZED SPIN ORBIT MATRIX
C     F.M.MUELLER MARCH 1966
C*****************************************************************************
C
      DIMENSION AD(12),AL(20,9),DIFER(20),BK(3)
      DIMENSION SOB(36,36),EV(36),D(5)
C
      COMMON SOB,EV,V0,V1,V2,V3,V4,ENGL,D,T,E1,R0,R1,EP,AD,AL,DIFER,BK
C*****************************************************************************
C
      C=SQRTF(3.)
C
C     SET UP THE IMAGINARY PARTS
```

```
C
      SOB(1,22) = 2.*EP
      SOB(1,29) = -EP
      SOB(2,21) = -EP
      SOB(2,28) = EP
      SOB(3,20) = EP
      SOB(3,31) = -EP
      SOB(3,32) = -C*EP
      SOB(4,19) = -2.*EP
      SOB(4,30) = EP
      SOB(5,30) = C*EP
      SOB(10,20) = -EP
      SOB(10,31) =-2.*EP
      SOB(12,22) = -EP
      SOB(12,23) = -C*EP
      SOB(12,29) = -EP
      SOB(11,19) = EP
      SOB(11,30) = EP
      SOB(13,21) = EP
      SOB(13,28) = 2.*EP
      SOB(14,21) = C*EP
C
C*****************************************************************
C     SET UP THE REAL PARTS
C*****************************************************************
C
      SOB(1,12) = EP
      SOB(3,10) = -EP
      SOB(2,13) = -EP
      SOB(2,14) = C*EP
      SOB(4,11) = EP
      SOB(5,11) = -C*EP
      SOB(19,30) = EP
      SOB(21,28) = -EP
      SOB(20,31) = -EP
      SOB(20,32) = C*EP
      SOB(22,29) = EP
      SOB(23,29) = -C*EP
C
      DO 30 I = 1,9
      DO 30 J = I,9
      DO 30 K = 1,3
C
      M = I+9*K
      N = J+9*K
C
30    SOB(M,N) = SOB(I,J)
C
C*****************************************************************
      CALL SYM(SOB,36)
C*****************************************************************
C
      RETURN
C
      END


      SUBROUTINE HELD
C
C*****************************************************************
C     SUBROUTINE TO GENERATE D PORTION OF HAMILTONIAN
C
C      THE ORDER OF THE BASIS STATES IS GIVEN BY.........
C      THE FIRST IS XY
C      THE SECOND IS XZ
C      THE THIRD IS YZ
```

```
C       THE FOURTH IS (X*X-Y*Y)/2
C       THE FIFTH IS (3*Z*Z-1)/(2*SQRT(3) )
C
C*****************************************************************************
C
C       F.M.MUELLER MARCH 1966
C
C*****************************************************************************
      DIMENSION PREM(16),BK(3),D(5),AD(12),S(36,36),EV(36),EG(9)
      DIMENSION AL(20,9),DIFER(20)
     1, CK(3),EZ(9)
      COMMON S,EV,V0,V1,V2,V3,V4,ENGL,D,T,EL,R0,R1,EP,A ,AL,DIFER,AK
     1,CNORM,MSIZE
C*****************************************************************************
C
C       SET CONSTANTS
C
      PI = 3.14159265
      B = SQRTF(3.0)
C
C       SET VARIABLES
C
      X = PI*AK(1)/8.
      Y = PI*AK(2)/8.
      Z = PI*AK(3)/8.
C
C       SET COS,SIN FUNCTIONS
C
      CX = COSF(X)
      CY = COSF(Y)
      CZ = COSF(Z)
      SX = SINF(X)
      SY = SINF(Y)
      SZ = SINF(Z)
      C2X = COSF(2.*X)
      C2Y = COSF(2.*Y)
      C2Z = COSF(2.*Z)
      S2X = SINF(2.*X)
      S2Y = SINF(2.*Y)
      S2Z = SINF(2.*Z)
C
C*****************************************************************************
C       GENERATION OF THE MATRIX ELEMENTS
C*****************************************************************************
C
C        THE DIAGONAL TERMS
      H(1,1) = A(1)+4.*A(2)*CX*CY+4.*A(3)*(CX*CZ+CY*CZ)
     1+2.*A(9)*(C2X+C2Y)+2.*A(10)*C2Z
      H(2,2) = A(1)+4.*A(2)*CX*CZ+4.*A(3)*(CX*CY+CZ*CY)
     1+2.*A(9)*(C2X+C2Z)+2.*A(10)*C2Y
      H(3,3) = A(1)+4.*A(2)*CY*CZ+4.*A(3)*(CX*CY+CZ*CX)
     1+2.*A(9)*(C2Y+C2Z)+2.*A(10)*C2X
      H(4,4) = A(6)+3.*A(7)*(CX*CZ+CY*CZ)+A(8)*(4.*CX*CY+CX*CZ +CY*CZ)
     1+(1.5)*A(11)*(C2X+C2Y)+(0.5)*A(12)*(C2X+C2Y+4.*C2Z)
      H(5,5) = A(6)+A(7)*(4.*CX*CY+CX*CZ+CY*CZ)+3.*A(8)*(CX*CZ+CY*CZ)
     1+(0.5)*A(11)*(C2X+C2Y+4.*C2Z)+(1.5)*A(12)*(C2X+C2Y)
C
C*****************************************************************************
C        THE OFF-DIAGONAL TERMS
C
      H(1,2) = -4.*A(4)*SY*SZ
      H(1,3) = -4.*A(4)*SX*SZ
      H(1,4) = 0.
      H(1,5) = -4.*A(5)*SX*SY
      H(2,3) = -4.*A(4)*SX*SY
      H(2,4) = 2.*B*A(5)*SX*SZ
      H(2,5) = 2.*A(5)*SX*SZ
```

