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
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
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  Kansas State University  The Ohio State University
Carnegie-Mellon University  The University of Kansas  Ohio University
Case Western Reserve University  Loyola University  The Pennsylvania State University
The University of Chicago  Marquette University  Purdue University
University of Cincinnati  Michigan State University  Saint Louis University
Illinois Institute of Technology  The University of Michigan  Southern Illinois University
University of Illinois  University of Minnesota  University of Texas
Indiana University  University of Missouri  Washington University
Iowa State University  Northwestern University  Wayne State University
The University of Iowa  University of Notre Dame  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
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

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

March 1969

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>II. TREATMENT OF THE BRILLOUIN ZONE</td>
<td>7</td>
</tr>
<tr>
<td>III. DESCRIPTION OF THE PROGRAM</td>
<td>9</td>
</tr>
<tr>
<td>IV. FORTRAN LISTINGS</td>
<td>13</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>43</td>
</tr>
</tbody>
</table>
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 \( k \) 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 \(k\)-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\textsuperscript{3} successfully extended the linear-approximation method to evaluate matrix elements for the energy dependence of the long-wavelength dielectric function $\varepsilon_2(\omega)$ of diamond. We believe that the linear-approximation method cannot be successfully applied to materials with more convoluted $k$ 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.\textsuperscript{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 ($\Gamma$-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 $k$ 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 ($\Gamma$-X)/(2*MESH), is mapped back onto one particular irreducible wedge. Thus the necessary energy eigenvalues at these points in $1/48$th 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 $k$-dependent scalar, vector, or tensor quantities. When $k$-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 \( \frac{1}{2} \) 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 \( \left( \frac{2\pi}{a} \times \frac{1}{MESH} \right) \); an inner mesh of length \( \left( \frac{2\pi}{a} \times \frac{1}{2\times MESH} \right) \) is the regular eigenvalue net where \( a \) 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 \( \Gamma \) 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 \( k \)-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: \( (A + iB) \mapsto \begin{pmatrix} A & B \\ -B & A \end{pmatrix} \), where \( A \) and \( B \) are block matrices. The spin-orbit coupling parameter is \( EP \), and we assume a spin-orbit interaction of the form

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

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, 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 <k|d> and <k|H|d>, respectively, used in the interpolation scheme 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>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESH</td>
<td>00010</td>
<td>P12</td>
<td>1.7928</td>
</tr>
<tr>
<td>ND</td>
<td>00000</td>
<td>P13</td>
<td>0.3700</td>
</tr>
<tr>
<td>HEAD</td>
<td>THE PLATINUM PARAMETERS</td>
<td>P14</td>
<td>0.2683</td>
</tr>
<tr>
<td>CNORM</td>
<td>0.9830</td>
<td>P15</td>
<td>0.0241</td>
</tr>
<tr>
<td>P1</td>
<td>-1.0689</td>
<td>P16</td>
<td>0.6545</td>
</tr>
<tr>
<td>P2</td>
<td>0.0027</td>
<td>P17</td>
<td>0.9745</td>
</tr>
<tr>
<td>P3</td>
<td>0.0033</td>
<td>P18</td>
<td>0.0010</td>
</tr>
<tr>
<td>P4</td>
<td>0.6768</td>
<td>P19</td>
<td>-0.00253</td>
</tr>
<tr>
<td>P5</td>
<td>-0.4961</td>
<td>P20</td>
<td>-0.0054</td>
</tr>
<tr>
<td>P6</td>
<td>-0.4845</td>
<td>P21</td>
<td>-0.0012</td>
</tr>
<tr>
<td>P7</td>
<td>-0.0636</td>
<td>MSIZE</td>
<td>00036</td>
</tr>
<tr>
<td>P8</td>
<td>0.0254</td>
<td>NT</td>
<td>01000</td>
</tr>
<tr>
<td>P9</td>
<td>0.0193</td>
<td>TOP</td>
<td>0.8000</td>
</tr>
<tr>
<td>P10</td>
<td>-0.0003</td>
<td>BOT</td>
<td>0.2000</td>
</tr>
<tr>
<td>P11</td>
<td>1.9493</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
to fcc platinum. The platinum energy bands have been fit to the accurate and extensive dHvA experiments of Ketterson and Windmiller. 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:

```fortran
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 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
1 FORMAT(F9.6)
```
FORMAT(15)
FORMAT(1H1,5X,'THE TOTAL NUMBER OF POINTS IS ',15,//)
FORMAT(1H1,5X,'THE RANGE IS FROM ',F10.4, '10.4,///)
FORMAT(1H1,5X,'THE DENSITY OF TOTAL ELECTRON STATES ',///)
FORMAT(1H1,5X,9X,'ENERGY',13X,'DENSITY OF STATES IN RY
1 AND EV',9X,'TOTAL STATES',///)
FORMAT(5X,21.5X,F10.5,10X,E15.5X,E15.5X,E15.5X,E15.5)
FORMAT(5X,15.4E15.4)
 FORMAT(1H1,10X,'THE TOTAL NUMBER OF POINTS USED HERE IS ',11.///
0 FORMAT(10X,3115.///)
502 FORMAT(10X,9F10.4,///)
505 FORMAT(1H1,10X,'THE TOTAL NUMBER OF K SPACE POINTS IS',110,///)
2121 FORMAT(3F10.4)
2122 FORMAT(3E15.8)
DIMENSION IVE(3), CEP(9,10), DK(3), C(10)
DIMENSION SMI(9), DMI(9), SM12(9), DMI1(9), DMIT(9)
DIMENSION D1(2000). SS(680)
1.NC(3). E(9)
DIMENSION ADV(3)
REAL*8 HIST(13,500). HOST(13,500). HLOW(13). SUM.SNUM.SC
COMMON /DEBUG/ HIST.HOST
COMMON/HOTDOG/N69
COMMON/BLOK/EMESH.MESH.IK.NTOT.MC.IP
C
C**************************************************************
C
C REAL*8 HIST(13,500). HOST(13,500). HLOW(13). SUM.SNUM.SC
COMMON /DEBUG/ HIST.HOST
COMMON/HOTDOG/N69
COMMON/BLOK/EMESH.MESH.IK.NTOT.MC.IP
C
C**************************************************************
C
C SET CONSTANTS
N69 = 0
SQ3=SQR(3.)
DO 111 I = 1,13
HLOW(I) = 0.000
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**************************************************************
C
C SET UP DATA FROM CARDS
CALL SETUP
C
C**************************************************************
C
C FILL UP LATTICE POINTS
CALL ALL
C
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)