```
      H(3,4) =-2.*B*A(5)*SY*SZ
      H(3,5) = 2.*A(5)*SY*SZ
      H(4,5) = B*A(7)*(CX*CZ-CY*CZ)-B*A(8)*(CX*CZ-CY*CZ)
     1+(0.5)*B*(A(11)-A(12))*(C2Y-C2X)
C
C*****************************************************************
C
      RETURN
      END


      SUBROUTINE HSOC
C
C*****************************************************************
C     THIS ROUTINE CALCULATES THE OPW-D STATE INTERACTION BLOCKS
C     F.M.MUELLER MARCH 1966
C*****************************************************************
C
      DIMENSION GG(4),GF(4),GN(4),ES(5,4),AK(4,3)
      DIMENSION S(36,36),EV(36),D(5),DIFER(20),BK(3),AL(20,9),AD(12)
      DIMENSION CK(3),ZK(4,3)
C
      COMMONS,EV,V0,V1,V2,V3,V4,ENGL
      COMMOND,T,EL,RB,RA,EP,AD,AL,DIFER,BK
      COMMON/THIEF/AK
C*****************************************************************
C
      CALL RECIP(BK)
C
      DO 2 I = 1,4
C
      GO = 0.
C
      DO 3 J = 1,3
      CK(J) = AK(I,J)+BK(J)
      ZK(I,J) = CK(J)
3     GO = CK(J)*CK(J) +GO
      GI = SQRT (GO)
C
      TEST=ABS(GI)
      BEST=10.E-20
C
      IF(TEST.GE.BEST)GOTO94
C
      GI=1.
C
94    CONTINUE
C
      A = CK(1)/GI
      B = CK(2)/GI
      C = CK(3)/GI
C
      ES(1,I) = A*B
      ES(2,I) = A*C
      ES(3,I) = B*C
      ES(4,I) = .5*(A*A-B*B)
      ES(5,I) = .5*SQRT (1./3.)*(3.*C*C-1.)
C
      CALL RG(GI,Z,T,RB)
C
      M = I+5
C
      DO 50 J = 1,5
50    S(J,M) = ES(J,I)*Z
C
      GG(I) = Z
C
      CALL RF(GI,Z,EL,RA)
```

```
C
      GF(I) = Z
      GN(I) = SQRT (1.-GF(I)*GF(I)/3.)
C
2     S(M,M) = GO*ENGL/48.+VO
C
C*****************************************************************
C     DEFINE OFF DIAGONAL ELEMENTS
C*****************************************************************
C
      S(6,7) = V1
      S(6,8) = V2
      S(6,9) = V1
      S(7,8) = V1
      S(7,9) = V2
      S(8,9) = V1
C
      DO 10 I = 6,9
C
      M = I-5
C
      DO 10 J = I,9
C
      N = J-5
C
      SUM = 0.
      SIM = 0.
C
      DO 11 K = 1,5
C
      SIM = SIM+ES(K,M)*ES(K,N)
C
      DO 11 L = 1,5
      A = S(K,L)
11    SUM = SUM+ES(K,M)*ES(L,N)*A
C
     OS(I,J) = S(I,J)-SUM*GF(M)*GF(N)+SIM*(GN(M)*GG(M)*GF(N)
     1+GN(N)*GG(N)*GF(M))
C
10    S(I,J) = S(I,J)/(GN(M)*GN(N))
C
C*****************************************************************
      RETURN
C*****************************************************************
C
      END


      SUBROUTINE MATMPY(M,N,X,A,Y)
C
C*****************************************************************
C      THIS ROUTINE IS A HELPER TO EIGENB
C*****************************************************************
C
      DIMENSION A(36,36),X(36),Y(36)
C
   10 DO 40 I=1,M
C
   20 Y(I)=0.0
C
   30 DO 40 J=1,N
C
   40 Y(I)=Y(I)+X(J)*A(J,I)
C
C*****************************************************************
   50 RETURN
C*****************************************************************
C
      END
```