CONTINUE

C
C  SET UP SCALE
IMESH=2*MESH+1
XL=IMESH-1
SCALE=8./XL

C  CLEAR HISTOGRAMS AND SET UP PARAMETERS
DO621=1,ND
DO 62 J = 1,13
HIST(J,1) = 0.000
HOST(J,1) = 0.000
CONTINUE

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  BEGIN LOOP ON INDIVIDUAL LATTICE BOXES
C
DO100IX=1,MESH
DO100IY=1,MESH
DO 100 IZ=1,MESH

I C E ( I ) = I X
I C E ( 2 ) = I Y
I C E ( 3 ) = I Z

C  SET UP COEFFICIENTS
C
CALLSNORT(I C E ,CEPI)

RUN ON INNER LOOP USING EXPANSION COEFFICIENTS
C
DO50NC=1,NT
SCALE UP VECTORS
DO631=1,3
DK(I) = RANF(-1)*2-1
CONTINUE

READY TO ADD AND CALL CELL
CALLCELL(DK,C)
C  LOAD IT IN 9 LEVELS
C
DO 40 JG=1,9

C ER=0.
PD1=0.0
PD2=0.0
PD31=0.0
PD32=0.0
PS1=0.0
PS2=0.0
C DO 30 IG=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
CONTINUE
C
30 CONTINUE
C
IF(ER.LT.BOT ) GOTO85
IF(ER.GT.TOP)GOTO80
NSET=(ER-BOT)/DEL+1.
C
COIUNT TWO ELECTRONS
C
80 CONTINUE
GO TO 86
85 CONTINUE
C
HLOW(JG) = HLOW(JG)+2.000
HLOW(10) = HLOW(10)+2.000
HLOW(11) = HLOW(11)+2.000*(PD1+PD2)
HLOW(12) = HLOW(12)+2.000*(PD31+PD32)
HLOW(13) = HLOW(13)+2.000*(PS1+PS2)
C
86 CONTINUE
C
SNUM=SNUM+2.000
C
DO 100 J = 1,13
SUM = HLOW(J)•SC
DO110I=1,NO
HIST(J,I)=HIST(J,I)•SC
SUM=SUM+HIST(J,I)
HIST(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=NO/55
DO200I=1,NG
WRITE(6,10)
NTOP=1+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,HIST(K,J)
C
PUNCH2,K,J,ENG,HIST(K,J),HS,HIST(K,J)
200 CONTINUE
300 CONTINUE
C
C*****************************************************************************
C  STOP
C*****************************************************************************
C
SUBROUTINE ALL
C*****************************************************************************
C  J353-FILLS ALL POSSIBLE LATTICE 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 GAMMA TO X
C  IK SORTS VECTORS. EMESH ARE EIGENVALUES
C  NTOT IS THE NUMBER OF DOUBLE CELL POINTS
C*****************************************************************************
C
DIMENSION DME21(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)
DIMENSIONS MI1(9),MI2(9),SM1(9),SM2(9),DM11(9),DM12(9),DMT2(9)
DIMENSION EMESH(9,2000),IP(25,25,25),IK(3,2000)
DIMENSION BK(3),EG49)
C
COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
7DME22,DE22,SM1,SM2,DM11,DM12,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
DO501Y=1,IMESH
DO501X=1,1Y
DO501Z=1,1X
C
C  PUT IN L PLANE
IT=IX*1Y*1Z-3
IF(MESH-3.LT.IT) GOTO50