```
      SUBROUTINE MDINV
C
C*****************************************************************************
C     J115 DOUBLE PRECISION MATRIX INVERTER
C     NOTE THAT THIS MATRIX INVERTER CAN ONLY BE USED ON NEARLY DIAGONAL
C     MATRICIES
C      WE ARE NOT RESTRICTED TO TO SYMMETRIC MATRICIES
C*****************************************************************************
      IMPLICIT REAL*8(A-E)
C*****************************************************************************
C
      DIMENSION A(35,35),IRL(35),ICL(35)
      COMMON/OVER/A,N
C
C*****************************************************************************
C      SET THE VARIOUS CALLS
C*****************************************************************************
C
      M = N
      M1 = M-1
      IC = 0
C
      DO 1 I = 1,M
      IRL( I) = 99
      ICL(I) = I
1     CONTINUE
C
      APIV = 0
      BPIV = 1.0E20
      ASUM = 0.
C
      DO 2 I = 1,M
      DO 2 J = 1,M
C
      AT = DABS(A(I,J))
C
      IF(AT.LT.1.E-30) AT = 0.
C
      ASUM = ASUM+AT*AT
C
2     CONTINUE
C
      IF(ASUM.LT.1.E-30) GO TO 5000
C
      ASUM = DSQRT(ASUM)
      ASUM = 1./ASUM
C
      DO 3 I = 1,M
      DO 3 J = 1,M
      A(I,J) = A(I,J)*ASUM
3     CONTINUE
C
C*****************************************************************************
C      FIND THE LARGEST ELEMENT
C*****************************************************************************
C
      DO 10 I = 1,M
      DO 11 J = 1,M
C
      A1=A(I,J)
      A2 = DABS( A1)
C
      IF(A2.LE.APIV) GO TO 11
C
      APIV = A2
C
      IT = I
```

```
      IU = J
C
11    CONTINUE
10    CONTINUE
C
C*****************************************************************************
C     BEGIN THE ITERATION SET
C*****************************************************************************
C
24    CONTINUE
C
      IF(APIV.EQ.0.)  GO TO 14
      IF(BPIV.LE.APIV) GO TO 15
C
      BPIV = APIV
C
15    IR = IT
      IS = IU
      ITEMP = IRL(IR)
      IRL(IR) = ICL(IS)
      ICL(IS) = ITEMP-100
C
      APIV = 0.
C
      DIV = A(IR,IS)
      A(IR,IS) = -1.
C
      DO 17 J = 1,N
17    A(IR,J) =-A(IR,J)/DIV
C
      DO 18 I = 1,M
C
      IF(I.EQ.IR) GO TO 18
C
19    AIS = A(I,IS)
C
      A(I,IS) = 0.
C
      DO 20 J = 1,M
      A(I,J) = A(I,J)+AIS*A(IR,J)
C
      IF(J.GT.M) GO TO 20
      IF(IRL(I).LE.50) GO TO 20
      IF(ICL(J).LT.1) GO TO 20
C
      AG = DABS(A(I,J))
C
      IF(AG.LT.APIV) GO TO 20
      APIV = AG
      IT = I
      IU = J
20    CONTINUE
C
18    CONTINUE
C
      IC = IC+1
C
      GO TO 24
C
C*****************************************************************************
C      THE END
C*****************************************************************************
C
14    CONTINUE
C
      DO 25 J = 1,M1
C
```

```
      K = 0
      KK = J+1
      JSM = ICL(J)
C
      DO 26 JJ = KK,M
C
      IF(JSM.LE.ICL(JJ)) GO TO 26
C
      K = JJ
      JSM = ICL(JJ)
C
26    CONTINUE
C
      IF(K.EQ.0) GO TO 25
C
      DO 29 I = 1,M
C
      ATP = A(I,J)
      A(I,J) = A(I,K)
      A(I,K) = ATP
C
29    CONTINUE
C
      ITEMP = ICL(J)
      ICL(J) = ICL(K)
      ICL(K) = ITEMP
C
25    CONTINUE
C
      DO 30 I = 1,M1
C
      K = 0
      KK = I+1
      ISM = IRL(I)
C
      DO 31 II = KK,M
C
      IF(ISM.LE. IRL(II)) GO TO 31
C
      K = II
      ISM = IRL(II)
C
31    CONTINUE
C
      IF(K.EQ.0) GO TO 30
C
      DO 34 J = 1,M
C
      ATP = A(I,J)
      A(I,J) = A(K,J)
      A(K,J) = ATP
C
34    CONTINUE
C
      ITEMP = IRL(I)
      IRL(I) = IRL(K)
      IRL(K) = ITEMP
C
30    CONTINUE
C
C*****************************************************************
C      RESCALE THE MATRIX
C*****************************************************************
C
      DO 50 I = 1,M
      DO 50 J = 1,M
C
      A(I,J) = A(I,J)*ASUM
```

```
C
50    CONTINUE
C
C*********************************************************************
      RETURN
C*********************************************************************
C
5000  CONTINUE
C
      WRITE(6,100)
100   FORMAT(10X,'THESE MATRIX ELEMENTS ARE TOO SINGULAR',////)
C
C*********************************************************************
      RETURN
C*********************************************************************
C
      END


      FUNCTION RANF(J)
C
C*********************************************************************
C     RANDOM NUMBER GENERATOR OF FORM X(I+1)=X(I)*(2**16+11) MOD 2**31
C     NOTE THAT THIS GENERATOR WORKS ON IBM 360 SYSTEM ONLY
C     MODIFICATION TO OTHER SYSTEMS IS STRAIGHT-FORWARD,HOWEVER
C     REPLACE 2**16 AND 2**31 BY RELATED WIDTH IN OTHER SYSTEM
C      WRITTEN BY NANCY CLARK OF AMD
C*********************************************************************
C
      EQUIVALENCE (X,IX)
      DATA IX/3125/
C
    2 IX=IX*65547
C
      IF(IX)5,6,6
C
    5 IX=IX+2147483647+1
C
    6 IF (J.GE.0) GOTO 8
C
      IF (IX.LT.8388608) GOTO 7
C
      YFL=IX/8
C
      RANF=YFL*.3725291E-8
C
C*********************************************************************
      RETURN
C*********************************************************************
C
    7 YFL=IX
      RANF=YFL*.4656613E-9
C
C*********************************************************************
      RETURN
C*********************************************************************
C
    8 RANF=X
C
C*********************************************************************
      RETURN
C*********************************************************************
C
C
C
C*********************************************************************
      ENTRY IRANF(J)
```

```
C*****************************************************************
C
   12 IX=IX*65547
C
      IF(IX)15,16,16
C
   15 IX=IX+2147483647+1
   16 IRANF=IX
C
C*****************************************************************
      RETURN
C*****************************************************************
C
C
C
C*****************************************************************
      ENTRY RANSET(J)
C*****************************************************************
C
      IX=J
      RANSET=0.
C
C*****************************************************************
      RETURN
C*****************************************************************
C
C
C
C*****************************************************************
      ENTRY RANGET(J)
C*****************************************************************
      J=IX
      RANGET=0.
C
C*****************************************************************
      RETURN
C*****************************************************************
      END


      SUBROUTINE RECIP(EK)
C
C*****************************************************************
C     THIS MODIFICATION TO COMBINED INTERPOLATION SCHEME
C     SO THAT BK MAY BE IN ANY PART OF THE BZ
C     INTRODUCED BY E.I.ZORNBERG
C*****************************************************************
C
      DIMENSION AK(4,3),BK(3),EK(3),IL(3)
      COMMON/THIEF/AK
C
C*****************************************************************
C
72    CONTINUE
C
      DO 10 I=1,3
10    BK(I)=ABS(EK(I))
C
      TES=BK(1)+BK(2)+BK(3)
C
      IF(TES.LE.12.) GO TO 73
C
      DO 13 I=1,3
13    EK(I)=EK(I)-SIGN(8.,EK(I))
C
      GO TO 72
```

```
C
C*****************************************************************************
73    CONTINUE
C*****************************************************************************
C
      DO 11 I=1,3
11    IL(I)=I
C
      DO 12 I=1,4
C
      DO 12 J=1,3
C
12    AK(I,J)=0.
C
      DO 70 I = 1,2
      DO 70 J = 2,3
C
      TEST = BK(I)-BK(J)
C
      IF(TEST.LE.0.) GO TO 70
C
      SAVE = BK(I)
      BK(I) = BK(J)
      BK(J) = SAVE
      KAVE=IL(I)
      IL(I)=IL(J)
      IL(J)=KAVE
C
70    CONTINUE
C
      BK(2) = BK(3)
      SAVE = BK(1)
      BK(1) = BK(2)
      BK(3) = SAVE
      KAVE=IL(1)
      IL(1)=IL(2)
      IL(2)=IL(3)
      IL(3)=KAVE
      I1=IL(1)
      I2=IL(2)
      I3=IL(3)
C
      IF(BK(I2).LE.8.) GO TO 71
C
      E2=SIGN(1.,EK(I2))
      EK(I2)=EK(I2)-E2 *16.
C
      GO TO 72
C
71    CONTINUE
C
      E2=SIGN(1.,EK(I2))
      AK(3,I2)=-E2*16.
      AK(2,I3)=-8.
      AK(4,I3)=8.
C
      DO 14 I=2,4,2
C
      AK(I,I1)=-SIGN(8.,EK(I1))
14    AK(I,I2)=-E2*8.
C
C*****************************************************************************
      RETURN
C*****************************************************************************
C
      END
```