INDEX=INDEX+1
I=INDEX
IP(IX,LY,IZ)=INDEX
IK(1,I)=IX
IK(2,I)=IY
IK(3,I)=IZ

50 CONTINUE

NTOT=INDEX

***** IK NOW DONE

SET SCALE OF ZONE

XL=IMESH-1
SCALE=8./XL

LOAD EMESH

DO100I=1,NTOT

RESET VECTOR SO THAT GAMMA=ZERO

DO60J=1,3
BK(J)=(IK(J,I)-1)*SCALE
60 CONTINUE

CALL MATRIX AND EIGENVALUE Routines

CALL ZEP (BK,EG)

FILE INTO EMESH

DO70J=1,9
EMESH(J,I) = EG(J)
SME1(J,I)=SM1(J)
SME2(J,I)=SM2(J)
DME31(J,I)=DM1(J)
DME32(J,I)=DM2(J)
DME21(J,I)=DM31(J)
DME22(J,I)=DM32(J)
70 CONTINUE

100 CONTINUE

*** MESH LOADED

RETURN

END

SUBROUTINE CELL(BK,C)

J351-FOR USE IN DENSITY OF STATES

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     DIMENSION BK(3), C(10)
C***********************************************************************
C     SET UP VALUE AT THIS POINT OF KSPACE
C***********************************************************************
C     SQ 3 = SQRT(3.)
C     C(1) = 1.
C     C(2) = BK(1)
C     C(3) = BK(2)
C     C(4) = BK(3)
C     C(5) = BK(1) * BK(2)
C     C(6) = BK(1) * BK(3)
C     C(7) = BK(2) * BK(3)
C     C(8) = (BK(1) * BK(1) - BK(2) * BK(2)) * .5
C     C(9) = .5 * (3. * BK(3) * BK(3) - 1.) / SQ3
C     C(10) = BK(1) * BK(1) + BK(2) * BK(2) + BK(3) * BK(3)
C***********************************************************************
C     RETURN
C END

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

C ************************************************************************************
C MODIFICATION OF AMLF202
C ************************************************************************************

DIMENSION A(36,36),R(36,36),VALU(36),VAUL(36),Q(36),QV(36),SCALAR(36),IND(36),U(36)
COMMON A,VALU

N=NSUB

C CALCULATE NORM OF MATRIX

ANORM2=0.0
DO 6 I=1,N
   DO 6 J=1,N
      IF(ABS(A(I,J)).LT.LE-12)A(I,J)=0.0
6 CONTINUE
ANORM2=ANORM2+A(I,J)**2

C GENERATE IDENTITY MATRIX

7 ANORM=SORT(ANORM2)
9 IF (M)10,45,10
10 DO 40 I=1,N
   DO 40 J=1,N
      IF(I-J).EQ.35.25.35
         8(I,J)=1.0
      30 GO TO 40
50 8(I,J)=0.0
40 CONTINUE

C PERFORM ROTATIONS TO REDUCE MATRIX TO JACOBI FORM

DIMENSION WVEC(36),PVEC(36),QVEC(36)

50 NN=N-2
52 IF (NN)80,170,55
50 DO 160 I=1,NN
   58 II=I+1
   60 12=I+2
   65 SUM=0.0
   68 DO 70 J=I+1,N
      69 SUM=SUM+A(I,J)**2
70 CONTINUE
   75 SUM=SORT(SUM*A(I,II)**2)
80 WVEC(II)=SORT(1.0*ABS(A(I,II))/SUM)
82 DIV=SIGN(SUM*WVEC(II),A(I,II))
83 DO 85 J=I+1,N
   84 WVEC(J)=WVEC(J)/DIV
85 CONTINUE
   88 NI=N-1
90 CALL MATMPY(NI,NI,WVEC(II),A(II,II),PVEC(II))
95 CALL MATMPY(NI,NI,WVEC(II),PVEC(II),SCALAR)
98 SCALAR=0.5*SCALAR
100 DO 105 J=II,N
   102 QVEC(J)=PVEC(J)-SCALAR*WVEC(J)
105 CONTINUE
110 A(K,L)=A(K,L)-(WVEC(K)*QVEC(L)+WVEC(L)*QVEC(K))
112 IF(ABS(A(K,L)).LT.LE-12)A(K,L)=0.0
120 CONTINUE