```
      SUBROUTINE RF(GI,Z,T,RR)
C
C*****************************************************************
C     THIS ROUTINE CALCULATES THE OVERLAP FORM FACTOR
C     F.M.MUELLER MARCH 1966
C*****************************************************************
C
31    A=GI*RR
C
      IF(A-5.8)3,3,32
C
3     Z = (3./A**3-1./A)*SIN (A)-3.*COS (A)/A**2
      Z = T*ABS (Z)
C
      GO TO 16
C
 32   Z=0.
C
C*****************************************************************
16    RETURN
C*****************************************************************
C
      END
C
      SUBROUTINE RG(GI,Z,T,RR)
C
C*****************************************************************
C     THIS ROUTINE CALCULATES THE HYBRIDIZATION FORM FACTOR
C     F.M.MUELLER MARCH 1966
C*****************************************************************
      C = T
      A=GI*RR
C
      IF(A-4.25) 20,20,10
C
10    IF(A-5.1) 16,15,15
C
15    Z = 0.
C
C*****************************************************************
      RETURN
C*****************************************************************
C
16    C = T*(5.1-A)/.85
20    Z = (3./A**3-1./A)*SIN (A)-3.*COS (A)/A**2
      Z = Z*C
C
C*****************************************************************
      RETURN
C*****************************************************************
C
      END


      SUBROUTINE SETUP
C
C*****************************************************************
C     J355 -- TO READ IN THE DATA FROM THE BACK DECK
C      AND SET UP ALL OF THE VARIOUS PARAMETERS
C     F.M.MUELLER MARCH 1966
C*****************************************************************
C
100   FORMAT(I5)
200   FORMAT(3E15.6)
300   FORMAT(F9.6)
350   FORMAT(10X,I5,10X,F10.4)
600   FORMAT(80A1)
```

```
700   FORMAT(1H1,10X,80A1,///)
800   FORMAT(///,10X,'D BAND PARAMETERS')
C
C*****************************************************************
C
      DIMENSION HEAD(80)
      DIMENSION AL(20,9),DIFER(20)
      DIMENSION PREM(21),BK(3),D(5),AD(12),S(36,36),EV(36),EG(9)
      COMMON S,EV,V0,V1,V2,V3,V4,ENGL,D,T,EL,R0,R1,EP,AD,AL,DIFER,BK
     1,CNORM,MSIZE
C
C*****************************************************************
C
      READ(5,600) HEAD
      WRITE(6,700) HEAD
C
C*****************************************************************
C     READ VALUES OF PARAMETERS
C*****************************************************************
C
      READ (5,300) CNORM
      READ(5,300) PREM
C
      I = 0
C
      WRITE(6,350) I,CNORM
      WRITE(6,350)(I,PREM(I),I = 1,21)
C
      READ (5,100)MSIZE
C
C*****************************************************************
C      SET UP PARAMETERS
C*****************************************************************
C
      V0=PREM(1)
      V1=PREM(2)
      V2=PREM(3)
      ENGL=PREM(4)
C
      DO20 I=1,4
      I1=I+4
 20   D(I)=PREM(I1)
      D(5)=PREM(10)
C
      T=PREM(11)
      EL=PREM(12)
      R0=PREM(13)
      R1=PREM(14)
      EP=PREM(15)
      EF=PREM(16)
      V3=0.
      V4=0.
C
C*****************************************************************
C      VARY THE WHOLE WIDTH OF THE D BAND
C*****************************************************************
C
      DO 169 JJ = 3,5
C
      D(JJ) = D(JJ)*PREM(17)
C
169   CONTINUE
C
C*****************************************************************
C     SET UP D BAND PARAMETERS
C*****************************************************************
C
```

```
      AD(1)=D(1)
      AD(2)=.25*(3.*D(3)+D(5))
      AD(3)=.5*(D(4)+D(5))
      AD(4)=.5*(D(4)-D(5))
      AD(5)=-.25*SQRT(3.)*(D(3)-D(5))
      AD(6)=D(2)+D(1)
      AD(7)=.25*(D(3)+3.*D(5))
      AD(8)=PREM(9)*PREM(17)
C       SET THE SECOND NEIGHBOR PARAMETERS
      AD(9) = PREM(18)*PREM(17)
      AD(10) = PREM(19)*PREM(17)
      AD(11) = PREM(20)*PREM(17)
      AD(12) = PREM(21)*PREM(17)
C
      WRITE(6,800)
      WRITE(6,200) AD
C
C*****************************************************************
      RETURN
C*****************************************************************
C
      END