C MOVE JACOBI FORM ELEMENTS AND INITIALIZE EIGENVALUE BOUNDS
C 170 DO 200 I=1,N
   VALU(I)=ANORM
   VALL(I)=-1.0*ANORM
200  DIAG(I)=A(I,1)
210  DO 230 I=2,N
   SUPERD(I-1)=A(I-1,1)
220  Q(I-1)=(SUPERD(I-1))^2
230 DO 410 I=1,N
      Tau=0.0
      T1=0.0
235   MATCH=0
240   T2=0.0
   T1=1.0
270  DO 450 J=1,N
   P=DIAG(J)-Tau
275   IF(T1) 350. 330.
   IF(T1) 335. 310.
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
   T1=-1.0
   T=-P
340   IF(T1) 335. 350. 350
   T1=1.0
355   T1=T
360 GO TO 410
370 IF(T1) 335. 350. 350
   T1=1.0
380 IF(T1) 335. 350. 350
390 T=-1.0
395 GO TO 410
400 T1=1.0
410 IF(T1) 335. 350. 350
   T1=1.0
   T=-P
420 IF(T1) 335. 350. 350
   T1=T
425 IF(T1) 335. 350. 350
   T1=1.0
430 IF(T1) 335. 350. 350
440 IF(Abs(T)-1.0E20).445.445.442
442 T1=T1/T
443 T1=1.0
444 T2=T1
445 T2=T1
450 T1=T
C COUNT AGREEMENTS IN SIGN
C 460  DO 530 K=1,N
   IF(K-MATCH) 470. 470. 520
   IF(TAU-VALL(K)) 530. 530. 480
480  VALL(K)=TAU
490 GO TO 530
520 IF(TAU-VALL(K)) 525. 530. 530
525  VALU(K)=TAU
530 CONTINUE
540 IF(VALL(I)-VALL(I-ACC))570.570.550
550 IF(VALL(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 JACOBI EIGENVECTORS BY ROTATIONAL TRIANGULARIZATION
C 590 IF(M)593.890.593
CONTINUE
DO 610 I = 1, N
610 A(I,J) = 0.0
DO 615 I = 1, N
IF (I-1.0E-7 > 0.0) THEN
621 IF (VALU(I-1) - VALU(I) - 5.0E-7 > 0.0) THEN
622 IF (VALU(I-1) > 0.0) THEN
623 IF (ABS(VALU(I)/VALU(I-1) - 1.0) - 5.0E-7 < 0.0) THEN
625 COS = 1.0
628 SIN = 0.0
DO 630 J = 1, N
635 IF (J-1) > 680.0, 680.0
640 T = SQRT(T1**2 + T2**2)
645 COS = T1/T
SIN = T2/T
650 IF (T1) = 0.0, 0.0
660 COS = 1.0
SIN = 0.0
DO 670 J = 1, N
675 IF (ABS(SMALLD - AHS(D(J))) > 0.0) THEN
680 T2 = SUPERD(J)
897 FORMAT(8E12.4)
700 BETA = SUPERD(J) * COS
710 D(N) = T
720 DO 725 J = 1, N
725 IND(J) = 0
730 SMALLD = ANORM
735 DO 780 J = 1, N
740 .IF (IND(J) = 1) THEN
750 IF (ABS(SMALLD - AHS(D(J))) > 0.0) THEN
760 SMALLD = D(J)
770 NN = J
780 CONTINUE
790 IND(NN) = 1
800 PRODS = 1.0
805 IF (NN-1) > 1810.0, 1850.0
810 DO 840 K = 2, NN
820 II = NN+1-K
830 A(II+1, I) = C(II) * PRODS
840 PRODS = PRODS * S(CII)
850 A(1, I) = PRODS
C FORM MATRIX PRODUCT OF ROTATION MATRIX WITH JACOBI VECTOR MATRIX
C
C = SQRT(3.0)
C SET UP THE IMAGINARY PARTS

SUBROUTINE FILL
C*****************************************************************************
C RETURNS THE SYMMETRIZED SPIN ORBIT MATRIX
C F. M. MUELLER MARCH 1966
C*****************************************************************************
C
DIMENSION AD(12), AL(20), DIFER(20), BK(3)
DIMENSION SOB(36), EV(36), D(5)
COMMON SOB, EV, VO, V1, V2, V3, V4, ENGL, D, T, EJ, RO, 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 J = 1,9
DO 30 K = 1,3
C
M = I+9*K
N = J+9*K
C
30  SOB(M,N) = SOB(I,J)
C
C***********************************************************************************************
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
C THE FOURTH IS \((X^2-Y^2)/2\)
C THE FIFTH IS \((3Z^2-1)/(2\sqrt{3})\)

C ******************************************
C F.M. MUeller March 1966
C ******************************************
DIMENSION PREM(16), BK(3), D(5), AD(12), S(36), EV(36), EG(9)
DIMENSION AL(20), DIFER(20)
1. CK(3), EZ(9)
COMMON S, EV, VO, V1, V2, V3, V4, ENGL, DT, ELK0, R1, EP, A, AL, DIFER, AK
1. CNORM, MSIZE

C SET CONSTANTS
C
PI = 3.14159265
8 = SQRTF(3.0)

C SET VARIABLES
C
X = PI*AK(1)/8.
Y = PI*AK(2)/8.
Z = PI*AK(3)/8.

C SET COS, SIN FUNCTIONS
C
CX = COSF(X)
CY = COSF(Y)
CZ = COSF(Z)
SX = SINF(X)
SY = SINF(Y)
SZ = SINF(Z)
CZX = COSF(2.*X)
CZY = COSF(2.*Y)
CZZ = COSF(2.*Z)

C GENERATION OF THE MATRIX ELEMENTS
C
C THE DIAGONAL TERMS
H(1,1) = A(1)+4.*A(2)*CX*CY+4.*A(3)*CY*CX
1+2.*A(9)*C2X+C2Y+2.*A(10)*C2Z
H(2,2) = A(1)+4.*A(2)*CX*CZ+4.*A(3)*CX*CZ
1+2.*A(9)*C2X+C2Z+2.*A(10)*C2Y
H(3,3) = A(1)+4.*A(2)*CY*CZ+4.*A(3)*CY*CZ
1+2.*A(9)*C2Y+C2Z+2.*A(10)*C2X
H(4,4) = A(6)+3.*A(7)*(CX*CY+CZ)+A(8)*(4.*CX*CY+CZ+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+CZ+CY+CZ
1+(0.5)*A(11)+(C2X+C2Y)+4.*C2Z+(1.5)*A(12)+(C2X+C2Y)

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*S2
H(3,5) = 2*B*A(5)*SY*S2
H(4,5) = B*A(7)*(CK*C2-CY*C2)-B*A(8)*(CK*C2-CY*C2)
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), EV(36), D(5), DIFER(20), BK(3), AL(20, 9), AD(12)
DIMENSION CK(3), ZK(4, 3)
C
COMMON EV, V0, V1, V2, V3, V4, ENGL
COMMON T, EL, RB, RA, EP, AD, AL, DIFER, BK
COMMON/THIEF/AK
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) G01094
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) = 0.5*(A+B)*B
ES(5, I) = 0.5*SQRT((1.3/3.)*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)
GF(I) = SQRT (1 - GF(I) * GF(I) / 3.)

S(M, M) = C0 * ENGL / 48. * V0

DEFINE OFF DIAGONAL ELEMENTS

S(6, 7) = V1
S(6, 8) = V2
S(6, 9) = V1
S(7, 8) = V1
S(7, 9) = V2
S(8, 9) = V1

DO 10 I = 6, 9
  M = I - 5
  DO 10 J = 1, 9
    N = J - 5
    SUM = 0.
    SIM = 0.
    DO 11 K = 1, 5
      SIM = SIM + ES(K, M) * ES(K, N)
      DO 11 L = 1, 5
        A = S(K, L)
        SUM = SUM + ES(K, M) * ES(L, N) * A
    11 continue
    OS(I, J) = S(I, J) - SUM * GF(M) * GF(N) * SIM * (GN(M) * GG(N) * GF(N) + GN(N) * GG(M) * GF(M))
    10 continue
    OS(I, J) = OS(I, J) / (GN(M) * GN(N))

SUBROUTINE MAT-MPY(M, N, X, A, Y)

THIS ROUTINE IS A HELPER TO EIGENB

DIMENSION A(36, 36), X(36), Y(36)

DO 40 I = 1, M
  20 Y(I) = 0.0
  DO 40 J = 1, N
    40 Y(I) = Y(I) + X(J) * A(J, I)

RETURN
END

END
SUBROUTINE MDINV

C**************************************************************
C JJ15 DOUBLE PRECISION MATRIX INVERTER
C NOTE THAT THIS MATRIX INVERTER CAN ONLY BE USED ON NEARLY DIAGONAL
C MATRICES
C WE ARE NOT RESTRICTED TO TO SYMMETRIC MATRICES
C**************************************************************
C IMPLICIT REAL*8(A-E)
C**************************************************************
C DIMENSION A(35,35),IRL(35),ICL(35)
COMMON/OVER/A,N
C**************************************************************
C SET THE VARIOUS CALLS
C**************************************************************
C
M = N
M1 = M-1
IC = 0

DO 1 I = 1,M
  IRL(I) = .99
  ICL(I) = I
  CONTINUE

APIV = 0
BPIV = 1.0E20
ASUM = 0.

DO 2 I = 1,M
  DO 2 J = 1,M
    AT = DABS(A(I,J))
    IF(AT.LT.1.E-30) AT = 0.
    ASUM = ASUM+AT*AT
  CONTINUE

IF(ASUM.LT.1.E-30) GO TO 5000

ASUM = DSQRT(ASUM)
ASUM = 1./ASUM

DO 3 I = 1,M
  DO 3 J = 1,M
    A(I,J) = A(I,J)*ASUM
  CONTINUE

C**************************************************************
C FIND THE LARGEST ELEMENT
C**************************************************************
C
DO 10 I = 1,M
  DO 11 J = 1,M
    A1=A(I,J)
    A2 = DABS(A1)
    IF(A2.LE.APIV) GO TO 11
    APIV = A2
    IT = I
  CONTINUE