      SUBROUTINE SNORT(IVEC,CEP)
C
C*****************************************************************
C     J354-FINDS EXPANSION COEF'S TO SECOND ORDER
C     EMESH AND IK ALREADY LOADED
C.....F.M.MUELLER,J.W.GARLAND,M.H.COHEN,AND K.H.BENNEMANN.......
C      JUNE 1968
C      MODIFIED BY S.G.DAS AUGUST 1969
C*****************************************************************
C
C     IVEC IS LOWEST CORNER POINT IN
C     MESH-SIZED LATTICE
C
C*****************************************************************
C
      DIMENSIONDME21(9,300),DE21(9,10),SME1(9,300),SEP1(9,10),SME2(9,
     8300),SEP2(9,10),DME31(9,300),DE31(9,10),DME32(9,300),DE32(9,
     4300),DME22(9,300),DE22(9,10)
      DIMENSIONSMI1(9),DMI1(9),SMI(9),DMI2(9),DMIT1(9),DMIT2(9)
      DIMENSION EMESH(9,2000),IP(25,25,25),IK(3,2000)
      DIMENSIONIVEC(3),ISVEC(3),IZ(3),IR(3)
      DIMENSIONEG(27),Z(3),G(10),IM(3,27)
      DIMENSION CEP(9,10)
      REAL U(10,27)
C
      REAL*8 TR(35,35),R(27,10)
C
      COMMON/OVER/TR,NGO
      COMMON/BLOK/EMESH,MESH,IK,NTOT,MC,IP
      COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
     7DME22,DE22,SMI,SMI1,DMI1,DMI2,DMIT1,DMIT2
      COMMON/HOTDOG/N69
C
C*****************************************************************
C
      NGO = 10
C
C*****************************************************************
C      INITIATE THE CEP'S TO ZERO
C*****************************************************************
C
      DO 1 I = 1,9
```

```
      DO 1 J = 1,10
C
      CEP(I,J) = 0.
      SEP1(I,J)=0.0
      SEP2(I,J)=0.0
      DE21(I,J)=0.0
      DE22(I,J)=0.0
      DE32(I,J)=0.0
      DE31(I,J)=0.0
C
    1 CONTINUE
C
      DO4I=1,10
      DO3J=1,27
3     R(J,I)=0.
C
4     CONTINUE
C
C*****************************************************************
C     SET SCALE FOR DOUBLE SIZED MESH
C*****************************************************************
C
      IMESH=2*MESH+1
      XL=IMESH-1
      SCALE=8./XL
C
C*****************************************************************
C     PUT IVEC INTO DOUBLE LATTICE
C*****************************************************************
C
      DO10I=1,3
C
      ISVEC(I) = 2*(IVEC(I)-1)
C
   10 CONTINUE
C
C*****************************************************************
C     SET UP TRIPLE-LOOP FOR 27 CELL POINTS
C*****************************************************************
C
      NR=0
C
      DO20I=1,3
      DO20J=1,3
      DO20K=1,3
C
C*****************************************************************
C     IF WANT RESTRICTION TO FCC PUT HERE
C*****************************************************************
C
      IZ(1) = I-2
      IZ(2) = J-2
      IZ(3) = K-2
C
      NR = NR+1
C
      DO15L=1,3
C
      IR(L) = IZ(L)
      IM(L,NR)=IR(L)
      Z(L)=IR(L)
C
   15 CONTINUE
C
      CALLCELL(Z,G)
C
      DO16L=1,10
```

```
C
      R(NR,L)=G(L)
   16 CONTINUE
C
   20 CONTINUE
C
C*****************************************************************************
C     FORM TR MATRIX AND INVERT IT
C*****************************************************************************
C
      IF (N69.EQ.69) GO TO 169
C
      DO30I=1,10
      DO30J=1,10
C
      TR(I,J)=0.
C
      DO30K=1,NR
C
      TR(I,J)=TR(I,J)+R(K,I)*R(K,J)
C
   30 CONTINUE
C
      CALL MDINV
C
169   CONTINUE
C
C*****************************************************************************
C     FORM U MATRIX
C*****************************************************************************
C
      DO35I=1,10
      DO35J=1,NR
C
      U(I,J)=0.
C
      DO34L=1,10
C
      U(I,J)=U(I,J)+TR(I,L)*R(J,L)
C
   34 CONTINUE
   35 CONTINUE
C
C*****************************************************************************
C     SUM OVER THE FITTING POINTS
C*****************************************************************************
C
      DO50L=1,NR
C
C*****************************************************************************
C     SET UP K VECTOR
C*****************************************************************************
C
      ISUM=0
C
      DO41K=1,3
C
      IR(K)=IM(K,L)+ISVEC(K)+2
C
C*****************************************************************************
C     CHECK ON LENGTH
C*****************************************************************************
C
      ISUM=ISUM+IR(K)-1
C
   41 CONTINUE
C
```