11 CONTINUE
IU = J

CONTINUE

CONTINUE

BEGIN THE ITERATION SET

CONTINUE

IF(APIV.EQ.0.) GO TO 14
IF(BPIV.LE.APIV) GO TO 15
BPIV = APIV

IR = IT
IS = IU
ITEMP = IRL(IR)
IRL(IR) = ICL(IS)
ICL(IS) = ITEMP-100
APIV = 0.
DIV = A(IR,IS)
A(IR,IS) = -1.
DO 17 J = 1,N
A(IR,J) = -A(IR,J)/DIV
DO 18 I = 1,M
IF(I.EQ.IR) GO TO 18
A(I,IS) = A(I,IS) - A(I,J)*A(IS,J)
IF(J.GT.M) GO TO 20
IF(IRL(I).LE.50) GO TO 20
IF(I.CL(J).LT.1) GO TO 20
AG = DABS(A(I,J))
IF(AG.LT.APIV) GO TO 20
APIV = AG
IT = I
IU = J
CONTINUE
CONTINUE
IC = IC+1
GO TO 24

THE END

CONTINUE

DO 25 J = 1,M1
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,M
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***********************************************************************
C RETURN
C***********************************************************************
C 500 CONTINUE
C
WRITE(6,100)
100 FORMAT(10X,'THESE MATRIX ELEMENTS ARE ALL SINGULAR',/)
C
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***********************************************************************
C RETURN
C***********************************************************************
C 7 YFL=IX
C RANF=YFL*.4656613E-9
C
C***********************************************************************
C RETURN
C***********************************************************************
C 8 RANF=X
C
C***********************************************************************
C RETURN
C***********************************************************************
C
ENTRY IRANF(J)
C***********************
C 12 IX=IX*65547
C  15 IX=IX+2147483647*16
C  16 IRANF=IX
C***********************
C
C***********************
ENTRY RANSET(J)
C***********************
C
C
C***********************
ENTRY RANGET(J)
C***********************
C
C
C***********************
END

SUBROUTINE RECIP(EK)

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.ZORNBEG
C***********************
C
C DIMENSION AK(4,3),BK(3),EK(3),IL(3)
C COMMON/THIEF/AK
C
C***********************
C
C 72 CONTINUE
C
C  DO 10 I=1,3
C  10  BK(I)=ABS(EK(I))
C  TES=8*BK(I)+BK(2)+BK(I)
C  IF(TES.LE.12.) GO TO 73
C
C  DO 13 I=1,3
C  13  EK(I)=EK(I)-SIGN(8.,EK(I))
C
C  GO TO 72
CONTINUE

DO 11 I=1,3
11 IL(I)=I

DO 12 I=1,4

DO 12 J=1,3
12 AK(I,J)=0.

DO 70 I = 1,2
DO 70 J = 2,3

TEST = BK(I)-BK(J)

IF(TEST.LE.0.) GO TO 70

SAVE = BK(I)
BK(I) = BK(J)
BK(J) = SAVE
KAVE=IL(I)
IL(I)=IL(J)
IL(J)=KAVE

CONTINUE

BK(2) = BK(3)
SAVE = BK(1)
BK(1) = BK(2)
BK(3) = SAVE
KAVE=IL(I)
IL(I)=IL(2)
IL(2)=IL(3)
IL(3)=KAVE
I1=IL(I)
I2=IL(2)
I3=IL(3)

IF(BK(I2).LE.8.) GO TO 71

E2=SIGN(1.,EK(I2))
E2=EK(I2)-E2*16.

GO TO 72

CONTINUE

E2=SIGN(1.,EK(I2))
AK(3,I2)=-E2*16.
AK(2,I3)=-8.
AK(4,I3)=8.

DO 14 I=2,4,2
14 AK(I,I1)=-SIGN(8.,EK(I1))

RETURN

END
SUBROUTINE RF(GI, Z, T, RR)
C
C*******************************************************************************
C THIS ROUTINE CALCULATES THE OVERLAP FORM FACTOR
C F.M. MUELLER MARCH 1966
C*******************************************************************************

31    A = GI * RR
C
3   IF (A - 5.8) 3, 32
C
32    Z = (3./A * A - 1./A) * SIN (A) - 3. * COS (A) / A ** 2
C
GO TO 16
C
31    A = GI * RR
C
C*******************************************************************************
RETURN
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
C  T
A = GI * RR
C
10    IF (A - 4.25) 20, 20, 10
C
16    Z = 0.
C
RETURN
C*******************************************************************************
C
C  T*(5.1 - A) / .85
20    Z = (3./A ** 3 - 1./A) * SIN (A) - 3. * COS (A) / A ** 2
C
Z = Z * C
C
RETURN
C*******************************************************************************
END
C
C
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*******************************************************************************