```
      IDIF=ISUM-3*MESH
      IF(IDIF.LE.0)GOTO43
C
C*****************************************************************
C     VECTOR OUTSIDE
C*****************************************************************
C
      DO42I=1,3
C
      IR(I) = (MESH*2)-(IR(I)-1)
      IR(I)=IABS(IR(I))+1
C
   42 CONTINUE
C
   43 CONTINUE
C
C*****************************************************************
C     FOLD INTO PROPER ZONE: Y>X>Z
C*****************************************************************
C
      DO45I=1,2
C
      I1=I+1
C
      DO45J=I1,3
C
      IDIF=IR(I)-IR(J)
C
      IF(IDIF.GE.0)GOTO45
C
      ISAVE=IR(I)
      IR(I)=IR(J)
      IR(J)=ISAVE
C
   45 CONTINUE
C
      IZE=IR(3)
      IX=IR(2)
      IY=IR(1)
C
C*****************************************************************
C     VECTOR NOW PROPER
C*****************************************************************
C
      INDEX=IP(IX,IY,IZE)
C
C*****************************************************************
C     SUM OVER THE BANDS
C*****************************************************************
C
      DO49N=1,9
C
      ENG=EMESH(N,INDEX)
      PRS1=SME1(N,INDEX)
      PRS2=SME2(N,INDEX)
      PRD1=DME21(N,INDEX)
      PRD2=DME22(N,INDEX)
      PRD32=DME32(N,INDEX)
      PRD31=DME31(N,INDEX)
C
      DO48K=1,10
C
C*****************************************************************
C     SETUP FINAL SUM
C*****************************************************************
C
      CEP(N,K)=CEP(N,K)+U(K,L)*ENG
```

```
      SEP1(N,K)=SEP1(N,K)+U(K,L)*PRS1
      SEP2(N,K)=SEP2(N,K)+U(K,L)*PRS2
      DE21(N,K)=DE21(N,K)+U(K,L)*PRD1
      DE22(N,K)=DE22(N,K)+U(K,L)*PRD2
      DE31(N,K)=DE31(N,K)+U(K,L)*PRD31
      DE32(N,K)=DE32(N,K)+U(K,L)*PRD32
C
   48 CONTINUE
   49 CONTINUE
   50 CONTINUE
C
      N69 = 69
C
C*****************************************************************
      RETURN
C*****************************************************************
C
      END


      SUBROUTINE SYM(S,N)
C
C*****************************************************************
C     SUBROUTINE TO SYMMERTIZE THE HAMILTONIAN MATRIX
C     F.M.MUELLER MARCH 1966
C*****************************************************************
C
      DIMENSION S(36,36)
C
      DO 10 I = 1,N
      DO 10 J = I,N
10    S(J,I) = S(I,J)
C
C*****************************************************************
      RETURN
C*****************************************************************
C
      END


      SUBROUTINE ZEP(CK,EZ)
C
C*****************************************************************
C      J356 --FIND THE EIGEN VALUES FOR THIS POINT IN THE B.Z.
C     F.M.MUELLER MARCH 1966
C      MODIFIED BY S.G.DAS AUGUST 1969
C*****************************************************************
C
      DIMENSIONSMI1(9),DMI1(9),SMI(9),DMI2(9),DMIT1(9),DMIT2(9)
      DIMENSIONDME21(9,300),DE21(9,10),SME1(9,300),SEP1(9,10),SME2(9,
     8300),SEP2(9,10),DME31(9,300),DE31(9,10),DME32(9,300),DE32(9,
     4300),DME22(9,300),DE22(9,10)
      DIMENSION AL(20,9),DIFER(20)
     1, CK(3),EZ(9)
      DIMENSION PREM(16),BK(3),D(5),AD(12),S(36,36),EV(36),EG(9)
C
      COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
     7DME22,DE22,SMI,SMI1,DMI1,DMI2,DMIT1,DMIT2
      COMMON S,E,V,V0,V1,V2,V3,V4,ENGL,D,T,EL,R0,R1,EP,AD,AL,DIFER,BK
     1,CNORM,MSIZE
C*****************************************************************
C
14    FORMAT(2I6,6F10.5)                                          00049712
C
```

```
      DO9I=1,9                                                          00050020
C
      SMI=0.0
      SMI1=0.0
      DMI1=0.0
      DMI2=0.0
      DMIT2=0.0
      DMIT1=0.0
C
    9 CONTINUE                                                          00050070
C
      DO 20 I = 1,3
C
      BK(I) = CK(I)
C
20    CONTINUE
C
      DO30I=1,MSIZE
      DO 30 J = 1,MSIZE
30    S(I,J)=0.
C
      CALL HELD
      CALL SYM(S,5)
      CALL HSOC
      CALL SYM(S,9)
C
      IF(MSIZE.LE.9) GO TO 35
      CALL FILL
C
C*****************************************************************
C     PUT ON SMALL MAGNETIC FIELD TO SPLIT DEGENRACIES              00050910
C*****************************************************************
C
      AMAG=0.0001                                                       00050920
C
      DO10I=1,9                                                         00050930
C
      J=I+9                                                             00050940
      L=I+18                                                            00050950
      M=J+18                                                            00050960
C
      S(I,I)=S(I,I)+AMAG                                                00050970
      S(L,L)=S(L,L)+AMAG                                                00050980
      S(J,J)=S(J,J)-AMAG                                                00050990
      S(M,M)=S(M,M)-AMAG                                                00050991
C
10    CONTINUE                                                          00050992
C
 35   CONTINUE
C
      ACC=.0001
C
      CALL EIGENB(MSIZE,MSIZE,MSIZE,ACC)                                00051400
C
      MC=MSIZE/36
C
      DO40I=1,9
C
      I1=10-I
      IT=(I1-1)*MC*3+I1
C
      EZ(I) = EV(IT)+CNORM
40    EG(I) = EV(IT)+CNORM
C
      DO11NLEVEL=1,9
C                                                                       00051910
      ISUM=10-NLEVEL
                                                                        00051920
```

```
C
      DO1000 NGO=1,1
C
      IS=ISUM+MC*3*(ISUM-1)                                             00051940
      IS=IS+NGO-1                                                       00051950
C
      D1=0.0                                                            00051980
      D2=0.0                                                            00051990
      D31=0.0
      D32=0.0
      S1=0.0
      S2=0.0
C
      DO28I=4,5
C
      K=I+9                                                             00051992
      J=I+18                                                            00051993
      L=K+18                                                            00051994
C
      D1=D1+S(I,IS)**2+S(J,IS)**2                                       00051995
      D2=D2+S(K,IS)**2+S(L,IS)**2                                       00051996
C
28    CONTINUE                                                          00051997
C
      DO281I=1,3
C
      K=I+9
      J=I+18
      L=K+18
C
      D31=D31+S(I,IS)**2+S(J,IS)**2
      D32=D32+S(K,IS)**2+S(L,IS)**2
C
281   CONTINUE
C
      DO271I=6,9
C
      K=I+9
      J=I+18
      L=K+18
C
      S1=S1+S(I,IS)+S(J,IS)
      S2=S2+S(K,IS)+S(L,IS)
C
271   CONTINUE
C
      S1=S1*S1
      S2=S2*S2
C
      SMI1(NLEVEL)=S1
      SMI(NLEVEL)=S2
      DMIT1(NLEVEL)=D31
      DMIT2(NLEVEL)=D32
      DMI1(NLEVEL)=D1
      DMI2(NLEVEL)=D2
C
1000   CONTINUE                                                         00052070
11    CONTINUE                                                          00052077
C
C*****************************************************************************
      RETURN
C*****************************************************************************
C
      END
```

```
/*
//GO.SYSIN DD *
     5
   340
 THE PARAMETERS FOR PLATINUM FOUND FROM THE WONDERFUL DHVA WORK OF LEESJOHN
 0.983
-1.068900
 0.002700
 0.003300
 0.676799
-0.496138
0.011668
-0.063561
 0.025417
 0.019276
-0.000300
 1.949329
 1.792763
 0.370036
 0.268328
 0.024065
 .6565
 0.974494
 0.001000
-0.002300
 0.005400
-0.001200
    36
 2200
1.4
-0.2
/*
```

REFERENCES

1. Mueller, F. M., and Phillips, J. C., Phys. Rev. *157*, 600 (1967).

2. Gilat, G., and Raubenheimer, L. J., Phys. Rev. *144*, 390 (1966) and references therein.

3. Saravia, L. R., and Brust, D., Phys. Rev. *170*, 683 (1968), and to be published.

4. Mueller, F. M., Phys. Rev. *153*, 659 (1967); Cohen, M. H., and Mueller, F. M., *The Atomic and Electronic Structure of Metals* (American Society for Metals, Metals Park, Ohio, 1967), Ch. 4, pp. 61-80.

5. Mueller, F. M., Garland, J. W., Cohen, M. H., and Bennemann, K. H., to be published.

6. Slater, J. C., and Koster, G. F., Phys. Rev. *94*, 1498 (1954).

7. Zornberg, E. I., to be published.

8. Ketterson, J. B., and Windmiller, L. R., Phys. Rev. Letters *20*, 7 (1968).

ARGONNE NATIONAL LAB WEST
3 4444 00007947 5

N

B