100 FORMAT(15)
200 FORMAT(3E15.6)
300 FORMAT(F9.6)
350 FORMAT(10X, I5, 10X, F10.4)
600 FORMAT(80A1)
700 FORMAT(1H1,10X,80AI,11D).
800 FORMAT(///,10X,'D BAND PARAMETERS')
C
C******************************************************************************
C
DIMENSION HEAD(80)
DIMENSION AL(20,9),ADIFER(20)
DIMENSION PREM(21),BK(3),D(5),AD(12),SE(36,36),EV(36),EG(9)
COMMON S,E,V,VO,VL,V2,V3,V4,ENGL,D,T,EL,RO,R1,EP,AD,AL,ADIFER,BK
1 ,CNORM,MSIZE
C
C******************************************************************************
C
READ(5,600) HEAD
WRITE(6,700) HEAD
C
C******************************************************************************
C
READ VALUES OF PARAMETERS
C
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
SET UP PARAMETERS
C
VO=PREM(1)
VL=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)
RO=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******************************************************************************
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******************************************************************************
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 J354-FINDS EXPANSION COEFS TO SECOND ORDER
C EMESH AND IK ALREADY LOADED
C........F.M.MUeller, J.W.GARLAND, M.H.FOoEN, AND K.H.BENNEMANN........
C JUNE 1968
C MODIFIED BY S.G.DAS AUGUST 1969
C*************************************************************************
C IVEC IS LOWEST CORNER POINT IN
C MESH-SIZED LATTICE
C*************************************************************************
C DIMENSION DME21(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 MI1(9),MI1(9),SMI(9),DM12(9),DM1T1(9),DM1T2(9)
DIMENSION EMESH(9,2000),IP(25,25,25),IK(3,2000)
DIMENSION IVEC(3),ISVEC(3),IZ(3),IR(3)
DIMENSION NEG(27),Z(3),G(10),IM(3,27)
DIMENSION CEP(9,10)
REAL U(10,27)
C REAL*8 TR(35,35),RI(27,10)
C COMMON/OVER/TR,NGO
COMMON/BLOK/EMESH,IK,NTOT,MC,IP
COMMON/GURU/DME21,DE21,SME1,SEP1,SME2,SEP2,DME31,DE31,DME32,DE32,
7DME22,DE22,SMI,SMI,DM1,DM12,DM1T1,DM1T2
COMMON/HOTDOG/N69
C*************************************************************************
C NGO = 10
C*************************************************************************
C INITIATE THE CEP'S TO ZERO
C*************************************************************************
C DO 11 I = 1,9
DO 1 J = 1, 10
C CEP(I, J) = 0.
SEP(1, J) = 0.0
SEP2(1, 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 DO4 I = 1, 10
DO3 J = 1, 27
R(J, I) = 0.
C 4 CONTINUE
C******************************************************************************
C SET SCALE FOR DOUBLE SIZED MESH
C******************************************************************************
C IMESH = 2 * MESH + 1
XL = IMESH - 1
SCALE = 8. / XL
C******************************************************************************
C PUT IVEC INTO DOUBLE LATTICE
C******************************************************************************
C DO10 I = 1, 3
C ISVEC(I) = 2 * (IVER(I) - 1)
C 10 CONTINUE
C******************************************************************************
C SET UP TRIPLE-LOOP FOR 27 CELL POINTS
C******************************************************************************
C NR = 0
C DO20 I = 1, 3
DO20 J = 1, 3
DO20 K = 1, 3
C******************************************************************************
C IF WANT RESTRICTION TO FCC PUT HERE
C******************************************************************************
C IZ(1) = 1 - 2
IZ(2) = J - 2
IZ(3) = K - 2
C NR = NR + 1
C DO15 L = 1, 3
C IR(L) = IZ(L)
IM(L, NR) = IR(L)
Z(L) = IR(L)
C 15 CONTINUE
C CALL CELL(Z, G)
C DO16 L = 1, 10
R(NR,L)=G(L)

CONTINUE

FORM TR MATRIX AND INVERT IT

IF (N69.EQ.69) GO TO 169

DO 30 I=1,NR

TR(I,J)=0.

DO 30 K=1,NR

TR(I,J)=TR(I,J)+R(K,I)*R(K,J)

CONTINUE

CALL MDINV

CONTINUE

FORM U MATRIX

DO 35 I=1,NR

U(I,J)=0.

DO 35 J=1,NR

U(I,J)=U(I,J)+TR(I,J)*R(J,L)

CONTINUE

SUM OVER THE FITTING POINTS

DO 50 L=1,NR

SET UP K VECTOR

ISUM=0

DO 41 K=1,3

IR(K)=IM(K,L)+ISVEC(K)+2

CHECK ON LENGTH

ISUM=ISUM+IR(K)-1

CONTINUE
IDIF=ISUM-3*MESH
IF(IDIF.LE.0)GOTO43

C******************************************************************************
C VECTOR OUTSIDE
C******************************************************************************

DO42I=1,3
IR(I)=(MESH+2)-(IR(I)-1)
IR(I)=ABS(IR(I))+1
42 CONTINUE

DO43I=1,3
II=I+1
DO45J=II,3
IDIF=IR(I)-IR(J)
IF(IDIF.GE.0)GOTO45
ISAVE=IR(I)
IR(I)=IR(J)
IR(J)=ISAVE
45 CONTINUE

IZE=IR(3)
IX=IR(2)
IY=IR(1)

C******************************************************************************
C VECTOR NOW PROPER
C******************************************************************************

INDEX=IP(IX, IY, IZE)

C******************************************************************************
C SUM OVER THE BANDS
C******************************************************************************

D049N=1,9
ENG=EMESH(N, INDEX)
PRSI=SME1(N, INDEX)
PRS2=SME2(N, INDEX)
PRD1=DME21(N, INDEX)
PRD2=DME22(N, INDEX)
PRD32=DME32(N, INDEX)
PRD31=DME31(N, INDEX)
D048K=1,10
C
C******************************************************************************
C SETUP FINAL SUM
C******************************************************************************

CEP(N,K)=CEP(N,K)+DE(K,L)*ENG
SUBROUTINE SYMCS(N)

C*---------------------------------------------------------------*
C SUBROUTINE TO SYMMETRIZE THE HAMILTONIAN MATRIX
C F.M. MUELLER MARCH 1966
C*---------------------------------------------------------------*

DIMENSION S(36,36)

DO 10 I = 1, N
DO 10 J = I, N
10 S(J,I) = S(I,J)

C*---------------------------------------------------------------*
C END
C*---------------------------------------------------------------*

SUBROUTINE ZEP(CK,EZ)

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*---------------------------------------------------------------*

DIMENSIONSMI I(9), DMI1(9), DMI2(9), DMI1(9), DMIT2(9)
DIMENSIONME21(9,300), DE21(9,10), SME1(9,300), SEPI(9,10), SME2(9,8300), SEPI(9,10), SME31(9,300), DE31(9,10), SME32(9,300), DE32(9,4300), SME32(9,300), DE32(9,10)
DIMENSION AL(20,9), DFER(20)
1. CK(3), EZ(9)
DIMENSION PREM(16), BK(3), AD(5), AL(12), SG(9), EV(36), EG(9)

COMMON/GURU/DE21,DE21, SME1, SEPI, SME2, SEPI, DME31, DE31, DME32, DE32,
7 DME22, DME22, SME1, SME2, SME31, DE32, SME32, SME32,
7 COMMON S, EV, V0, V1, V2, V3, V4, ENGL, T, EL.R0, R1, EP, AD, AL, DFER, BK
1. CNORM, MSIZE
C*---------------------------------------------------------------*
C 14 FORMAT(2I5,6F10.5)
DO91=1.9
C
SM1=0.0
SM1=0.0
DMI1=0.0
DMI2=0.0
DM1I2=0.0
DMIT1=0.0
C
9 CONTINUE
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 SYMS(S,5)
CALL HSOC
CALL SYMS(S,9)
C
IF(MSIZE.LE.9) GO TO 35
C
35 CONTINUE
C
ACC=.0001
C
CALL EIGENB(MSIZE,MSIZE,MSIZE,ACC)
C
MC=MSIZE/36
C
DO401=1,9
C
II=10-I
IT=(II-1)*MC+3+II
C
EZ(I) = EV(IT)+CNORM
40 EG(I) = EV(IT)+CNORM
C
DO11NLEVEL=1,9
C
ISUM=10-NLEVEL
DO1000 NGO=1,1
IS=ISUM+MC*3*(ISUM-1)
IS=IS*NGO-1
D1=0.0
D2=0.0
D31=0.0
D32=0.0
S1=0.0
S2=0.0
DO281=4.5
K=I+9
J=I+18
L=K+18
D1=D1*S(I,IS)**2*S(J,IS)**2
D2=D2*S(K,IS)**2*S(L,IS)**2
28 CONTINUE
D281=1.3
K=I+9
J=I+18
L=K+18
D31=D31*S(I,IS)**2*S(J,IS)**2
D32=D32*S(K,IS)**2*S(L,IS)**2
28 CONTINUE
D271I=6.9
K=I+9
J=I+18
L=K+18
S1=S1*S(I,IS)*S(J,IS)
S2=S2*S(K,IS)*S(L,IS)
271 CONTINUE
S1=S1*S1
S2=S2*S2
SMI1(NLEVEL)=S1
SMI(NLEVEL)=S2
DMT1(NLEVEL)=D31
DMT2(NLEVEL)=D32
DMT1(NLEVEL)=D1
DMT2(NLEVEL)=D2
1000 CONTINUE
11 CONTINUE
C**************************************************************
RETURN
C**************************************************************
END
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
0.6565
0.974494
0.001000
-0.002300
0.005400
-0.001200
36
2.200
1.4
-0.2
REFERENCES


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


